[[Table of Contents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#toc)] [[UAAG 2.0 Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/)]

[W3C](http://www.w3.org/)

Implementing UAAG 2.0

A guide to understanding and implementing User Agent Accessibility Guidelines 2.0

W3C Working Draft 7 November 2013

**This version:**

<http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20131107/>

**Latest version:**

<http://www.w3.org/TR/IMPLEMENTING-UAAG20/>

**Previous version:**

<http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20130606/>

**Latest Editor's Draft:**

<http://www.w3.org/WAI/UA/IMPLEMENTING-UAAG20/>

**Editors:**

James Allan, Texas School for the Blind and Visually Impaired

Kelly Ford, Microsoft

Kim Patch, Redstart Systems

Jeanne Spellman, W3C/Web Accessibility Initiative

[Copyright](http://www.w3.org/Consortium/Legal/ipr-notice#Copyright) © 2013 [W3C](http://www.w3.org/)® ([MIT](http://www.csail.mit.edu/), [ERCIM](http://www.ercim.eu/), [Keio](http://www.keio.ac.jp/), [Beihang](http://ev.buaa.edu.cn/)), All Rights Reserved. W3C [liability](http://www.w3.org/Consortium/Legal/ipr-notice#Legal_Disclaimer), [trademark](http://www.w3.org/Consortium/Legal/ipr-notice#W3C_Trademarks) and [document use](http://www.w3.org/Consortium/Legal/copyright-documents) rules apply.

Abstract

This document provides supporting information for the [User Agent Accessibility Guidelines (UAAG) 2.0](http://www.w3.org/TR/UAAG20/) for designing [user agents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) that lower barriers to Web accessibility for people with disabilities. User agents include browsers, media players and applications that retrieve and render [Web content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content). A user agent that [conforms](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#conformance) to these guidelines will promote accessibility through its own user interface and its ability to communicate with other technologies (especially [assistive technologies](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-assistive-technology)). All users, not just users with disabilities, should find conforming user agents to be more usable. In addition to helping developers of browsers and media players, this document will also benefit developers of assistive technologies because it explains what types of information and control an assistive technology may expect from a conforming user agent.

This document provides explanation of the intent of UAAG 2.0 success criteria, examples of implementation of the guidelines, best practice recommendations and additional resources for the guideline.

The "User Agent Accessibility Guidelines 2.0" (UAAG 2.0) is part of a series of accessibility guidelines published by the W3C [Web Accessibility Initiative](http://www.w3.org/WAI/) (WAI).

Status of this document

May be Superseded 可能作废

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of currentW3C publications and the latest revision of this technical report can be found in the*[*W3C technical reports index*](http://www.w3.org/TR/)*at* [*http://www.w3.org/TR/*](http://www.w3.org/TR/)*.*

Working Draft of Implementing UAAG 2.0

This is the W3C Working Draft of 7 November 2013. This working draft is updated from comments received on the 23 May 2013 Working Draft.

Substantial changes in this draft include:

* [1.1.1 Render Alternative Content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_111-e) Added an example of <abbr> alternative content and moved the intent and examples from 2.11.8 to 1.1.1 when it was agreed that 2.11.8 was a subset of 1.1.1.
* [1.2.1 Support Repair by Assistive Technologies:](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_121-e)Changed the intent and added a new example to provide a clearer example of the desired behavior. .
* [2.3.5 Customize Keyboard Commands](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_235) added language to the intent that the customizations should be saved as required in [2.7.1 Allow Persistent Accessibility Settings](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_271).

The Working Group seeks feedback on the following points for this draft: .

* What accessibility needs specific to mobile web application user interfaces have not been addressed that should be addressed by UAAG 2.0?
* Are there sufficient examples of other technologies that support accessibility, e.g. WAI-ARIA, HTML5 Canvas, IndieUI?

Comments on this draft should be sent to [public-uaag2-comments@w3.org](mailto:public-uaag2-comments@w3.org) ([Public Archive](http://lists.w3.org/Archives/Public/public-uaag2-comments/)) **by 16 December 2013**.

The [User Agent Accessibility Guidelines Working Group](http://www.w3.org/WAI/UA/) (UAWG) expects to advance [UAAG 2.0](http://www.w3.org/TR/UAAG20/) through the W3C Recommendation track with Implementing UAAG as a supporting Note. Until that time [User Agent Accessibility Guidelines 1.0](http://www.w3.org/TR/UAAG10/) (UAAG 1.0) [[UAAG10]](http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20130523/#ref-UAAG10) is the stable, referenceable version. This Working Draft does not supersede UAAG 1.0.

Web Accessibility Initiative

This document has been produced as part of the W3C [Web Accessibility Initiative](http://www.w3.org/WAI/) (WAI). The goals of the User Agent Working Group (UAWG) are discussed in the [Working Group charter](http://www.w3.org/WAI/AU/charter4.html). The UAWG is part of the [WAI Technical Activity](http://www.w3.org/WAI/Technical/Activity).

No Endorsement

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Patents

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](http://www.w3.org/Consortium/Patent-Policy-20040205/). The group does not expect this document to become a W3C Recommendation. W3C maintains a [public list of any patent disclosures](http://www.w3.org/2004/01/pp-impl/36791/status) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim(s)](http://www.w3.org/Consortium/Patent-Policy-20040205/#def-essential) must disclose the information in accordance with [section 6 of the W3C Patent Policy](http://www.w3.org/Consortium/Patent-Policy-20040205/#sec-Disclosure).

Table of Contents

* [Abstract](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#abstract)
* [Status of this Document](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#status)
* [Introduction](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#introduction)
  + [Levels of Conformance](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#intro-conf-levels)
  + [Definition of User Agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#intro-def-ua)
  + [Relationship with Web Content Accessibility Guidelines(WCAG) 2.0](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#intro-wcag)
  + [Relationship with Authoring Tool Accessibility Guidelines (ATAG) 2.0](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#intro-atag)
* [UAAG 2.0 Guidelines](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#Guidelines)
  + [UAAG 2.0 Conformance Applicability Notes](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#applicability-notes)
  + [PRINCIPLE 1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#principle-perceivable): Perceivable
    - [Guideline 1.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-access-alternative-content): Alternative content 替代内容
    - [Guideline 1.2](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-missing-alt): Missing content 内容缺失
    - [Guideline 1.3](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-interaction-highlight): Highlighting 高亮显示
    - [Guideline 1.4](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-text-config): Text configuration 文本配置
    - [Guideline 1.5](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-volume-config): Volume configuration 音量配置
    - [Guideline 1.6](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-speech-config): Synthesized speech configuration 合成语音配置
    - [Guideline 1.7](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config): User style sheet configuration 用户样式配置
    - [Guideline 1.8](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-viewport-orient): Orientation in viewports 视窗中的定位
    - [Guideline 1.9](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-alternative-views): Alternative views
    - [Guideline 1.10](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-info-link): Element Information 元素信息
  + [PRINCIPLE 2](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#principle-operable). Operable 可操作的
    - [Guideline 2.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-keyboard-access): Keyboard access 键盘访问
    - [Guideline 2.2](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-sequential-navigation): Sequential navigation 循序导航
    - [Guideline 2.3](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-direct-navigation-and-activation): Direct navigation and activation
    - [Guideline 2.4](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-search-text): Text Search 文本搜索
    - [Guideline 2.5](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-nav-structure): Structural navigation 结构导航
    - [Guideline 2.6](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-device-independent-handlers): Event handlers 事件处理器
    - [Guideline 2.7](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-store-prefs): Preference settings 偏好设定
    - [Guideline 2.8](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-configure-controls): Graphical controls 图形控制（件）
    - [Guideline 2.9](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-time-independent): Time-independent interaction 不随时间变化的交互
    - [Guideline 2.10](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-prevent-flash): Flashing 闪光
    - [Guideline 2.11](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-control-inaccessible-content): Time-based media 时基媒体
    - [Guideline 2.12](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-other-devices): Other Input Devices 其它输入设备
  + [PRINCIPLE 3](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#principle-understandable): Understandable 可以理解
    - [Guideline 3.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-avoid-unnecessary-messages): Unnecessary messages 不必要的消息
    - [Guideline 3.2](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-avoid-mistakes): Mistakes 错误
    - [Guideline 3.3](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-doc-access-features): Documentation 文档
    - [Guideline 3.4](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-predictable-operation): Predictable 可预测的
  + [PRINCIPLE 4](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#principle-AT-access). Programmatic access 程序化访问
    - [Guideline 4.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-AT-access): Assistive technology 辅助技术
  + [PRINCIPLE 5](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#principle-follow-specs): Specifications and conventions 规范及惯例
    - [Guideline 5.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-obs-env-conventions): Follow specifications 遵循规范
* [Conformance](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#conformance)
* Appendix A: [Glossary](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#glossary)
* Appendix B: [How to refer to UAAG 2.0 from other documents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-this-doc)
* Appendix C: [References](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#references)
* Appendix D: [Acknowledgements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#acknowledgments)

Introduction 介绍

This document – Implementing UAAG 2.0 – is a companion document to [User Agent Accessibility Guidelines (UAAG) 2.0](http://www.w3.org/TR/2013/WD-UAAG20-20131107/).

UAAG 2.0 provides three layers of guidance for lowering barriers to accessibility: overall principles, general guidelines, and testable success criteria. The principles, guidelines and success criteria are [normative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-normative), meaning user agent developers are required to follow them to claim conformance to UAAG 2.0.

Implementing UAAG 2.0 is [informative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-informative). It provides more detail for each success criterion, including explanatory intent, examples of how the criterion may apply in different user situations, and links to resources. It also contains more detail on levels of conformance, the definition of user agent, the relationship between UAAG 2.0 and with WCAG 2.0 requirements, and the role of user agents and web authoring.

Levels of Conformance

User agents may conform to UAAG 2.0 at one of three conformance levels: levels A (basic), AA (recommended), and AAA (advanced). These levels provide a path for user agent developers to improve their product over time and to prioritize new ways to improve accessibility. The three levels of UAAG 2.0 conformance are based on the corresponding level designations (A, AA, or AAA) of the individual success criteria (i.e., specific requirements). The user agent can conform to a level by meeting the success criteria of that level and the levels below it.

UAAG 2.0 has many options that can be managed through preference settings. Having too many options may be overwhelming for some users. The levels can help user agent developers decide which options to provide in a basic user interface, and which to provide through progressive disclosure to advanced users.

* Level A conformance: The user agent complies with all applicable level A success criteria.
* Level AA conformance: The user agent complies with all applicable level A and AA success criteria
* Level AAA conformance: The user agent complies with all applicable level A, AA, and AAA success criteria.

Factors that were considered in the process of determining the level to a success criterion include:

* severity of impact to the user
* inconvenience to other groups of users, including those with other disabilities
* how common implementations are today
* difficulty of implementation
* size of scope

The level designations of the individual success criteria balance the needs of people with disabilities with the difficulty the user agent developer could experience in meeting that need. There are many different types of disabilities and different types of user agents, so the UAAG level assigned to a success criterion may not precisely match the definition of the level in all circumstances. To avoid over-complication, the various combinations of factors were separated into 3 levels:

Level A success criteria address needs where (a) groups of people with disabilities are blocked from information or accomplishing a task, and/or (b) provide solutions that are relatively minor for developers to implement or are common in the marketplace.

Level AA success criteria address needs where (a) groups of people with disabilities have difficulty accessing information or accomplishing a task (including tasks causing excessive fatigue) and/or (b) the solutions may be more difficult to implement.

Level AAA success criteria address needs where (a) the solution improves accessibility or reduces fatigue for specific groups of people with disabilities and/or (b) the solution is very difficult to implement.

Even user agents that conform at the highest level (AAA) may not be accessible to individuals with all types, degrees, or combinations of disability.

Definition of ***User Agent 定义用户代理***

A user agent is any software that retrieves, renders and facilitates end-user interaction with web content. See the Glossary definition of [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) for details.

The classic user agent is a browser. A media player, which only performs these functions for time-based media, is also user agent. Web applications and some mobile apps that render web content are also user agents.

For information on authoring tool features included in user agents: see [Relationship to the Authoring Tool Accessibility Guidelines (ATAG) 2.0](http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20131107/#intro-atag). For information on the difference between web applications and content see [Relationship to the Web Content Accessibility Guidelines](http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20131107/#intro-wcag).

Relationship to the Authoring Tool Accessibility Guidelines (ATAG) 2.0 与ATAG2.0的关系

While it is common to think of user agents retrieving and rendering web content for one group of people (end-users) that was previously authored by another group (authors), user agents are also frequently involved with the process of authoring content.

For these cases, it is important for user agent developers to consider the application of another W3C-WAI Recommendation, the [Authoring Tool Accessibility Guidelines (ATAG)](http://www.w3.org/WAI/intro/atag.php). ATAG (currently 2.0 is in draft) provides guidance to the developers of tools regarding the accessibility of authoring interfaces to authors (ATAG 2.0 Part A) and ways in which all authors can be supported in producing accessible web content (ATAG 2.0 Part B).

For more information on the role of user agents in web authoring see [Implementing UAAG 2.0](http://www.w3.org/TR/2013/WD-IMPLEMENTING-UAAG20-20131107/#intro-atag-role)

Relationship to the Web Content Accessibility Guidelines (WCAG) 2.0 与WCAG2.0的关系

The W3C recommendation, [Web Content Accessibility Guidelines (WCAG](http://www.w3.org/WAI/intro/wcag.php)), applies to all web content; UAAG provides additional advice on the application user interface. WCAG针对所有Web内容，UAAG提供应用程序用户界面的附加建议。

Some user agents are used to package web content into non-web-based applications, especially on mobile platforms. If the finished application is used to retrieve, render, and facilitate end-user interaction with web content of the end-users choosing, then the application should be considered a stand-alone user agent. If the finished application only renders a constrained set of content specified by the developer, then the application might not be considered a user agent. In both cases, the WCAG 2.0 Guidelines apply to the web content. If the application is not a user agent, application developers are not responsible for UAAG 2.0 requirements that extend beyond WCAG 2.0 requirements. For more detail, see the definition of [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent).

The Role of User Agents in Web Authoring

The following is a list of several ways in which user agents are commonly involved in web content authoring and the relationship between UAAG 2.0 and ATAG 2.0.

1. **Preview tool**: Authors often preview their work in user agents to test how the content will be appear and operate. ATAG 2.0 includes a special exception when previews are implemented with pre-existing user agents, so there are no additional requirements on user agent developers in this case.
2. **Checking tool**: Authors often make use of user agent error panels (e.g. HTML validity, JavaScript errors) during authoring. ATAG 2.0 Part A applies, but may not include additional accessibility requirements beyond the UAAG 2.0 success criteria. If a user agent includes an "accessibility checker", the developer should consult checker implementation guidance in ATAG 2.0 Part B.
3. **Edit modes**: Some user agents include a mode where the user can edit and save changes to the web content, modifying the experience of other users. In this mode, the user agent is acting as an authoring tool and all of ATAG 2.0 applies.
4. **Automatic content changes**: Some user agents or plug-ins may automatically change retrieved web content before it's rendered. This functionality is not considered an authoring tool because changes are made to the user's own experience, not the experience of other users.
5. **Providing a platform for web-based authoring tools**: Many web applications serve as authoring tools and make use of user agent features to deliver functionality (e.g. undo text entry, adjust font size of the authoring tool user interface etc.) User agent developers should consult ATAG 2.0 to understand the ways in which web-based authoring tools can depend on user agent features.

UAAG 2.0 Guidelines

UAAG 2.0 Conformance Applicability Notes:

1. **Recognized Content Only:**UAAG 2.0 success criteria only apply to web content and its behaviors that can be [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) by user agents. 仅针对可识别内容：UAAG 2.0 成功标准仅针对web内容，以及其能够被用户代理识别的行为。
2. **Optional Settings:** Throughout UAAG 2.0, all required behaviors may be provided as optional preference settings unless a success criterion explicitly says otherwise. For example, if a success criteria requires high contrast between foreground text and its background, the user agent may also provide choices with low contrast. A required behavior does not need to be the default option unless the success criteria explicitly says otherwise. 可选的设置：通过UAAG 2.0，所有需要的行为可能提供作为可选的设置，除非成功标准明确表示例外。举例说明，如果一个成功标准要求前景文字和背景色之间的高对度，用户代理可能也提供低对度。一个必要的
3. **RFC 2119 language not used:**UAAG 2.0 does not use RFC 2119 language (must, may, should) because these are guidelines and not interoperable specifications. These words in UAAG 2.0 don't have the same sense as they do in RFC 2119.不使用RFC 2119语言：UAAG 2.0不使用RFC 2119语言（必须、可能、应该）因为这些是指导方针，没有互操作特性。
4. **Simultaneous satisfaction of success criteria:**Users can access all behaviors required by UAAG 2.0 at the same time (e.g. when the user resizes the viewport per 1.8.9, content is reflowed per 1.8.6), except where those behaviors are mutually exclusive.

**PRINCIPLE 1 - Ensure that the user interface and rendered content are perceivable 确认用户界面和呈现的内容是可以被接受的。**

Implementing Guideline 1.1 - Provide access to alternative content [[Guideline 1.1](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-access-alternative-content)] 实施指南 1.1 - 提供替代内容

###### Summary: The user can choose to render any type of alternative content available. (1.1.1). The user can also choose at least one alternative such as alt text to be always displayed (1.1.3), but it's recommended that users also be able to specify a cascade (1.1.5), such as alt text if it's there, otherwise longdesc, otherwise filename, etc. It's recommended that the user can configure the caption text and that text or sign language alternative cannot obscure the video or the controls (1.1.4). The user can configure the size and position of media alternatives (1.1.6).

**1.1.1 Render Alternative Content: 1.1.1 呈现替代内容**

The user can choose to render any type of [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) that is present for a content [element](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element).(Level A) **用户可以选择呈现任何类型的可识别的为内容元素展现的替代内容**

* *Note*: It is recommended that the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) allow the user to choose whether the alternative content replaces or supplements the original content element. 建议让用户代理允许用户去选择是否替代内容替换或者补充原始内容元素。

[Return to 1.1.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_111)

**Intent of Success Criterion 1.1.1:**

Users with some disabilities may find a specific content element causes them physical pain (e.g. an image with high contrast) or distress (e.g. an image that triggers post traumatic stress disorder), or that the element's size may make the page difficult to use (because of difficulty scrolling, or shifting gaze, or moving the pointer more than a certain distance). In these cases the user needs to be able to hide that element or replace it with alternative content or a placeholder. Other users may find specific elements are simply unusable (e.g. an image that is too low contrast for the user's vision). In these cases the user needs to be able to access author-provided alternative content (e.g. Alt text or Longdesc) or the very minimum a placeholder (e.g. filename).

Some users with disabilities need alternate languages or audio tracks (e.g. descriptive video). Users need the ability to choose tracks that best meet their accessibility needs (e.g. the caption track in their own language) when authors have provided many alternatives.

*Note:* If the user cannot directly select or choose the element in order to perform commands upon it (e.g. if the browser does not support clicking on HTML background images or moving focus to them with the keyboard), the user agent must provide an alternative user interface for this feature.

**Examples for Success Criterion 1.1.1:**

* Rodney has low vision and finds some images on a web page are painful to look at. He can hide foreground images by right-clicking on them or pressing the Tab key to move focus to them, and then choosing a command from the element's context menu. However, the browser does not let him right-click or navigate to the background image. Therefore it provides a user preference setting to hide all background images or to hide the background image on the current site.
* Carly has repetitive strain injuries and uses speech recognition to communicate with her computer. She's a paralegal and often has to search long documents for terms that are rendered as acronyms. It's often easier for her to speak full terms into the search box rather than acronyms because full terms are easier to say and better recognized, and this allows her to keep her vocabulary less cluttered by acronyms that may increase misrecognitions. She chooses a setting on her browser that automatically expands all acronyms and abbreviations before she does a search.
* Laura has impaired vision and uses the shapes of words to help her recognize them easily. It is more difficult for her to read strings of seemingly random upper case letters than strings of familiar words written with only their initial letters capitalized, so she changes her preference settings to have all abbreviations replaced by their expanded strings.
* Betty has an auditory processing problem. She is watching a movie that has several caption and audio tracks. She pulls up a menu to determine which tracks are available and switches between audio tracks until she find one she can understand. She also reinforces the audio by selecting a caption track.
* Marka is blind. She is listening to a video of a lecture. The professor is demonstrating a chemistry experiment and is not speaking during a key portion of the video. Since Marka cannot see what the professor is demonstrating, she brings up a menu of the available tracks and discovers that there is an audio description track available. Marka selects the description track, skips back, and listens to the description of the experiment.
* Gorges has profound hearing loss. He enjoys current run movies, and subscribes to a web service that streams major popular movies. Although he speaks English, he doesn't understand all the slang in the movie. He pauses the movie, selects a menu of caption tracks and finds a Spanish caption track. He watches the rest of the movie with Spanish captioning.

**Related Resources for Success Criterion 1.1.1:**

* Firefox extension "[Text Equivalent Expand Abbreviations](http://firefox.cita.illinois.edu/txtequiv/expandabbr.php)"

**1.1.2 Indicate Unrendered Alternative Content: 1.1.2 指明不可呈现的替代内容**

The user can specify that indicators be displayed along with rendered content when [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) unrendered [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) is present. 当可识别的没有呈现的替代内容被展示时，用户可以让指示器连同呈现的内容一起显示(Level A)

[Return to 1.1.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_112)

**Intent of Success Criterion 1.1.2:**

Users need to be able to easily discover when authors have provided alternative web content that may be of interest so they can decide whether to have it rendered (see [Success Criterion 1.1.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_111)). While the type of indicator is not prescribed, the success criterion requires that the indicator be placed along with the rendered content that has the alternative. This rules out indicators such as in the status bar, that don't clearly identify which content within a document has the alternative. Suitable indicators include outlines, adjacent icons, and adjacent links. As with any other feature of the user agent, the indicator itself must be accessible (e.g. keyboard accessible, alternative text, zoomable).

**Examples for Success Criterion 1.1.2:**

* Chris is color blind. He doesn't use a screen reader or other assistive technologies, but sometimes has difficulty interpreting diagrams where the author has used color alone to convey meaning (e.g. to label lines). Therefore, Chris likes to read long descriptions when the author has provided them. To let himself know when long descriptions are available, Chris sets a setting in his browser that places a "Description" link adjacent to any image with a long description. Chris clicks the link the open the long description in a new browser pane.
* Aosa is blind. When rendering a web page using synthesized speech, the user agent generates an audible tone to signify that the word being read is an acronym, and Aosa can press the \* key to hear the expansion. When the phrase being read is the Alt text for an image, another tone indicates that Aosa can press + to hear the long description. 盲障用户在使用读屏技术阅读网页时，当碰到一个缩写的时候，按\*去了解缩写的扩展意思，当碰到一个图像时，按+听到该图像的替代内容
* Brin is deaf. The video player she is using has a button displayed beneath the playing video that indicates that captions are available. She clicks the button to toggle the captions on so she can understand the video. On her mobile phone, Brin touches a video, which displays the controls including the "display caption" control.

**Related Resources for Success Criterion 1.1.2:**

* "HTML5 Image Description Extension (longdesc)" (http://www.w3.org/TR/html-longdesc/#longdesc)
* "User Agents and longdesc Discoverability" (http://www.d.umn.edu/~lcarlson/research/ld-ua.html)

**1.1.3 Replace Non-Text Content: 1.1.3 替代非文本内容**

The user can request a placeholder that incorporates [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) text alternative content instead of recognized non-text content, until explicit user request to render the non-text content. 用户能够要求一个占位符合并可识别的文本替代内容，用以替代可识别的非文本内容，除非用户明确要求呈现非文本内容(Level A)

[Return to 1.1.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_113)

**Intent of Success Criterion 1.1.3:**

Users may wish to hide images for a number of different reasons. Some users with cognitive disabilities may wish to hide images in order to avoid those that would be severely distracting. Some users with visual disabilities may wish to hide images in order to avoid those that are painful (such as those with high contrast). Other users may wish to replace images with alternative content because they are unlikely to be able to visually discern, understand, or otherwise benefit from the images. Some users with impaired motion or dexterity may wish to replace images with smaller alternative content to reduce the amount of scrolling they have to do, while some users with attention deficit disorder may wish to do the same thing in order to keep as much information visible on the screen as possible.

**Examples for Success Criterion 1.1.3:**

* Tyler has a learning disability and finds images very distracting. He sets a preference in his browser to always display alternative text instead of an image. 图像对于学习障碍用户非常麻烦，可以通过设置浏览器，可以直接浏览图像的替代文字。
* Ben has low vision and needs to use a very large font size to be able to read text. On his mobile device, enlarging the page makes any images so large that they use up too much screen space and require excessive scrolling. He sets a preference to render all images as text (if available) and to reflow the page so that the text flows smoothly with no space for the missing images.
* Lee has low vision where text is blurred. She finds that the transparency effect that shows detail "bleed through' under the window she is reading makes the text she is trying to read illegible. She sets a preference that hides all background images and transparency effects that the author may have written into a web page or application.
* James has a reading disability where he needs text to be clear from distractions that are not related to the text. He configures his user agent to not load background images. Then when he navigates to a web page, James gets only the text from the web page without any images interfering with what he is reading.
* Sasha requires high contrast to be able to discriminate the shape of letters. She always sets a preference in her browser to turn off background images, so that she can see the text clearly without the variations in the background.
* Betty is a low vision user and has difficulty reading text on her mobile device when it is displayed over a background image. Using her user-defined style sheet, she can disable all background images from being rendered in her browser.

**Related Resources for Success Criterion 1.1.3:**

* [1.4.1 Configure Rendered Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_141) for more information related to background colors.
* [1.1.4 Configure Alternative Defaults](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_113) allows the user to hide foreground images by replacing them with alternative content, but 2.11.1 extends this to background images in languages such as HTML that do not support alternative content for background images.
* [CSS 2.1: section 14.2.1 The Background](http://www.w3.org/TR/2011/REC-CSS2-20110607/colors.html#background) In CSS, background images may be turned on/off with the background and background-image properties.
* [CSS 2.1: section 9.9.1 Specifying the stack level: the 'z-index' property](http://www.w3.org/TR/2011/REC-CSS2-20110607/visuren.html#z-index)

**1.1.4 Provide Configurable Alternative Content Defaults: 1.1.4提供可配置的替代内容的默认配置：**

The user can specify which type(s) of [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) to render by default for each type of non-text content, including time based media.用户可以指定，为每种非文本内容提供替代内容的默认呈现，包括时基媒体 (Level AA)

[Return to 1.1.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_114)

**Intent of Success Criterion 1.1.4:**

Alternative content is wasted if the user agent doesn't render it for users who need it. Default alternative content is a global setting because it is an unreasonable burden for users to change the rendering options every time they visit a new page. 如果用户代理不能为需要它的用户呈现替代内容，替代内容就被浪费掉。默认的替代内容应是一个全局设定，因为如果每次访问新页面都需要重新改变呈现选择项，对于用户是一个不合理的负担。

**Examples for Success Criterion 1.1.4:**

* Sally has low vision. In the browser's preferences dialog box, Sally specifies that she wants alt text displayed in place of images, and that the document should reflow to allow the entire alt text to be displayed rather than truncated. 低视力用户可以在浏览器中设定替代文本内容显示在图像位置中，那么文本应该可以允许重构显示所有替代文字而不是截取文字部分内容。
* Ben has low vision. In the browser's preferences dialog box, he chooses to always display the alternative ("fallback") content for embedded objects, such as a transcript in place of a video. 低视力用户可以在浏览器中设定总是显示嵌入目标的替代，比如某个视频的文字记录
* Ben has low vision and keeps his mobile phone browser "zoomed" so he can read the text. Because images can become pixelated when enlarged, he prefers the alternative text. In the mobile settings dialog box, he chooses to always display the alternative ("fallback") content for images and to reflow the page without a placeholder for the image. This saves screen space and reduces the amount of scrolling he has to do.
* Brin is deaf. She toggles a menu item which turns on the display of all captions for video and audio content.

**Related Resources for Success Criterion 1.1.4:**

* Success criterion 1.1.4 adds additional required options to success criterion [1.1.3 Replace Non-Text Content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_113).

**1.1.5 Facilitate Clear Display of Alternative Content for Time-based Media: 1.1.5 促进时基媒体替代内容的清晰显示**

For [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) on-screen alternative content for time-based media (e.g. captions, sign language video), the following are all true: 可识别的时基媒体的替代内容 （例如，字幕，手语视频，如下情况应为真(Level AA)

* **Don't obscure controls:** Displaying time-based media alternatives doesn't [obscure](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-obscure) recognized controls for the primary time-based media. 不要掩盖控件：为主要时基媒体提供体替代内容不要掩盖识别控件
* **Don't obscure primary media:** The user can specify that displaying time-based media alternatives doesn't obscure the primary time-based media. 不要掩盖主媒体：用户可以指明显示时基媒体的替代不掩盖主要的时基媒体。
* **Use configurable text:** The user can configure recognized text within time-based media alternatives (e.g. captions) in conformance with[1.4.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_141). 用可配置的文本：用户可以配置时基媒体的替代（例如：字幕）中可识别的文本应遵守1.4.1
* *Note*: Depending on the screen area available, the display of the primary time-based media may need to be reduced in size to meet this requirement. 注意：依据屏幕区域的可用性，主要时基媒体的显示可能需要去减小尺寸以符合此需求。

[Return to 1.1.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_115)

**Intent of Success Criterion 1.1.5:**

Users who require or can benefit from alternative media tracks in video or audio may not find that the default or authored position and size of those tracks is usable. Enabling the user to move and scale any displayed alternate media tracks (e.g. captions) allows displayed content to be positioned and sized to meet the needs of the user.

**Examples for Success Criterion 1.1.5:**

* Justin has low vision and works in a noisy environment that makes it difficult to listen to instructional videos. When he enlarges the text of the captions to a viewable size, they block most of the video image. Justin selects an option that displays the caption track in a separate window, which he positions below the video image so the captions don't block the video image. 噪音环境下，很难听到指示性的音视频，但如果显示放大字体可能导致文字遮挡住视频，有一个选项可以让其选择在单独的视窗显示字幕
* Jaime is deaf and is taking courses from on online university. She prefers to use ASL if it is available for online media. A course she is taking offers captions and a signing avatar for the recorded lectures. The default size of the avatar window is small, making it difficult to follow the signing. The avatar also overlays a significant part of the lecture video. Jaime drags the avatar out of the video and enlarges it, so that they are side by side and equally sized.
* Jaime is deaf and prefers to always display captions on her mobile phone. She has set her global settings on the phone to turn on closed captions. All videos displayed on the phone will automatically display captions.
* Ben has low vision that becomes worse throughout the day as he becomes more tired. He keeps a floating control on his mobile phone that allows one touch access to his configuration so that he can change the font size. The floating control can be easily moved around the screen so it is not in the way of other controls, and it becomes translucent after it is idle for a few seconds.

**Related Resources for Success Criterion 1.1.5:**

* [1.4.1 Configure Rendered Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_141)

**1.1.6 Allow Resize and Reposition of Time-based Media Alternatives: 允许时基媒体的替代内容的调整大小和复位**

The user can configure [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) alternative content for time-based media (e.g. captions, sign language video) as follows: 用户可以为时基媒体配置一个可识别的替代内容（例如， 字幕和手语视频）(Level AAA)

* **Resize:** The user can resize alternative content for time-based media up to the size of the user agent's [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport). 调整大小：用户能够根据用户代理视察的大小调整时基媒体的替代内容大小。
* **Reposition:** The user can reposition alternative content for time-based media to two or more of the following: above, below, to the right, to the left, and overlapping the primary time-based media. 重新定位：用户能够重新定位时基媒体的替代内容，在如下两种或两种以上的情况：上、下、左、右、与主要时基媒体重合。
* *Note 1:* Depending on the screen area available, the display of the primary time-based media may need to be reduced in size or hidden to meet this requirement. 根据屏幕大小，主要媒体的显示可能需要减少尺寸或者隐藏以符合需要。
* *Note 2:* Implementation may involve displaying alternative content for time-based media in a separate viewport, but this is not required. 实施可能需要让时基媒体的替代内容显示在一个独立的视察里，但这不是必需的。

[Return to 1.1.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_116)

**Intent of Success Criterion 1.1.6:**

Users may want to reposition the alternative in close proximity to the most important portion of the main media to reduce the visual scanning distance between them. For example, if the video frequently includes on-screen text near the top of the video then the captions will be easier to read if they are located above the video.

**Examples for Success Criterion 1.1.6:**

* Maximilian is hard of hearing and prefers to use captions. He adjusts the position of his captions depending on what he's watching. When watching a sporting event with a dashboard displaying statistics at the top, he positions the captions immediately above the top so the captions are close to the statistics. However, when he watches a movie, he positions the captions so that they overlap the video frame near the bottom. When he watches financial news with a stock ticker along the bottom, he moves the captions to be immediately below the ticker.
* Raymond has one functioning hand. He positions captions so that they're not covered by the hand he's using to hold his tablet.
* Tom is deaf. When Tom watches narrow-aspect video on a wide-aspect screen or in landscape mode on his mobile device, he moves the window displaying sign language interpretation to the side, allowing the primary video to take up the entire height of the screen without the interpretation getting in the way.

**Related Resources for Success Criterion 1.1.6:**

* None

Implementing Guideline 1.2 - Repair missing content [[Guideline 1.2](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-missing-alt)] 实施指南 1.2 - 修复缺失内容

**Summary**: The user can request useful alternative content when the author fails to provide it. For example, showing metadata in place of missing or empty (1.2.1) alt text. The user can ask the browser to predict missing structural information, such as field labels, table headings or section headings (1.2.2).

**1.2.1 Support Repair by Assistive Technologies: 1.2.1 支持辅助技术的修复**

If [text alternatives for non-text content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) are missing or empty then both of the following are true: 如果为非文本内容提供的文本内容缺失或为空时，如下两种情况应为真(Level AA)

* The[user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) doesn't attempt to [repair](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-repair-text) the text alternatives by substituting text values that are also available to assistive technologies. 用户代理不尝试通过去修复文本替代值去修复文本替代，这也针对于辅助技术。
* The user agent makes available metadata related to the non-text content available programmatically, but not via fields reserved for text alternatives. 用户代理应让相关的非文本内容的元数据可被程序化使用，而不是通过文本替代的保留字段。

[Return to 1.2.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_121)

**Intent of Success Criterion 1.2.1:**

When alternative content is missing, it can be helpful for users to access metadata such as the filename, which can be substituted as repair text. However, these are not as helpful as alternative content that's properly authored for the original document. In these cases assistive technology can provide users with information that may be more helpful than any one piece of repair text the user agent could provide. Therefore it's important that assistive technology have access to as much information as possible about the non-text content that needs repair, and also to be able to inform the user that no author-provided text alternative is available. User agents should provide assistive technology with available metadata for the non-text content that needs repair, but not substitute repair text in ways assistive technology will mistake for author-provided text alternatives.

**Examples for Success Criterion 1.2.1:**

* Ray is blind and counts on alternative text for images. When his screen reader is reading a web page and encounters an image, it asks the user agent for alternative text. If the user agent reports that no alternative text is available, the screen reader accesses the DOM to retrieve the title attribute associated with the image, its original file name, and path to the downloaded image file. It extracts embedded metadata from the image file, such as its original title and caption fields. It can then tell Ray that there is an image with no alternative text, provide him with the value it considers most likely or that Ray has selected through his preferences, then provide a command that lets him hear the other values so he can make his own judgement about the nature and purpose of the image.

**Related Resources for Success Criterion 1.2.1:**

* None

**1.2.2 Repair Missing Structure: 1.2.2 修复缺失的结构**

 The user can specify whether or not the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) should attempt to insert the following types of structural markup on the basis of author-specified presentation attributes (e.g. position and appearance): 用户可以指定是否用户代理应该尝试在作者指定的展示属性去插入如下内容的结构标记 (Level AAA)

* Labels 标签
* Headers (e.g. heading markup, table headers) 标题 （例如：标题标记、表单标题）

[Return to 1.2.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_122)

**Intent of Success Criterion 1.2.2:**

When an author neglects to provide labels and/or headers as necessary for accessibility, the user agent can sometimes use heuristics to determine potential labels and/or headers from presentation attributes. Once potential headings and/or labels have been identified, the user agent can proceed (e.g. with communication via platform accessibility services) as if the relationship was defined in the markup. The user can specify whether heuristics should be applied because some users want to experience the content as the author provided it (e.g. to perform evaluations or when heuristics fail).

**Examples for Success Criterion 1.2.2:**

* Frank uses speech input. When content markup includes a checkbox without a specified label, the user agent detects that a static text component is positioned immediately to the right of the checkbox and no other static text components are nearby. The nearby text is treated as the label. This enables Frank to toggle the checkbox by speaking its name.
* Maria uses a screen reader. When a table lacks marked up header rows, the user agent gives her the option to have the first row treated as the table header row.
* Li uses a scanning keyboard and makes use of the user agent's outline view to more efficiently navigate web pages. The content markup includes instances of static text that differ from surrounding text because they are sentence fragments and styled with a larger font and bold weight. The instances are treated as headings as therefore appear in the outline view, enabling Li to more efficiently navigate to them.

**Related Resources for Success Criterion 1.2.2:**

* [Understanding WCAG 2.0: Heading and Labels](http://www.w3.org/TR/2012/NOTE-UNDERSTANDING-WCAG20-20120103/navigation-mechanisms-descriptive.html)

Implementing Guideline 1.3 - Provide highlighting for selection, keyboard focus, enabled elements, visited links [[Guideline 1.2](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-interaction-highlight)] 实施指南 1.3 - 选择、键盘聚焦、激活元素、访问链接时提供突出显示

**Summary**: The user can visually distinguish selected, focused, and enabled items, and recently visited links (1.3.1), with a choice of highlighting options that at least include foreground and background colors, and border color and thickness (1.3.2).

**1.3.1 Highlighted Items: 1.3.1 突出显示项目**

The user can specify that the following classes be [highlighted](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-highlight) so that each is uniquely distinguished:用户能够指定如下类别被突出显示，以使每种类别都是唯一可区分的 (Level A)

* Selection 选择
* [Active keyboard focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-active-input-focus) (indicated by focus cursors and/or text cursors) 激活键盘焦点 （通过光标焦点并且或者文字光标指示）
* Recognized enabled input elements (distinguished from disabled elements) 可识别的激活的输入元素（与禁用元素区分）
* Recently visited links 最近访问过的链接

[Return to 1.3.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_131)

**Intent of Success Criterion 1.3.1:**

Users need to be able to easily discover web content they can interact with. One effective way to do this is to highlight enabled elements and links (including recently visited links). Highlighted selection and content focus lets people who use keyboard, gesture and speech input know where they are working. On some pages controls may be difficult to discern amid a large amount of other content, or may be styled so the controls are difficult to distinguish from other content. This can be particularly difficult for people with visual impairments, who may not be able to distinguish subtle visual differences. People with some cognitive impairments may have difficulty distinguishing between items with similar or non-standard appearance. Visually distinguishing these items reduces the amount of time or number of commands these groups require to examine a page.

* *Note 1:*In addition to these required categories, it is recommended that user agents also allow the user to highlight the active viewport, even when it is a frame or similar within the active window. This makes it much easier for the user to visually locate the active focus.
* *Note 2:*Platform conventions will dictate whether or not keyboard focus in an inactive viewport is visually indicated by an inactive cursor.
* *Note 3:*the definition of visited and unvisited links is up to the user agent. Visited links may be links visited during the current session or in the browser's history.

**Examples for Success Criterion 1.3.1:**

* Jerry has low vision. He goes to a website that uses Cascading Style Sheets (CSS) to override visited link color. He wants to know what links have yet to be explored. The user agent provides a dialog box for setting overrides to author-selected link colors. Jerry uses the dialog box to override these so visited links are indicated.
* Jerry goes to a website with styles that remove the content focus outline. The user agent provides a dialog box for setting overrides to the author's focus outline declaration. Jerry uses the dialog box to display the content focus outline so he can tell where the focus is on the page.
* Binh gets easily frustrated when he cannot locate the buttons and links on a page. This happens when buttons and links don't have the standard appearance he's used to. The user agent provides a dialog box for setting overrides to author-selected link colors. Binh turns on the option to have all links appear in bright purple, and buttons drawn with a bright purple border so he can easily scan the page and find the items he's looking for.
* George has limited hand use and uses custom gestures on his mobile phone. He wants a visible focus indicator to know what element on the page has focus so when gestures are used on the mobile phone, he will know what element will be activated.
* Tinan has repetitive strain injuries and seeks to limit scrolling. The user agent renders distinct visual icons in proximity of content that has short text alternatives, long descriptions, and captions. If the icon forces the text to extend beyond a fixed size viewport, the user agent uses global preference settings to determine whether to expand the viewport, provide scroll bars, or truncate the content.
* Aosa is blind. When rendering a web page using synthesized speech, the user agent generates an audible tone to signify that the word being read is an acronym, and Aosa can press the \* key to hear the expansion. When the phrase being read is the Alt text for an image, another tone indicates that Aosa can press + to hear the long description.
* Brin is deaf. The video player she is using has a button displayed beneath the playing video that indicates that captions are available. She clicks the button to toggle the captions on so she can understand the video. On her mobile phone, Brin touches a video, which displays the controls including the "display caption" control.

**Related Resources for Success Criterion 1.3.1:**

* [1.3.2 Highlighting Options](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_132) describes user options to configuring how these categories are highlighted
* [2.1.2 Keyboard Focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_212) requires every window have an active or inactive keyboard focus at all times
* [1.8.1 Highlight Viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_181) provides guidance for highlighting viewports
* [1.8.2 Move Viewport to Selection and Focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_182) provides guidance for highlighting an active window

**1.3.2 Highlighting Options: 1.3.2 突出显示可选择性：**

When highlighting classes specified by [1.3.1 Highlighted Items](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_131), the user can specify highlighting options that include at least:当突出显示的类别被指定 1.3.1突出显示项中的内容，用户可以指定突出显示的选项，至少包括以下(Level AA)

* Foreground colors 前景色
* Background colors 背景色
* Borders (color, style, and thickness) 边框 （颜色、样式和密度）
* Size when the indicator is an image 当一个指示器是一个图像，它应该实现比例缩放
* Blink rate (where implemented) 闪动频率 （实现）

[Return to 1.3.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_132)

**Intent of Success Criterion 1.3.2:**

Low vision users and users with some cognitive disabilities need control over visual properties to meet their individual needs. These include foreground colors, background colors, and visual borders (with the same configurable range as the operating environment's conventional selection utilities)

**Examples for Success Criterion 1.3.2:**

* Alex has low vision. He sometimes has difficulty distinguishing fields on web forms. The user agent provides a dialog box allowing the user to override any author settings. He chooses to have all form fields displayed with a yellow background and outlined with a thick black border.
* Marcy has a cognitive disorder that makes it difficult to stay focused on the task she wants to accomplish. The user agent provides a dialog box allowing the user to override any author settings. She chooses to have all form fields displayed with a yellow background and outlined with a thick black border.

**Related Resources for Success Criterion 1.3.2:**

* [CSS 2 specification](http://www.w3.org/TR/CSS21/box.html#propdef-border-style) – box border styles

Implementing Guideline 1.4 - Provide text configuration [[Guideline 1.4](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-text-config)] 实施指南 1.4 - 提供文本配置

**Summary:** The user can set text scale, color, and font family globally (1.4.1, Level A); set text size, color, and font family for element types (1.4.2, Level AA); set line spacing, character spacing, word spacing, text style, and justification globally (1.4.3, Level AA); set text style, margins, and borders for elements (1.4.5, Level AAA); set line spacing, capitalization, hyphenation, margins, and borders globally (1.4.6, Level AAA); and print configured and reflowed text (1.4.4 Level AA).

*Note 1:*Success criteria 1.4.1, 1.4.3, and 1.4.6 address configuration at a **global** level, that is, it changes all of the text. Success criteria 1.4.2 and 1.4.5 are at an **element type** level, such as configuring just the heading text. 成功标准1.4.1，1.4.3和1.4.6

*Note 2*: All of the success criteria under guideline 1.4 allow users to override the text characteristics specified by authors, and override user agent defaults.

*Note 3*: The success criteria in guideline 1.4 can be met through user style sheets. For platforms without user style sheets, text configuration needs to be provide to users through the user agent's main user interface.

**1.4.1 Text Scale, Color, Font (Globally): 文本比例、颜色、字体 （全局地）：**

The user can [globally](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-global-configuration) set all of the following characteristics of visually [rendered text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-text) content: 用户可以全局地配置所有以下可视觉呈现的文本内容(Level A)

* Text scale with preserved size distinctions (e.g. keeping headings proportional to main font)
* Text color and background color, choosing from all platform color options 可以从所有平台中选择文本颜色、背景色。
* Font family, choosing from all platform fonts 可以从所有平台字体中选择字体族群。

[Return to 1.4.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_141)

**Intent of Success Criterion 1.4.1:**

Users with some types of disabilities have difficulty reading text as the author has formatted it. For example, some users with low vision, dyslexia, and related conditions and situations cannot read author-formatted text. However, they can read text that has a format customized to their individual needs (e.g. larger letters, different font, more spacing). Users with some cognitive disabilities and those who need to reduce the number of commands they enter to scroll through a document, may need to fit more information on the screen (e.g. smaller letters or spacing). Users need to access a wide range of font sizes, styles, colors, and other attributes in order to find the combination that works best for their needs. In providing preferences, it is important to avoid making assumptions. For example, some users may increase font size to make text more legible, while other users may reduce the font size to decrease the need to scroll the content.

The relative size of text provides visual cues that help in understanding and navigating web content. Some content may be authored in a way that makes it difficult or impossible to understand when font distinctions are hidden, such as headlines that are in not in a larger font than body text. It's important that users who need to enlarge or reduce text size be able to preserve these visual cues.

**Examples for Success Criterion 1.4.1:**

* Lee has low vision from albinism and when the background is white, it is hard for her to read and hurts her eyes. She changes the colors in her operating system to white text on a black background. However, many web pages specify their own colors, so she adjusts the user preferences in her browser to always use the system colors (or another set of colors of her choosing), regardless of how the pages are authored.
* Betty Sue has low vision from nystagmus and has difficulty seeing small fonts. The browser on her laptop provides five font sizes. However, the largest isn't big enough. The browser settings provide an option to override the five font sizes with customized font range, so that Betty Sue can select the specific font size she needs.
* Mike has a reading disability. A website uses a fancy script font for the headings that he cannot understand. He uses his browser's font settings to select plain fonts that he can read.
* Ben has low vision. In the mobile settings dialog box, he chooses a large text for font size. All applications on the mobile phone display text in large font.
* Sebeeya has low vision. She finds text easiest to read at 16 pt Palatino and chooses to have her browser display body text in the 16 pt Palatino font. She needs the headlines to scale proportionally (e.g. 24 pt) in order to preserve headline prominence.

**Related Resources for Success Criterion 1.4.1:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)
* Success Criteria [1.8.5 Allow Zoom](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_185)
* Success Criteria [1.8.7 Reflow Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_187)
* Success Criteria [1.8.9 Allow Viewport Resize](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_189)

**1.4.2 Text Size, Color and Font (by Element): 文字大小、颜色和字体（元素）**

The user can set all of the following characteristics of visually rendered text content for text element types including at least headings and input fields: 用户可以以文本元素类型设置所有以下可视觉呈现的文本内容特征，至少应包括标题栏和输入栏(Level AA)

* Text size 文本尺寸
* Text color and background color, choosing from all platform color options 可以从所有平台选择文本颜色、背景色
* Font family, choosing from all platform fonts 可以从所有平台字体中选择字体族群

[Return to 1.4.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_142)

**Intent of Success Criterion 1.4.2:**

Some users with low vision, dyslexia, and related conditions and situations cannot read normally-formatted text. However, they can read text that has a format customized to their individual needs (e.g. larger letters, different font, more spacing). Users need to access a wide range of font sizes, styles, colors, and other attributes in order to find the combination that works best for their needs. In providing preferences, it is important to avoid making assumptions. For example, some users may increase font size to make text more legible, while other users may reduce the font size to decrease the need to scroll the content.

Users who need large amount of screen magnification need to control the appearance of types of elements. For instance, a magnification of 300% can make headlines too large for display. Users need to be able to set the characteristics of element types (e.g. heading 1, heading 2, table heading) to make the web content readable. Magnification users who find that text size distinction greatly increases scrolling and fatigue also need to be able to display important elements such as headings (including table headings) and input fields independently from global settings.

**Examples for Success Criterion 1.4.2:**

* Tomas has low vision and uses a screen magnifier. He chooses to have his browser display all text the same size, and sets that size as large as he can without making the letters too tall for his screen. Tomas chooses not to have headings be proportionately larger than normal text because that would make them taller than his screen and so unreadable. He uses a user stylesheet to make the different levels of headings a different color and surrounded with the number of equals signs (=) that matches of heading level.
* Adele is 84 and has macular degeneration and is experiencing fine motor problems. She loves using her mobile tablet to read her email and follow her grandchildren on social media. She has difficulty selecting text areas to enter forms. Her granddaughter helps her create a font size setting for input fields so that the text is larger and increases the area where she can tap, making it easier to enter text.

**Related Resources for Success Criterion 1.4.2:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)
* Success Criteria [1.8.5 Allow Zoom](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_185)
* Success Criteria [1.8.7 Reflow Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_187)
* Success Criteria [1.8.9 Allow Viewport Resize](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_189)

**1.4.3 Text Spacing and Style (Globally): 文本间隔和样式 （全局）**

The user can [globally](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-global-configuration) set all of the following characteristics of visually rendered [blocks of text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-blocks-of-text):用户可以全局的设置如下可视觉呈现的文本块的特性(Level AA)

* Line spacing of at least 1.0, 1.3, 1.5, and 2.0 times the font height 行间距至少是字高的1.0，1.3，1.5，2.0倍
* Character spacing of at least 0.01, 0.03, 0.06, 0.09 times the base character width 字符间距至少是字符宽度0.01，0.03，0.06，0.09倍
* Word spacing of at least 0.01, 0.03, 0.06, 0.09 times the base character width 词间距至少是字符间距的0.01，0.03，0.06，0.09倍
* Text style (underline, italic, bold) 文字类型（下划线、斜体、黑体）
* Justification (left, right, or full) 对齐（左、右、全对齐）

*Note*: For the purposes of UAAG 2.0, the base character width is the font width of the character commonly accepted as the base character for calculating kerning in the typography for that language (e.g. zero character in English). 注意：大意是说字符宽度和语言有关。

**Intent of Success Criterion 1.4.3:**

Some users with low vision, dyslexia, and related conditions and situations cannot read normally-formatted text. However, they can read text that has a format customized to their individual needs (e.g. larger letters, different font, more spacing). In providing preferences, it is important to avoid making assumptions. For example, some users may increase font size to make text more legible, while other users may increase font spacing to improve readability.

Calculating font height and base character width can be complicated. For accessibility purposes, the formula is not important as long as the result gives users multiple choices of font height and character kerning. For UAAG 2.0 purposes, it does not matter what unit of measure is used, as long the multiples of that unit are provided.

Many users with low vision have difficulty reading italic and underline and must be able to globally remove these styles. This can be accomplished with user stylesheets, but if the browser doesn't support user stylesheets, the browser needs to provide this ability through settings.

Many users with autism or reading disabilities have difficulty reading justified text, because uneven spacing can create distracting "rivers of white". These users need to be able to change fully justified text to left or right justified depending on the text language.

**Examples for Success Criterion 1.4.3:**

* Suzanne's vision has deteriorated due to aging and she has had cataract surgery. When she reads blocks of text that have normal spacing, the text becomes a gray mass and Suzanne cannot skim or track from the end of one line to the next. In one browser she uses a toolbar that doubles the line spacing. For another browser, a friend coded a bookmarklet that Suzanne uses to double the line spacing.
* Abdullah has dyslexia and low vision, and has trouble telling letters and words apart. When he increases the character spacing and word spacing he can better distinguish each letter and understand the words. When text is underlined, such as links, it makes it hard for him to read, so he turns off underlining.
* Jamee has dyslexia that makes it hard for her to keep her focus on words. When text is fully justified, there are sometimes lots of space between words and in long paragraphs of text, these create "rivers of white". She sees the spaces and has trouble focusing on the words. When Jamee changes from full justification to left justification, there is normal space between words and she can read much more easily with less fatigue.

**Related Resources for Success Criterion 1.4.3:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)
* Success Criteria [1.8.7 Reflow Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_187)
* CSS 2 [Letter and word spacing](http://www.w3.org/TR/CSS2/text.html#spacing-props)
* CSS 2 [10.8.1 Leading and half-leading](http://www.w3.org/TR/CSS2/visudet.html#leading)

**1.4.4 Configured and Reflowed Text Printing: 配置和重排文本打印**

The user can print any rendered visual, non-time-based content to the user's choice of available printing devices.用户能够打印任何呈现过的、可视化的、非时基内容到用户选择的可打印设备。 (Level AA)

*Note*: The user must be able to print content as it is rendered on screen, reflecting user scaling, highlighting, and other modifications, but reflowable content is reflowed for the print margins. 用户必须能够打印屏幕上被呈现的内容、反映用户的比例缩放、高度显示和其它修改，但是

**Intent of Success Criterion 1.4.4:**

The ability to print content is important for users who have difficulty reading or interacting with Web content directly in the user agent due to software, hardware, or ergonomic issues. Printing to virtual printer devices can also be a necessary step in converting the content to another electronic format that the user finds more accessible. It is also important that the user be able to print content with modifications they have applied, such as scaling or highlighting, or else they may find the printed version unusable. At the same time, they need to be able to have content reflowed to fit the width of the page, or else content may be cut off.

User agents are strongly encouraged to let the user print a portion of the document, such as a selection or specified range of pages, because otherwise printing won't be a practical option for long documents.

**Examples for Success Criterion 1.4.4:**

* Jamee has dyslexia that makes it hard for her to keep her focus on words. If she tries to read much online, she loses her place. Therefore, she often prints text then uses a blank piece of paper to cover up the line below where she is reading.
* Ansgard has multiple sclerosis that causes low vision, fatigue, and muscle spasticity. He cannot sit and read from his computer for very long at a time, so Ansgard prints long articles and reads them in a more comfortable position. He wants to configure the text to be readable and to use as little paper as possible, so he sets large text overall, making headings the same size, then he makes headings bold and underlined to stand out. Ansgard sets overall margins narrow to get more text on a printed page.
* Ralph finds it moderately difficult to use the computer as he ages, so while he can operate the Web browser, he finds long documents much easier to read when he prints them out so that he can hold them in a more comfortable position. When a web page has small text, he enlarges it on the screen using his web browser's Zoom command, which also makes the printed text large enough for him to read.

**Related Resources for Success Criterion 1.4.4:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)
* Success Criteria [1.8.7 Reflow Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_187)

**1.4.5 Text Style, Margins, Borders (by Element): 文本样式、边缘、边框 （元素）**

The user can set all of the following characteristics of visually rendered text content for main text and for text element types including at least headings and input fields:用户可以设置为如下所有可以为主要文本和文本元素视觉呈现的文本内容，至少应包括标题和输入域。 (Level AAA)

* Text style (underline, italic, bold) 文本样式（下划线、斜体字、粗体字）
* Margins (for example, space above headings, indentation of lists) 边际（例如，标题上方的空间，列表的划线）
* Borders 边框

[Return to 1.4.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_145)

**Intent of Success Criterion 1.4.5:**

Some users need to turn off underline, italic and bold because they can be hard to read. Bold characters can "bleed" when characters are too close together which reduces readibility for some. Users who are changing font size need to be able to change margin size to ensure efficient use of screen space. Sometimes when font sizes are increased proportionally, too much white space appears. When borders appear too close to text, it can become difficult for some users to discriminate the text letter shape, so those users need a way to turn borders off. At the same time, increasing the weight of the visible border around an input field makes it more visible, so users need control of the visible characteristics of the borders, for example border color, weight and spacing.

User stylesheets can be used where available. Users can control characteristics by element type, so that, for example, headings at the same level will have a similar appearance.

**Examples for Success Criterion 1.4.5:**

* Wade has low vision and needs large text, extra line spacing, italics changed to regular font, and a muted background color in order to read. To make web pages readable, Wade uses a user stylesheet to change the way the headings are displayed. He makes the headings a little smaller than the main text, changes the font to Arial and red, adds space before and after the headings, and adds border above the headings. This ensures that headings aren't too big, navigation links don't break across lines as much, and a reasonable amount of text is visible at a time, so Wade doesn't have to do extra scrolling. By changing the heading elements to have a different appearance from the main text, Wade can easily distinguish headings without them being them too large.

**Related Resources for Success Criterion 1.4.5:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)

**1.4.6 Spacing, Capitalization and Hyphenation (Globally): 间距、大写、断字 （全局）**

The user can [globally](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-global-configuration) set all of the following characteristics of visually rendered [blocks of text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-blocks-of-text): 用户能够全局的设定所有如下的可视化的、被呈现的文字块的特征 (Level AAA)

* Line spacing between 0.7 and 3.0 times the font height, at increments of 0.10 行间距在字体高度0.7到3.0倍数之间，0.1的增量值。
* Capitalization (overriding upper case and small caps style) 大写
* Word-breaking properties (auto-hyphenation) 断字属性（自动连字符）
* Margins 边际
* Borders 边框

*Note:* This success criteria does not apply to text entered as all caps. Content authors are encouraged to use styles instead of typing text as all caps.

[Return to 1.4.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_146)

**Intent of Success Criterion 1.4.6:**

Advanced features that improve readibility include more granular control of line spacing, override of capitalization and auto-hyphenation, and control of margins and borders. Some users need to turn off capitalization because words that are in all caps can be hard to read. This should not be interpreted as a requirement to provide heuristics to interpret text that has been entered in all caps into appropriate mixed case. The intent is that if all caps has been controlled by a style, the user can override that style.

Wide margins around blocks of text help users focus on the text and not get distracted by surrounding text and images. Some users with low vision prefer to use borders (including border line color, width, style) to indicate headings, rather than making headings larger than the main text.

**Examples for Success Criterion 1.4.6:**

* Lori has low vision. She is using her mobile phone to find a restaurant. The text on the search page is too small. She clicks on the column to put one column width on the screen, then she clicks the Settings button and the font size increase button to incrementally increase the font size to 230%. The headline spacing increases so much that there is not enough text displayed, so Lori selects a Settings button to decrease the space between lines.

**Related Resources for Success Criterion 1.4.6:**

* [1.4.2 Text Size, Color and Font (by Element)](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_142)
* [1.4.3 Text Spacing and Style (Globally)](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_143)
* CSS 2.1 [16.5 Capitalization](http://www.w3.org/TR/CSS2/text.html#caps-prop)
* CSS 3 [5.3. Breaking Rules for Letters](http://www.w3.org/TR/2013/WD-css-text-3-20131010/#word-break-property) (as of this publication, CSS 3 is not a completed W3C Recommendation and could change)
* CSS 2.1 [10.8.1 Leading and half-leading](http://www.w3.org/TR/CSS2/visudet.html#leading)
* CSS 2.1 [16.5 Capitalization](http://www.w3.org/TR/CSS2/text.html#caps-prop)

Implementing Guideline 1.5 - Provide volume configuration [[Guideline 1.5](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-volume-config)] 提供音量配置

**Summary**: The user can adjust the volume of each audio track relative to the global volume level (1.5.1).

**1.5.1 Global Volume: 全局音量**

The user can adjust the volume of each [audio tracks](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-track) independently of other tracks, relative to the global volume level set through [operating environment](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-environment) mechanisms. 用户能够调节每个独立于其它的音轨，通过操作环境机制设定于相对于全局的音量 (Level A)

[Return to 1.5.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_151)

**Intent of Success Criterion 1.5.1:**

User agents can render audio tracks from a variety of sources, and in some cases, multiple audio tracks may be present on a single page. Users should be able to globally set the volume of audio tracks, rather than having to adjust the volume of each audio track being played.

**Examples for Success Criterion 1.5.1:**

* An operating system provides a master audio volume control that applies to all audio tracks rendered within the environment, including the user agent. The user may define a default volume level through a preferences dialog that is retained across sessions. 操作系统提供主要音频音量控制，针对于所有环境内的所有被呈现的音轨。用户能够通过参数对话定义一个默认的音量等级，保留在交叉会话中
* A user encounters a page with two advertisements and one video which begins playback on page load complete. A global mute command, supported via a mute key on the user's keyboard, allows the user to immediately silence the playing audio tracks.

**Related Resources for Success Criterion 1.5.1:**

* None

Implementing Guideline 1.6 - Provide synthesized speech configuration [[Guideline 1.6](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-speech-config)] 提供合成语音配置

**Summary**: If synthesized speech is produced, the user can specify speech rate, volume, and voice (1.6.1, Level A), pitch and pitch range (1.6.2, Level AA), advanced synthesizer speech characteristics such as emphasis (1.6.3, Level AAA) and features such as spelling (1.6.4, Level AAA).

**1.6.1 Speech Rate, Volume, and Voice: 语速、音量、声音**

If synthesized speech is produced, the user can specify the following: 如果产生合成的语音，用户能够定义如下： (Level A)

* Speech rate 语速
* Speech volume (independently of other sources of [audio](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio)) 音量（独立于其它的音频源）
* Voice, when more than one voice is available 声音，当有一种以上的声音

[Return to 1.6.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_161)

**1.6.2 Speech Pitch and Range: 语音音高和范围**

If synthesized speech is produced, the user can specify the following if offered by the speech synthesizer: 如有合成的语音产生，用户能够指明如下要素是否被语音合成器指供(Level AA)

* Pitch (average frequency of the speaking voice) 音域 （说话声音的平均频率）
* Pitch range (variation in average frequency) 音域范围（平均频率的变化）
* *Note:* Because the technical implementations of text to speech engines vary (e.g. formant-based synthesis, concatenative synthesis), a specific engine may not support varying pitch or pitch range. A user agent should expose the availability of pitch and pitch range control if the currently selected or installed text to speech engine offers this capability.

[Return to 1.6.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_162)

**1.6.3 Advanced Speech Characteristics: 高级语音特征**

The user can adjust all of the speech characteristics provided by the speech synthesizer.用户能够通过语音合成器调节所有的语音特点。 (Level AAA)

[Return to 1.6.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_163)

**Intent of Success Criteria 1.6.1, 1.6.2, and 1.6.3:**

These success criteria allow users to control speech characteristics so they can perceive and understand the audio information.

For example, a user may need to increase the volume to a level within the user's range of perception. Or a user may increase the rate of synthesized speech presentation because the user understands it at a rate faster than the default setting of the user agent.

Success criterion 1.6.1 covers characteristics that users most commonly need to adjust and that are adjustable in most technologies. Success criterion 1.6.2 covers characteristics that are less widely altered and less widely supported.

**Examples for Success Criteria 1.6.1, 1.6.2, and 1.6.3:**

* Jamie is blind. He uses a mobile-based web browser to read a web page. He presses a key to increase the rate at which the information is read back. He also uses a mobile browser in a noisy environments such as a crowded subway. With a key press, Jamie quickly increases the volume.
* Randy has a hearing disability where speech at lower pitches is difficult to hear. He is using an audio browser that reads web pages back to him. He issues a voice command saying "raise pitch" and the overall pitch of the synthetic speech is raised.

**Related Resources for Success Criteria 1.6.1, 1.6.2, and 1.6.3:**

* None

**1.6.4 Synthesized Speech Features: 合成语音特征**

If synthesized speech is produced, the following features are provided: 如果产生的是合成语音，如下特性应被提供 (Level AA)

* **User-defined extensions** to the synthesized speech dictionary. 用户定义扩展的合成语音字典。
* **"Spell-out":** text is spelled one character at a time, or according to language-dependent pronunciation rules. 文本每次被拼读成一个字符，或者根据语言的发音规则
* **At least two ways of speaking numerals:**spoken as individual digits and punctuation (e.g. "one two zero three point five" for 1203.5 or "one comma two zero three point five" for 1,203.5), and spoken as full numbers are spoken (e.g. "one thousand, two hundred and three point five" for 1203.5). 至少两种方式去读出数字。
* **At least two ways of speaking punctuation:** spoken literally, and with punctuation understood from speech characteristics like pauses. 至少两种方式去发音：照字面意思朗读，也能根据可理解的发音特点朗读

[Return to 1.6.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_164)

**Intent of Success Criterion 1.6.4:**

The synthetic speech presentation of text can be difficult to understand. These success criteria improve understandability by giving the user the ability to adjust the way the speech synthesizer presents text.

**Examples for Success Criterion 1.6.4:**

* Penny has a reading disability. She is using a browser that reads web pages to her so she can review her most recent banking transactions. She has configured her browser to speak currency instead of digits. With this setting she hears a transaction as "Deposit, two hundred fifty five dollars".
* Penny's speech synthesizer is repeating a phone number. She wishes to copy this number, so she switches to the mode where each digit is spoken as a unique word (e.g. five, five, five, seven, nine).
* George is blind. His speech synthesizer incorrectly pronounces technical terms employed in his organization. The terms are consistently mispronounced in a way that makes it difficult for George to distinguish them. A dictionary allows George to enter a spelling of the name that causes the synthetic speech to produce the correct pronunciation.

**Related Resources for Success Criterion 1.6.4:**

* None

**1.6.5 Synthesized Speech Language:  合成语音语言**

If synthesized speech is produced and more than one language is available, the user can change the language. 如果产生的是合成语音，并且是一种以上的语言可用，那么用户可以改变语言设置。(Level AA)

[Return to 1.6.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_165)

**Intent of Success Criterion 1.6.5:**

Synthesized speech users who are multi-lingual need to be able to respond to changing content by quickly changing the language of the speech synthesizer, overriding any values specified by the author or inferences made by the user agent. Much web content lacks the appropriate language indication or has an incorrect language attribute, and the user agent may not be able to accurately determine the language from the text, so that the user needs a convenient mechanism to change the language.

**Examples for Success Criterion 1.6.5:**

* Hosea is blind. He speaks Spanish but his instructors only speak English. Hosea keeps a floating control on his mobile device that allows one-touch access to his configuration so he can quickly change the language the speech synthesizer reads. He is reading class-related material on the internet in Spanish, but must refer to an explanatory reference link in English. Because the reference link isn't properly coded with a language attribute, his speech synthesizer doesn't recognize the language change. Hosea uses the floating control to quickly switch to English for the reference, then back to Spanish when he returns to the main article he was reading.

**Related Resources for Success Criterion 1.6.5:**

* [W3C Internationalization resources](http://www.w3.org/International/resources)

Implementing Guideline 1.7 - Enable configuration of user stylesheets [[Guideline 1.7](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-style-sheets-config)] 启用用户样式配置

**Summary**: The user agent supports user stylesheets (1.7.1, Level A), the user can choose which if any user-supplied (1.7.2, Level A) and author-supplied (1.7.3, Level A) stylesheets to use, and the user can save stylesheets (1.7.4, Level AA).

**1.7.1 Support User Stylesheets:支持用户样式表**

If the user agent supports a mechanism for author [stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/" \l "def-style-profile" \o "definition: Stylesheet), the user agent also provides a mechanism for user stylesheets.如果用户代理作者样式表提供一种机制，那用户代理也应为用户样式表提供一种机制(Level A)

[Return to 1.7.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_171)

**1.7.2 Apply User Stylesheets: 适用用户样式表**

If [user stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-stylesheet) are supported, then the user can enable or disable user stylesheets for: 如果支持用户样式表，用户可以启用或禁用用户样式表，为 (Level A)

* All pages on specified websites, or 指定网站的所有页面，或者
* All pages 所有页面

[Return to 1.7.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_172)

**1.7.3 Disable** **Author Stylesheets: 禁用作者样式表**

If the user agent supports a mechanism for author [stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/" \l "def-style-profile" \o "definition: Stylesheet), the user can disable the use of [author stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author-stylesheet) on the current page. 如果用户代理支持一种机制使用作者样式表，用户能够在当前页面禁用作者样式表。(Level A)

[Return to 1.7.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_173)

**Intent of Success Criteria 1.7.1, 1.7.2, and 1.7.3:**

CSS stylesheets allow users and authors to customize the rendering of web content. Such customization is frequently used to make web content accessible to a wide range of user needs. These success criteria ensure that users can take full advantage of this ability to customize stylesheets. Since different websites may require different style changes to be readable, it is recommended that user agents provide a feature that lets the user specify which stylesheet should be automatically applied to different web pages as they are loaded (e.g. based on a list of domain names or URL templates).

**Examples for Success Criteria 1.7.1, 1.7.2, and 1.7.3:**

* Tanya has low vision. She finds yellow text on a black background easiest to read. When a website loads, the user agent provides a menu that allows Tanya to select among several stylesheets that the web author has created for the website. Tanya selects the "yellow on black" stylesheet from the menu. The web content is then rendered using this stylesheet.
* Tanya has changed a website that is normally in full color to yellow text on a black background. When her husband Jeromy sits at her computer to look at the site, he goes to the user agent menu to quickly de-select the user-defined stylesheet that Tanya previously applied to the web page. The website is now rendered in full color.
* Lee has low vision and finds text easiest to read on her mobile device when it is presented in yellow on a black background. She has configured her browser to override the author stylesheets to always display text in her browser using this color scheme.
* Mattias has attention deficit hyperactivity disorder (ADHD) and finds text easiest to read if text is highlighted in blue as it is being read out loud on his desktop or mobile device. Both the highlight and text color are configurable and override the author stylesheets so text is readable and has sufficient color contrast.

**Related Resources for Success Criteria 1.7.1, 1.7.2 and 1.7.3:**

* [CSS 2 Recommendation: cascade weight rules](http://www.w3.org/TR/CSS2/cascade.html#cascade)
* [Custom Styles for iOS](http://simplyaccessible.com/article/custom-styles-for-ios/)
* [Stylish for FF and Chrome](http://userstyles.org/)
* [Web Developer](http://chrispederick.com/work/web-developer/)for Firefox and Chrome

**1.7.4 Save Copies of Stylesheets: 保存样式表的副本**

The user can save copies of the stylesheets referenced by the current page. This allows the user to edit and load the copies as user [stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/" \l "def-stylesheet" \o "definition: Stylesheet). 用户能够保存当前页面的样式表的副本，允许用户去修改和加载此副本作为用户样式表。(Level AA)

[Return to 1.7.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_174)

**Intent of Success Criterion 1.7.4:**

Stylesheets provide for powerful customization of rendered content. Occasionally a user may need to make slight modifications to the author-supplied external stylesheets used on a website to satisfy certain accessibility needs. At other times a web author may have created a stylesheet that a user with a disability finds helpful. The intent of this success criteria is to allow users to easily save the CSS for a website and make needed modifications without having to create full stylesheet of their own and to apply well designed stylesheets to other web pages where they find the stylesheets helpful.

**Examples for Success Criterion 1.7.4:**

* Mikki browses to a new website and discovers that the Arial 14-point type used for all headlines does not work well with her level of vision. A friend helps Mikki discover that the headlines are created with a CSS stylesheet. They save this stylesheet and study how it is created. Mikki modifies this stylesheet to adjust the headline text to a larger font and a different typeface. She uses the modified stylesheet by applying it as a user style sheet.
* Tanya has low vision. She browses to a new website on her mobile phone and finds that the site is not optimized for mobile devices. She alters the stylesheet to provide better layout and larger fonts. The custom settings for the stylesheet are saved and applied when she returns.

**Related Resources for Success Criterion 1.7.4:**

* None

Implementing Guideline 1.8 - Help users to orient within, and control, windows and viewports [[Guideline 1.8](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-viewport-orient)] 帮助用户确定和控制窗口和视窗内的行为

**Summary**: The user agent provides programmatic and visual cues to keep the user oriented. These include highlighting the viewport (1.8.1, Level A) and customizing the highlighting attributes (1.8.8, Level AA), keeping the focus within the viewport (1.8.2 & 1.8.6, Level A), resizing the viewport (1.8.9, Level A), providing scrollbars that identify when content is outside the visible region (1.8.3, Level A) and which portion is visible (1.8.4, Level A), changing the size of graphical content with zoom (1.8.5, Level A & 1.8.7, Level A), and restoring the focus and point of regard when the user returns to a previously viewed page (1.8.10, Level AA). The user can specify that all viewports have the same user interface elements (1.8.13, Level AA), if and how new viewports open (1.8.11, Level AA), and whether the new viewport automatically gets focus (1.8.12, Level AA). The user can mark items in a webpage and use shortcuts to navigate back to marked items. (1.8.14, Level AAA).

**1.8.1 Highlight Viewport: 高亮显示视窗**

The user can have the viewport with the input focus be highlighted. 用户可以让输入焦点的视窗高亮显示(Level A)

[Return to 1.8.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_181)

**Intent of Success Criterion 1.8.1:**

When a user agent presents content using multiple viewports, users benefit from a clear indication of which viewport has focus. Text foreground and background colors may not provide enough indication of viewport focus for users with low vision.

**Examples for Success Criterion 1.8.1:**

* Tanya has low vision. Her favorite music website allows her to select which of the top 10 songs are available for listening. Each song is represented by a thumbnail of the album cover – a graphical viewport containing a music player. Tanya uses a keyboard-based screen magnification tool to tab between songs. This highlights the currently selected player with a thick, yellow border against a dark gray background.

**Related Resources for Success Criterion 1.8.1:**

* [1.8.8 Customize Viewport Highlighting](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_188)

**1.8.2 Move Viewport to Selection and Focus: 移动视窗作为选择点和聚集点**

When a viewport's [selection](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-selection) or [input focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-input-focus) changes, the [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport)'s content moves as necessary to ensure that the new selection or input focus location is at least partially in the visible portion of the viewport. 当一个视窗的选择或输入焦点发生变化，视窗内容移动有必要确保新选择或输入焦点位置，至少是在可见窗口部分。 (Level A)

[Return to 1.8.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_182)

**Intent of Success Criterion 1.8.2:**

When content extends horizontally or vertically beyond the visible bounds of its viewport, users must be able to move to one or more selectable elements that may be out of view and to have the selected content automatically move into view. This gives keyboard users and screen magnification users an efficient means to view selected content without having to scroll to locate and view the selection.

**Examples for Success Criterion 1.8.2:**

* John has low vision and uses a screen magnifier. He is spellchecking his blog, which is contained within a scrollable viewport. The blog text exceeds the vertical size of the viewport. The blogging software provides a command to move to the first, and then any subsequent, unrecognized words. With two unrecognized words in the posting, John ignores the first selected word, and presses the key to move to the next unrecognized word, which is out of view. As the key is pressed, the viewport scrolls to show the selected word. 低视力用户用屏幕放大器，当在给他的博客作拼写检查，博客的文本超出视窗窗体，有两个词拼写有问题，当每一个词修改完成后，第二个词应该随着视窗滚动到下方。
* George uses a screen reader. He is showing a sighted colleague how to complete a registration form that's contained within a viewport. The form exceeds the vertical bounds of the viewport, requiring George to scroll vertically to view the complete form content. When George completes each form entry, if the next form is not already visible in the viewport, it scrolls into view.
* Lee typically views web content on her mobile phone at a high level of zoom, frequently positioning elements outside the viewport. When moving between focusable elements, the viewport automatically scrolls to the element currently in focus.

**Related Resources for Success Criterion 1.8.2:**

* [1.8.6 Maintain point of regard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_186)

**1.8.3 Provide Viewport Scrollbars: 提供视窗滚动条**

 When the rendered content extends beyond the viewport dimensions, users can have graphical viewports include scrollbars, [overriding](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-override) any values specified by the author.当呈现的内容超出视窗的规模，用户可以有可视化的图形视窗，覆盖任何被作者指定的值，包括滚动条 (Level A)

[Return to 1.8.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_183)

**Intent of Success Criterion 1.8.3:**

When rendered content exceeds the bounds of a graphical viewport, horizontal or vertical scrollbars show that not all of the rendered content is currently visible within the viewport and provide a means of navigation to that content. The scrollbars make it clear that the rendered content is not fully visible.

**Examples for Success Criterion 1.8.3:**

* Tanya has low vision. She is reading a recipe on a website with a fixed size popup window. Tanya has set a preference in her browser to override the author setting and always display scrollbars on content overflow. She increases the font size causing the recipe to exceed the vertical and horizontal dimension of the viewport. The presence of the scrollbar shows her that additional ingredients may be present. She uses the scrollbar to move them into view.
* Terry has memory issues. She configures her mobile computer so that scrollbars are always on so she can instantly see where she is in a document. 记忆力有问题的用户，可以配置视窗上总是有滚动条，这样他就可以快速的知道他在什么位置

**Related Resources for Success Criterion 1.8.3:**

* None

**1.8.4 Indicate Viewport Position: 表明视窗位置**

The user can determine the viewport's position relative to the full extent of the [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content). 用户可以决定，视窗相对于全面呈现内容的位置。(Level A)

[Return to 1.8.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_184)

**Intent of Success Criterion 1.8.4:**

Users who have fine-motor problems that make it difficult to scroll, users who have cognitive issues that make it difficult to orient on the page, and screen reader users, who rely on audio to scan the page, all need to quickly assess the amount of content on the page and where they are located within the content.

**Examples for Success Criterion 1.8.4:**

* Ally has cognitive issues that make it difficult to orient. She navigates to a lengthy web page and begins paging through the content. A scroll bar indicates her position within the content as she pages and shows that with each paging action only a small portion of the content is rendered.
* Ally has cognitive issues that make it difficult to orient. When looking at a map on her mobile device, she must frequently zoom in to view her current location or destination and zoom out to put the location into the context of the large map.
* George uses a screenreader to access a lengthy web page. His screen reader speaks the percentage that the page is scrolled, allowing George to know where the cursor is relative to the entire page.

**Related Resources for Success Criterion 1.8.4:**

* None

**1.8.5 Allow Zoom: 允许缩放**

The user can rescale content within top-level graphical viewports as follows: 用户能够以如下方式在顶层的图形视窗里重新调节内容 (Level A)

* **Zoom in:** to 500% or more of the default size 放大：放大至默认尺寸的500%或更高
* **Zoom out:** to 10% or less of the default size, so the content fits within the height or width of the viewport 缩小：至默认尺寸的10%或者更小，内容可以适应视察的高度或宽度。

[Return to 1.8.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_185)

**Intent of Success Criterion 1.8.5:**

Some users want to be able to magnify content to so it is more legible. Some users  want to be able to shrink content so that more of it is visible onscreen. This can help them understand the structure of the content and their position in  the content, even if text has become too small to read.

**Examples for Success Criterion 1.8.5:**

* Tanya has low vision and needs large text. On web pages where graphics are strictly decorative, she increases the text size and leaves the graphics small. If she is using a web application with important graphics, however, both the graphics and the text need to be bigger, so she uses the zoom page feature.
* When Tanya views a web site on her mobile phone, she first scans the website at a very small size to guess where she wants to zoom in first. The zoom feature increases the size of both text and images.
* Perttu has a learning disability. He is studying an organization chart but has difficulty maintaining a mental representation of the organizational linkages for items out of view. In order to facilitate his understanding of the organization, Perttu zooms out to allow the entire chart to be displayed.

**Related Resources for Success Criterion 1.8.5:**

* [1.8.9 Allow Viewport Resize](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_189)
* [1.8.6 Maintain point of regard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_186)
* [1.8.7: Reduce Horizontal Scrolling](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_187)

**1.8.6 Maintain Point of Regard: 保持关注点**

The [point of regard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-point-of-regard) remains visible and at the same location within the viewport when the viewport is resized, when content is zoomed or scaled, or when content formatting is changed. (Level A)

[Return to 1.8.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_186)

**Intent of Success Criterion 1.8.6:**

Users can be confused and disoriented when the area where they are working suddenly shifts outside the visible region of the viewport. When this happens, users may have to expend considerable time and effort to re-navigate back to their previous point of regard. Just as the location in audio does not change when the user increases the volume, the point of regard should not change when the user changes the size of the window or zooms the content.

The point of regard is the information within the viewport that is visible to the user. When there is focused or selected content inside a viewport, and the viewport is resized, or content is zoomed, scaled, or formatted differently, that content will remain visible in the viewport. Otherwise, the user agent should maintain the same top-left (top-right for text read right-to-left) corner as the initial viewport.

*Note:* User agents are encouraged to allow user to override author instructions not to wrap content (e.g. nowrap).

**Examples for Success Criterion 1.8.6:**

* Jorge has low vision. While viewing a webpage he sees a picture with a caption that is too small to read. He highlights the caption, then uses the browser's zoom feature to increase the size of the content so he can read the caption. Throughout the zooming process the highlighted caption remains in the viewport, allowing Jorge to keep oriented on the caption and begin reading it when the appropriate content size is reached. Later, while reading on a page, Jorge finds some text that's too large to read. The beginning of the large text is at the top of the browser content area. Jorge uses the zoom feature to make the content smaller. The text is reduced to a comfortable reading size and the beginning of the text remains at the top of the browser window.
* Melissa has a distraction disorder. She is doing research for school. She scrolls the content so the relevant section heading is the top line of the browser. She resizes the browser so she can see her notes and the browser at the same time. After resizing the browser the heading is still the top line.
* Xu has a reading disability. He is reading a page with footnotes that are too small to read on his mobile device. Xu places the footnote at the top of the browser, and using the increase font-size feature, he increases the font-size of the text on the page. The footnotes stay on the top of the viewport.

**Related Resources for Success Criterion 1.8.6:**

* [1.8.2 Move Viewport to Selection and Focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_182)

**1.8.7 Reflow Text: 重构文本**

The user can specify that text content in a graphical top-level viewport reflows so the text forms a single column that fits within the width of the viewport. (Level A)

* *Note 1:* Reflow applies even when text is rescaled or zoomed.
* *Note 2:*For vertical layout languages (e.g. Mongolian, Han), text should fit within the height of the viewport, rather than its width, to reduce vertical scrolling.

[Return to 1.8.7 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_187)

**Intent of Success Criterion 1.8.7:**

It's important that reflowed content remains easily usable. Content is not easily usable if the user has to scroll back and forth horizontally to see a single line of text or has to scroll to get from the bottom of a column to the top of the next column to read a block of text. It is especially important that users can blocks of text reflow when zoomed.This is an especially acute issue for users who find it difficult or impossible to use the mouse to scroll and for users who find it difficult to reorient when the content changes. This does not require or prohibit the user agent from providing an option to turn off reflow.

**Examples for Success Criterion 1.8.7:**

* Frank has repetitive strain injuries and uses speech input. As he's gotten older he sometimes needs to increase the font size to comfortably read text on his computer screen. If the text does not flow properly, he overuses his voice to carry out multiple speech commands, or uses the mouse, which exacerbates his injuries, or simply gives up.
* Frank has repetitive strain injuries and uses speech input. When Frank is using his mobile phone to read a web page, he will zoom in to read a article on a web site. He configures his mobile phone so that the text reflows to always display zoomed content to fit in one column.
* Maggie has cognitive issues that make it difficult for her to reorient when the computer screen changes. Finding and activating scrollbars, then finding the next words in the sentence requires extra effort and causes Maggie to lose her orientation, which degrades her reading flow and comprehension. When text is reflowed so it is in one column that doesn't require horizontal scrolling or vertical scrolling to get to another column, she can read and understand it.
* Ansgard has low vision and physical disabilities that makes it difficult to use the mouse. He zooms text to read it. When reading PDF documents, he uses a special viewer that lets him zoom and reflow the content into a single column that fits the window. This is especially useful for scientific papers and newsletters that are in that are multiple columns.

**Related Resources for Success Criterion 1.8.7:**

* [1.8.5 Allow Zoom](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_185)
* [1.4.4 Configured and Reflowed Text Printing](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_144)

**1.8.8 Customize Viewport Highlighting:  自定义视窗高亮显示**

When highlighting viewports as specified by [1.8.1 Highlight Viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_181), the user can customize attributes of the viewport highlighting mechanism (e.g. blink rate for blinking, color and width of borders). (Level AA)

[Return to 1.8.8 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_188)

**Intent of Success Criterion 1.8.8:**

When a user agent presents content in multiple viewports, users benefit from a clear indication of which viewport has focus. Text foreground and background colors may not provide enough indication of viewport focus for users with low vision. These users need to customize viewport frames using color, contrast, and border thickness to provide multiple visual highlighting cues.

**Examples for Success Criterion 1.8.8:**

* Tanya has low vision. She tabs through the open windows on her new browser, but can't determine which viewport is the active viewport. She opens the browser settings and customizes the viewport frame in bright yellow with a thick black border so it's easier to see.

**Related Resources for Success Criterion 1.8.8:**

* [1.8.1 Highlight Viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_181)

**1.8.9 Allow Viewport Resize: 允许视窗重新定义尺寸**

The user can resize [viewports](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport) within restrictions imposed by the platform, [overriding](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-override) any values specified by the [author](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author). (Level AA)

[Return to 1.8.9 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_189)

**Intent of Success Criterion 1.8.9:**

If a viewport contains content that exceeds the dimensions of the viewport, users can increase the size of the viewport – up to the limits of the physical display screen – to allow the content to be displayed without horizontal scrolling. This benefits keyboard users who may find it difficult to scroll content. Other users with cognitive or learning disabilities may improve their ability to read the text by making the viewport narrower, and thus shortening the line lengths.

**Examples for Success Criterion 1.8.9:**

* Perttu has a learning disability. He is studying an organization chart and has difficulty maintaining a mental representation of the organizational linkages for items out of view. In order to facilitate his understanding of the organization, he sizes the viewport to allow the entire chart to be displayed.
* Vaalu has tunnel vision and can only see a few words at a time. If blocks of text are in long lines, it is difficult for him to get from the end of one line to the beginning of the next line. He often makes the viewport narrower and rewraps the text so the text is in shorter lines.

**Related Resources for Success Criterion 1.8.9:**

* [W3C: Fullscreen](http://www.w3.org/TR/fullscreen/) (not complete at this writing)

**1.8.10 Provide Viewport History:提供视窗历史**

For [user](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent)that implement a history mechanism for [top-level viewports](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport-toplevel) (e.g. "back" button), the user can return to any state in the viewport history that is allowed by the content, including a restored [point of regard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-point-of-regard), input focus and selection. (Level AA)

[Return to 1.8.10 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1810)

**1.8.11 Allow Top-Level Viewport Open on Request: 允许最上层的视窗根据需求打开**

The user can specify whether author content can open new [top-level viewports](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport-toplevel) (e.g. windows or tabs). 用户可以指明是否作者内容可以打开新最上层视窗(Level AA)

[Return to 1.8.11 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1811)

**1.8.12 Allow Top-Level Viewport Focus Control:允许最上层的视窗聚焦控制**

If new [top-level viewports](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport-toplevel) (e.g. windows or tabs) are configured to open without explicit user request, the user can specify whether or not top-level viewports take the active [keyboard focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-focus) when they open. (Level AA)

[Return to 1.8.12 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1812)

**Intent of Success Criteria 1.8.10, 1.8.11 & 1.8.12:**

Unexpected focus and viewport changes can be disorienting for all users, requiring time and effort for the user to orient to the change. These success criteria are intended to allow the user to be in control of when viewport changes happen so the user can orient to the changes in a predictable fashion.

**Examples for Success Criteria 1.8.10, 1.8.11 & 1.8.12:**

* Justin has an attention deficit disorder. He is studying an online college course and finds a word that he doesn't know. He follows a link to the definition of the word. After reading the definition, he uses the back button (viewport history) to return. The word remains selected so he doesn't have to reorient or find it on the page.
* George uses a screen reader. He visits a web page that ordinarily opens a popup page. George receives an audio alert that additional content is available. He can choose to have this pop-up content open on request.
* John has low vision and uses a screen magnifier to access his browser. He configures his browser so popup windows do not take focus because the popups can open outside the view of his magnifier.
* Frank has repetitive strain injuries and uses speech input. He configures his browser so that pop-ups always take focus, so he doesn't have to take multiple steps to locate the window and take the desired action.
* Ray is blind. His mobile device automatically opens location links and calendar dates found on web pages in native apps available on the device. When he returns to the browser, focus on the original link is maintained.

**Related Resources for Success Criteria 1.8.10, 1.8.11 & 1.8.12:**

* [3.4.1 Avoid unpredictable focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_341)
* [3.2.2 Back button](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_322)

**1.8.13 Allow Same User Interface:允许同一用户界面**

The user can specify that all [top-level viewports](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport-toplevel) (e.g. windows or tabs) follow the defined user interface configuration. 用户能够指明所有的上层视窗（例如，窗口或标签）遵循已明确的用户界面配置(Level AA)

[Return to 1.8.13 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1813)

**Intent of Success Criterion 1.8.13:**

Users orient themselves to a browsing environment with a variety of techniques. This success criteria is designed to ensure that the user does not have to learn multiple strategies to use the browsing viewport.

**Examples for Success Criterion 1.8.13:**

* Robert uses magnification software. After setting up his user agent, he knows that web content begins predictably one inch from the top of the window, so he can configure his magnification software to present content starting at that location.
* Courtney has difficulty understanding and using complex user interfaces. She has worked with her sister to set up her browser to have only a few of the most common browser controls displayed and she knows to expect them on every browser window.

**Related Resources for Success Criterion 1.8.13:**

* [2.7.3 Multiple Sets of Preference Settings](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_274)

**1.8.14 Provide Webpage Bookmarks:提供网页书签**

The user can mark items in a webpage, then use shortcuts to navigate back to marked items. The user can specify whether a navigation mark disappears after a session, or is persistent across sessions. (Level AAA)

[Return to 1.8.14 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1814)

**Intent of Success Criterion 1.8.14:**

This success criterion is crucial for users who have trouble navigating a webpage. People who use speech input, have memory problems, or use small screens may be able to go from one area of a webpage to another area once or twice, but may have trouble frequently repeating the action. The ability to mark areas of the page allows these types of users to navigate more quickly with less fatigue.

**Examples for Success Criterion 1.8.14:**

* Jamie is a quadriplegic who uses speech input. She is a professor who reads long documents online and often finds herself comparing different portions of the same document. It is tedious carrying out multiple scrolling commands by speech every time she needs to change to another portion of the document. She sets several bookmarks instead. This allows her to instantly jump among sections, eliminating the time and effort penalties she usually has to pay for slow scrolling. Jamie also uses bookmarks on her mobile phone to cut down on scrolling.
* Julie has memory problems. She finds it difficult to remember key points from a document she has just read. She uses bookmarks to mark important points she needs to read again. Without the ability to bookmark, she won't remember what she needs to read multiple times in order to do so.
* George is blind and occasionally travels to unfamiliar places for work. He sometimes uses an iPhone to orient himself within a map of the building he's in. Once he's found a key place on the map – a room where a conference session is held, or the bathroom – he bookmarks it to build a map of useful places. This makes navigation easier the second time around. He sets the marks to be persistent across sessions so next time he visits he won't have to repeat his work.
* Mary has repetitive strain injuries that make it painful to use a mouse. She is a college professor who uses an elaborate web application to correct papers. After putting a comment in a comment field she has to scroll all the way to the bottom of the document to enter the comment. The bookmarks make even this badly designed application something she can use successfully without hurting herself.

**Related Resources for Success Criterion 1.8.14:**

* None

Implementing Guideline 1.9 - Provide alternative views [[Guideline 1.9](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-alternative-views)] 提供替换的视图方式

**Summary**: The user can view the source of content (1.9.2, Level AAA), and an outline view of important elements. (1.9.1, Level AA).

**1.9.1 Outline View: 大纲视图**

Users can view a navigable outline of [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content) composed of labels for [important elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-important-elements), and can move focus efficiently to these elements in the main viewport. (Level AA)

* *Note:* The important elements depend on the [web content technology](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-Web-Content-Technology), but may include headings, table captions, and content sections.

[Return to 1.9.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_191)

**Intent of Success Criterion 1.9.1:**

Outline views allow users to get a simplified view or overview of a document. They are particularly useful for users with memory or cognitive disabilities, blind users, and users who find it difficult or impossible to use a mouse. A navigable outline views reduce orientation and navigation time and fatigue.

**Examples for Success Criterion 1.9.1:**

* Frank has repetitive strain injuries and uses speech input. He uses a web browser to read financial information. His browser provides an optional panel displaying a hierarchical list of the headers and tables in the current document. Frank is able to expand or shrink portions of the outline view for faster access to the information he wants.
* George uses a screen reader. He reads long standards documents and uses the headings to navigate quickly so he can compare sections of the standards. George also finds the outline view useful when he is quickly checking a reference on his mobile phone.

**Related Resources for Success Criterion 1.9.1:**

* [Guideline 2.5 Provide structural navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-nav-structure)
* [Success Criterion 2.3.1 Direct Navigation to Important Elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_231)
* [Success Criterion 2.5.3 Configure Elements for Structural Navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_253)

**1.9.2 Source View: 源视图**

The user can view all [source text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text-source) that is available to the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). (Level AAA)

[Return to 1.9.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_192)

**Intent of Success Criterion 1.9.2:**

The source view is the ultimate fallback for a person with disabilities when the browser cannot properly render some content, or when the user cannot take advantage of the content as rendered or using the mechanisms provided.

**Examples for Success Criterion 1.9.2:**

* George is a web developer who uses a screenreader. He visits a web page where the content author failed to provide alt text or a long description for an image he wants to access. As a last resort, George examines the source to see the image's URI, class, and similar attributes. He sees that part of the URI for the image is "home.jpg" and concludes that he can click on that image to return to the home page of the site. George also uses the source view feature on his mobile device when he needs to identify an image.
* Mikki has low vision. She wants to create a user stylesheet for her web-based email program. She examines the source code to identify the CSS class of the email heading. She then creates a user stylesheet that increases the size and contrast of the email headings.

**Related Resources for Success Criterion 1.9.2:**

* None

Implementing Guideline 1.10 - Provide element information [[Guideline 1.10](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-info-link)] 提供元素信息

**Summary**: The user can access information about relationships between elements (e.g. form labels, table headers) (1.10.1, Level AA), and extended link information (e.g. title, internal vs. external) (1.10.2, Level AAA)

**1.10.1 Show Related Elements: 显示相关的元素**

The user can access related elements based on the user's position in content (e.g. show the label of a form control, show the headers of a table cell). (Level AA)

[Return to 1.10.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1101)

**Intent of Success Criterion 1.10.1:**

Some users have difficulty perceiving, remembering, or understanding the relationships between elements and their descriptions. Certain elements relate to others in a recognizable manner, such as relationships with 'id' attributes and child elements (e.g. Ajax widgets and with form elements). This allows users to better understand these relationships even if the elements are not adjacent on the screen or the DOM.

**Examples for Success Criterion 1.10.1:**

* John has low vision and uses a screen magnifier to access his browser. When interacting with tables and spreadsheets John has to move the viewport of the magnifier to understand the row and column titles of the cell with which he is interacting. This takes additional time and effort and is therefore frustrating. John switches to a browser that presents the row and column titles when he hovers over a cell - this makes him much more productive at his accounting job.
* Courtney has a cognitive disability that makes it difficult for her to comprehend complex user interfaces. She is completing an online job application. It is not clear where she should enter her phone number. She mouses over a blank box and sees a tooltip that says "home phone number". She is able to complete the form.

**Related Resources for Success Criterion 1.10.1:**

* [WAI-ARIA: 6.5.4 Relationship Attributes](http://www.w3.org/TR/wai-aria/states_and_properties#attrs_relationships)
* [WAI-ARIA Authoring Practices: 4. Relationships](http://www.w3.org/TR/wai-aria-practices/#relations)

**1.10.2 Show Element Hierarchy: 显示元素层级**

The user can determine the path of element nodes going from the root element of the element hierarchy to the currently focused or selected element. (Level AAA)

[Return to 1.10.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_1102)

**Intent of Success Criterion 1.10.2:**

Users who have difficulty working with a web page or document can use user stylesheets or scripts to modify page presentation or interaction so they can gain information or accomplish a task. Stylesheets and scripts may require the user to identify specific elements, element attributes, and element position in the hierarchy. The user agent can facilitate this process by allowing the user to navigate to an element, select it if it's not navigable, and query for element information. If this feature is not provided, the user may be forced to try to find the corresponding element in the source view or entire document tree.

**Examples for Success Criterion 1.10.2:**

* Jack has low vision and uses user stylesheets frequently. He wants certain content on a web page to be displayed in a larger font, and wants to create a user style sheet that would modify its appearance. He needs to identify the class or ID of the particular element, so he selects the text he's interested in, opens the browser's debug window, which shows him that the selected text is an element with class "story" inside a paragraph inside a DIV with class "Premiere". He then knows the combination of classes and element types to specify in the user style sheet.

**Related Resources for Success Criterion 1.10.2:**

* Guideline [1.7 Enable Configuration of User Stylesheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-style-sheets-config)

**PRINCIPLE 2. Ensure that the user interface is operable 确认用户界面是可操作的**

* *Note*: Users interacting with a web browser may do so using one or more input methods including keyboard, mouse, speech, touch, and gesture. It's critical that each user be free to use whatever input method or combination of methods works best for a given situation. If every potential user task is made accessible via modality independent controls that any input technology can access, a user can use what works best. For instance, if a user can't use or doesn't have access to a mouse, but can use and access a keyboard, the keyboard can call a modality independent control to activate an OnMouseOver event. See [Independent User Interface: Events](http://www.w3.org/TR/indie-ui-events/) for additional information on APIs and techniques for modality independent controls.

Implementing Guideline 2.1 - Ensure full keyboard access 确保全键盘访问[[Guideline 2.1](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-keyboard-access)]

**Summary**: Every viewport has a keyboard focus (2.1.2, Level A). Users can operate all functions using just the keyboard (2.1.1, Level A), activate important or common features with shortcut keys, (2.1.6, Level A), escape keyboard traps (2.1.3, Level A), specify that selecting an item in a dropdown list or menu not activate that item (2.1.4, Level A) and use standard keys for its platform (2.1.5, Level A).

**2.1.1 Provide Full Keyboard Functionality: 提供全键盘功能**

All functionality can be operated via the [keyboard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard) using sequential or direct keyboard commands that do not require specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints (e.g. free hand drawing). This does not forbid and should not discourage providing other input methods in addition to keyboard operation including mouse, touch, gesture and speech. (Level A)

[Return to 2.1.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_211)

**Intent of Success Criterion 2.1.1:**

A user has many ways to input information into a computer or device, including mouse, keyboard, gesture, and speech. The keyboard paradigm is the most universal interface for text input – even devices that do not have a keyboard (like mobile phones) support a software interface for them. A user should be able to navigate, read and use all of the web page or application without needing to use a mouse. Some users do not use a mouse. Others can only use a pointing device that uses the keyboard API. It's important that these users be able to interact with enabled components, select content, navigate viewports, configure the user agent, access documentation, install the user agent, and operate user interface controls, all entirely through keyboard input. User agents generally support at least three types of keyboard operation:

1. Direct (e.g. keyboard shortcuts such a "F1" to open the help menu; see checkpoint 11.4 for single-key access requirements)
2. Sequential (e.g. navigation through cascading menus)
3. Spatial (e.g. when the keyboard is used to move the pointing device in two-dimensional visual space to manipulate a bitmap image)

User agents should support direct or sequential keyboard operation for all functionalities. The user agent should offer a combination of keyboard-operable user interface controls (e.g. keyboard operable print menus and settings) and direct keyboard shortcuts (e.g. to print the current page).

**Examples for Success Criterion 2.1.1:**

* Amal has a medical condition that prevents him from using the mouse. Jeremy finds it difficult or impossible to use a mouse or keyboard and therefore uses speech to control the keyboard. Tanya is blind and uses a screenreader and must be able to accomplish all tasks with the keyboard. Jamie uses a switch device to control the computer. Each of these users must be able to do the following through keyboard alone, speech input alone, or switch device alone:
  + Select content and operate on it. For example, if it is possible to select rendered text with the mouse and make it the content of a new link by pushing a button, these users also need to be able to do so through the keyboard and other supported devices. Other examples include cut, copy, and paste.
  + Set the focus on viewports and on enabled elements.
  + Install, configure, uninstall, and update the user agent software.
  + Amal, Jeremy and Jamie use the graphical user interface menus even though they cannot use the mouse.
  + Fill out forms.
  + Access documentation.
* Amal is reading a web page where the author used the CSS overflow property to constrain the size of a block of content. Amal's browser provides scroll bars to display text that overflows the viewport. He uses the keyboard to enter the element and operate the scrollbars to read all the content, he then uses the keyboard to return focus to the main web page. (see [2.1.3 No Keyboard Trap](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_213))
* Tanya needs to use a volume control widget to change the volume on a video she is watching. She use the keyboard to navigate to the widget, uses the arrow keys to increase the volume, then uses the keyboard to navigate to the comments below the video player.
* Cade has low vision. He can't identify an icon and so he needs to read its alternative text. Mouse users could hover the mouse over the icon and have its alternative text displayed as a pop-up or tooltip, but Cade cannot use a mouse. He uses the keyboard to move focus to the icon. The browser automatically displays the "hover" tooltip, or allows him to press a key to have it displayed.
* Jeremy is a speech-input user who cannot use his hands to control his computer. It is much easier for him to speak keyboard shortcuts then click on an icon in a new program. He needs to be able to see tooltips to discover keyboard shortcuts without having to use the mouse.
* Karen has muscular dystrophy and cannot easily use the onscreen keyboard to navigate web pages on her mobile phone. Instead, she uses simple gestures to move between elements on the page. As focus moves from one element to another, there is a visble focus indicator.

**Related Resources for Success Criterion 2.1.1:**

* [Modality Independent Controls](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#intro-modality-independence)
* [2.1.3 No Keyboard Trap](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_213)

**2.1.2 Show Keyboard Focus: 显示键盘焦点**

Every [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport) has an active or inactive [keyboard focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-focus) at all times. (Level A)

[Return to 2.1.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_212)

**Intent of Success Criterion 2.1.2:**

Both the user and some types of assistive technology need to know what will be affected by any keyboard input, so it's important that they be able to tell which window, viewport, and controls have the keyboard focus at any time. This applies whether window and viewport are active (active keyboard focus) or inactive (inactive keyboard focus). Even when a window is inactive, it can be affected by simulated keyboard input sent by assistive technology tools. Active keyboard focus is indicated to the user by focus cursors and text cursors, as required by Guidelines 1.3, and made available to assistive technology, as required by [Success Criterion 4.1.6](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_416).

**Examples for Success Criterion 2.1.2:**

* Amal has a medical condition that prevents him from using the mouse. He launches a web browser and navigates to a web page. Initially the keyboard focus is on the entire document, which is exposed to assistive technology, but there is no visible cursor. When Amal presses the tab key, the focus moves to the first link on the web page, and a cursor in the form of a dotted rectangle appears around that link.
* Amal launches a web browser and navigates to a web page that has an enabled edit field. The browser places the keyboard focus on the edit field so Amal can immediately start entering text, and its location is shown using a text cursor (usually a vertical line or i-beam). As he types, the text cursor moves to show where the next character will appear. If Amal activates another window, the browser may hide the cursor in the now inactive window, but its location is still available to assistive technology.
* Raymond has low vision. As the keyboard focus moves from one control to another, or one window to another, his screen enlarger utility detects the focus change and pans its viewport to keep the focus location visible.
* Jeremy is a speech-input user who cannot use his hands to control his tablet. He opens a webpage using a speech command. The webpage has a search field, and normally comes up with the keyboard focus in the search field. Jeremy sees the indicator in the search field and knows he does not have to navigate to the search field before saying a search term.
* Erin has dyslexia which often causes her to confuse directions. She uses gestures to navigate her mobile phone. As focus moves from one element to another, there is a visble focus indicator, which allows her to find the focus easily.

**Related Resources for Success Criterion 2.1.2:**

* [Guideline 1.3 Provide highlighting for selection, keyboard focus, enabled elements, visited links](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-interaction-highlight)
* [4.1.2 Expose Basic Properties](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_412) ensures that the focus location is available to assistive technology.

**2.1.3 Avoid Keyboard Traps: 避免键盘陷井**

If keyboard focus can be moved to a component using a keyboard interface (including nested user agents), then focus can be moved away from that component using only a keyboard interface. If this requires more than unmodified arrow or Tab keys (or standard exit methods like Escape), users are advised of the method for moving focus away. (Level A)

[Return to 2.1.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_213)

**Intent of Success Criterion 2.1.3:**

If users can put focus on an element, they can remove focus and move on to the next element. This is often not possible with embedded objects. The user agent needs to provide a way to always return to the previous or next element in the content, or a known location such as the address bar. The user agent also needs to take control back from the embedded object, no matter what it is.

**Examples for Success Criterion 2.1.3:**

* Ari has repetitive strain injuries that are exacerbated when he uses the mouse. He is using a video hosting site where each page hosts a nested media player. He presses Tab until the focus in on the media player, then presses Enter to activate and put the keyboard focus on it. When he's finished watching the video, he presses Tab to navigate to the comments below the video, but cannot get the focus to leave the video player. He presses Alt+Left to return to the previous page, but that also fails because the video player is consuming those keystrokes. Luckily, Ari knows that Shift+Esc will return focus from a nested user agent, with or without its cooperation. Thus, even a badly behaved nested user agent cannot prevent Ari from getting on with his work.
* Ari moves the focus to a toolbar extension that does not relinquish control back to the user agent. He presses Alt-D to move the focus to the address bar.
* Mary is a blind user who does not use the mouse. She moves the focus to an embedded scripted application that was poorly programmed. She presses a documented key combination –  Alt+N – to override the scripting and move the focus to the next element in the content.
* Katan cannot use the mouse and has trouble with short-term memory. He is using a virtual machine. The escape sequence is Shift+Ctrl+Escape. Every time Katan opens the program, it briefly shows the escape sequence near the top right corner of the screen.  The virtual machine also has an Exit option in its menu system. The Escape keystrokes are also indicated on the menu.

**Related Resources for Success Criterion 2.1.3:**

* None

**2.1.4 Separate Selection from Activation: 从激活的内容中分离选择**

The user can specify that focus and selection can be moved without causing further changes in focus, selection, or the state of controls, by either the user agent or author-supplied content. (Level A)

[Return to 2.1.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_214)

**Intent of Success Criterion 2.1.4:**

People do not expect side effects when moving the keyboard focus regardless of whether the side effect is caused by the user agent or author content. If users fail to notice side effects, they could end up doing something disastrous. This is especially likely for users of assistive technology who cannot see changes happening elsewhere on the screen. Users may also find it confusing or disorienting if the effect causes unexpected focus movement or changes in context. If the user agent does implement side effects to keyboard navigation, it is recommended that it provide a user preference setting to disable them. However, in some cases it may be more appropriate to provide a separate navigation mechanism that avoids side effects, such as allowing the user to hold down the Ctrl key while navigating to avoid changing selection or choice. Note: It may not be possible for the user agent to detect or prevent side effects implemented by scripts in the content, but the user agent is required to prevent side effects that are under its control.

**Examples for Success Criterion 2.1.4:**

* Murray uses a screen magnifier that allows him to see the element with the focus and a small area around it. He explores a dialog box by repeatedly pressing the Tab key to move to, and read each control in succession. He uses the arrow keys to navigate through a dropdown menu. When he moves the focus to the first option he goes to that page. He wants to navigate to the third selection, but can't get there. Fortunately, the platform also has a convention that holding down the Ctrl key while navigating that moves the focus without changing selection or option choice. Murray uses this while exploring. His web browser implements its own form controls and navigation mechanisms rather than using the platform's infrastructure, but also implements this Ctrl-key mechanism for users like Murray.
* Ari uses the screen reader on his smartphone to navigate a webpage. He selects an item and is able to activate the element using gestures. This requires sufficient screen real estate to perform gestures without changing focus.

**Related Resources for Success Criterion 2.1.4:**

* [WCAG 2.0: 3.2.2 - On Input](http://www.w3.org/TR/2008/REC-WCAG20-20081211/#consistent-behavior-unpredictable-change)
* [WCAG 2.0: 3.2.5 Change on Request](http://www.w3.org/TR/2008/REC-WCAG20-20081211/#consistent-behavior-no-extreme-changes-context)
* [WCAG Techniques: SCR19](http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/SCR19): Using an onchange event on a select element without causing a change of context
* [WAI-ARIA: aria-selected (state)](http://www.w3.org/TR/wai-aria/states_and_properties#aria-selected)
* [1.8.12 Allow Top-Level Viewport Focus Control](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_1812)

**2.1.5 Follow Text Keyboard Conventions: 遵循文本键盘规律**

The user agent follows keyboard conventions for the [operating environment](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-environment). 用户代理遵循操作环境的键盘规则。 (Level A)

[Return to 2.1.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_215)

**Intent of Success Criterion 2.1.5:**

Keyboard users rely on the user agent to provide keyboard support that is full-featured and consistent among applications. Following platform conventions for keyboard access helps ensure that the functions that people rely on are not accidentally omitted. In addition, making these inputs consistent within and across programs greatly reduces learning curve, cognitive load, and errors. User agents are encouraged to add keyboard commands when the commands provide additional features or benefit for users. User agents should avoid omitting the standard commands, or assigning them to different keys.

**Examples for Success Criterion 2.1.5:**

* Jack is blind and uses a screenreader. He edits blog posts in his browser's text area control, and can use the same keys for navigation, editing, and formatting that he's used to using in other applications on the platform (e.g. arrow keys, commands to move to the beginning and end of the line, cut, copy, paste, ctrl-b for bolding).
* Jack's user agent uses a custom tree control showing a hierarchical outline view of the document headings. He can navigate within that control using the same keys as work in the native tree controls used in other applications (e.g. up, down, left and right arrow keys, and navigating directly to entries by typing the beginning of their text).
* Jack puts his browser into a caret browsing mode so he can move the text cursor through the text on the page. The browser supports the same keys for navigation and editing as are used in other applications.

**Related Resources for Success Criterion 2.1.5:**

* None

**2.1.6 Make Keyboard Access Efficient: 确保键盘访问有效：**

The [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) user interface includes mechanisms to make keyboard access more efficient than sequential keyboard access. 用户代理的用户界面包括某种机制使键盘访问比顺序键盘访问更有效率。(Level A)

[Return to 2.1.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_216)

**Intent of Success Criterion 2.1.6:**

Efficient keyboard navigation is especially important for people who cannot easily use a mouse, are quickly fatigued, or find it difficult to memorize the menu structure for sequential navigation. This is important in all types of user agent environments.

* **In a browser:** A browser provides keyboard shortcuts for its menu functions as well as access keys in the design of its menus and dialog boxes. The choice of shortcut keys follows platform conventions where applicable (e.g. for open document, save document, cut, copy, paste).
* **In a mobile environment:** A social networking application on a mobile device has only a very few keyboard shortcuts available on its targeted devices. These few keyboard shortcuts are used for the most commonly accessed functions of the application (e.g. home, list of friends).
* **In a media player:** An embedded media player provides shortcut keys for commonly used functions (e.g. pause and play).

**Examples for Success Criterion 2.1.6:**

* Jack is blind and uses a screenreader. He opens his browser to write his email. He uses a shortcut to start a new message. He uses the same Ctrl-N command he uses in his desktop email application.
* Jean has a medical condition that makes it difficult to visualize complex visual structures like dropdown menus. She is writing a new blog entry and discovers that her blogging application has updated and all the menus have changed. She cannot find the menu option to format the text, but is pleased to find that Ctrl-B still works to bold the text she wants.
* George is blind and uses the gestures on his mobile device to move focus to the top of the page, return to the previous web page and activate links.

**Related Resources for Success Criterion 2.1.6:**

* [5.1.3 Implement Accessibility Features of platform](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_513)

Implementing Guideline 2.2 - Provide sequential navigation [[Guideline 2.2](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-sequential-navigation)] 提供顺序导航

**Summary**:Users can use the keyboard to navigate sequentially to all the operable elements in the viewport (2.2.1, Level A) as well as between viewports (2.2.2, Level A), and the default navigation order is document order (2.2.3, Level A). Users can optionally disable wrapping or request a signal when wrapping occurs (2.2.4, Level AA).

**2.2.1 Sequential Navigation Between Elements: 在元素间的顺序导航**

The user can move the keyboard focus backwards and forwards through all [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) enabled elements in the current viewport. 用户可以通过当前视窗内所有可识别可用的元素，向前向后移动键盘聚集(Level A)

[Return to 2.2.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_221)

**Intent of Success Criterion 2.2.1:**

Sequential keyboard navigation is a fundamental, universal method of keyboard access. While it can be slower and require more input than other methods (such as direct, structural, or search-based navigation) it is a simpler mechanism that requires very little cognitive load or memorization, and is consistent across contexts. Users need keyboard access to all viewports and all enabled elements so that they can manipulate them, view them with screen magnifiers, or have them described by screen readers. The ability to move both forward and backward through the navigation order greatly reduces the number of keystrokes and allows the user to more easily recover from mistakes in overshooting a destination.

**Examples for Success Criterion 2.2.1:**

* Sooj cannot use a pointing device, so she moves the keyboard focus to the next enabled element by pressing the Tab key, and to the previous enabled element by pressing Shift+Tab. Within list boxes and radio button groups she uses the up and down arrow keys to move to the next and previous items.
* George is blind and uses a screenreader on his computer and the speech output and gesture features of his mobile phone. When completing a web form on his phone, he uses the swipe gesture to advance through the form. If George goes past the next form field, or wishes to return to a previous form field, he can use a gesture to go backward.

**Related Resources for Success Criterion 2.2.1:**

* [2.3.1 Direct Navigation to Important Elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_231)

**2.2.2 Sequential Navigation Between Viewports: 视窗间的顺序导航**

The user can move the keyboard focus backwards and forwards between viewports, without having to[sequentially navigate](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-sequential-navigation) all the elements in a viewport. (Level A)

[Return to 2.2.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_222)

**Intent of Success Criterion 2.2.2:**

Users need to be able to jump directly to next or previous viewports without having to visit every element in a viewport on the way to the next viewport. Not being able to jump directly can add an exorbitant number of navigation commands to operations that should be easy and efficient. Users need keyboard access to all viewports and enabled elements so that they can manipulate them, view them with screen magnifiers, or have them described by screen readers. The ability to move both forward and backward through the navigation order greatly reduces the number of keystrokes and allows the user to more easily recover from mistakes in overshooting a destination. This navigation can be among applications, windows, or viewports within an application. This includes the user agent's user interface, extensions to the user interface (e.g. add-ons), and content.

**Examples for Success Criterion 2.2.2:**

* Sooj has a repetitive stress injury and cannot use a mouse, so she moves the keyboard focus to the next pane by pressing F6 or to the previous pane by pressing Shift+F6. She moves between tabbed document views by pressing Ctrl+Tab and Shift+Ctrl+Tab.
* Sooj is working in her web browser, where one document window (viewport) is active (has the active keyboard focus). When she switches to her word processor, the web browser's window and its keyboard focus become inactive, and it hides its cursor. When Sooj switches back to the browser window, it reactivates that viewport, its keyboard focus becomes active again, and its cursor reappears in the same location as when she switched to a different application.
* A developer creates an extension to a user agent that allows the user to add notes about each web page being visited. Sooj presses a shortcut key to move focus to the user interface of this extension and interact with the functionality offered by the extension. Similarly, Sooj presses another key to move focus back to the main viewport for the user agent in the same location as when she moved to the plugin.

**Related Resources for Success Criterion 2.2.2:**

* [2.2.1 Sequential Navigation Between Elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_221) for background information on the importance of sequential navigation.

**2.2.3 Default Navigation Order: 默认的导航顺序**

If the author has not specified a navigation order, the default [sequential navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-sequential-navigation) order is the document order. (Level A)

[Return to 2.2.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_223)

**Intent of Success Criterion 2.2.3:**

When the content author doesn't explicitly define a consistent tab order, the browser will provide one. Users need to have a mental map of where the focus will land when they press the Tab key or use other sequential navigation commands. If the focus jumps in seemingly random fashion, skipping up or down, it becomes impossible to use this method efficiently because users must stop, find the focus, reorient, and determine what direction they should proceed every time they press navigation keys. This is a particular problem for users with some cognitive limitations or whose disability makes input difficult, tiring, or painful. Content authors are expected to define a logical navigation order in their documents, but if they have not, this success criterion ensures that the order will at least be consistent between user agents.

**Examples for Success Criterion 2.2.3:**

* Alec is filling out an HTML form. Because the form's author has not specified a navigation order using the tabindex attribute, when Alec presses the Tab key the focus moves to the next control in the order defined in the underlying HTML. This order is logical as long as the author is not using styles to change the visual order. Alec has the same experience completing this form on his mobile phone.

**Related Resources for Success Criterion 2.2.3:**

* [WCAG 2.0 success criterion 2.4.3 Focus Order](http://www.w3.org/TR/2008/REC-WCAG20-20081211/#navigation-mechanisms-focus-order)

**2.2.4 Options for Wrapping in Navigation:**

The user can request notification when [sequential navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-sequential-navigation) wraps at the beginning or end of a document, and can prevent such wrapping. (Level AA)

[Return to 2.2.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_224)

**Intent of Success Criterion 2.2.4:**

Users need a good mental map of the navigation sequence and behavior, and particularly need to know when they have started over again so they can maintain that mental map and not waste time and energy inadvertently revisiting information. This is a greater problem for users who have limited short-term memory, perceive a narrow field of vision, or use a screen magnifier, screen reader or small screen device. This also prevents people with mobility issues from having to use extra navigation commands.

**Examples for Success Criterion 2.2.4:**

* Betsy has low vision. She is using a screen magnifier that only shows her a single line of text. She's navigating through a long list of unsorted items in a list box, searching for an entry that she does not realize is not in the list. Each time she presses the down arrow she is presented with the next item in the list. The list box wraps, so after she reads the final entry and presses the down arrow, she is once again presented with the first entry in the list. Unfortunately, it takes her a long time to realize that she's scrolling through the same set of items again and again. To avoid this, she can turn on an option to prevent wrapping, or have the user agent play a sound or display a message to indicate that it is wrapping back to the first item. This behavior should be under the user's control, because keyboard users who can see the entire screen may not want to be interrupted by a pop-up dialog box.
* Jeff has a mobility impairment. He uses gestures to navigate the page. When he reaches the last active element on the page there is an indicator that the end of the page is reached before changing focus (e.g. wrapping to the top, switching pages).

**Related Resources for Success Criterion 2.2.4:**

* None

Implementing Guideline 2.3 - Provide direct navigation and activation [[Guideline 2.3](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-direct-navigation-and-activation)] 提供直接导航和激活

**Summary**: Users can navigate directly (e.g. using keyboard shortcuts) to important elements (2.3.1, Level AA) with the option of immediate activation of the operable elements (2.3.3, Level A). Display commands with the elements to make it easier for users to discover the commands (2.3.2 & 2.3.4, Level AA). The user can remap and save direct commands (2.3.5, Level AA).

**2.3.1 Allow Direct Navigation to Important Elements: 允许直接导航至重要元素**

The user can navigate directly to [important elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-important-elements) in rendered content. 用户能够直接地导航至呈现内容中的重要元素 (Level AA)

[Return to 2.3.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_231)

**Intent of Success Criterion 2.3.1:**

People who are blind or have mobility problems often find it difficult or impossible to use a mouse to move the viewport to, and focus on, important elements. Some other form of direct navigation – such as numbers or key combinations assigned to important elements – should be available. Direct navigation can be accessed via keyboard, which also supports other forms of input, such as gesture, speech and touch.

**Examples for Success Criterion 2.3.1:**

* Mary cannot use the mouse or keyboard due to a repetitive strain injury. She uses speech input with a mouseless browsing plug-in for her browser. She is able to use the same plug-in on her smartphone. The plug-in overlays each link with a number that can then be used to directly select it (e.g. by speaking the command "link 12"). This prevents Mary from having to say "tab" numerous times to select a link.

**Related Resources for Success Criterion 2.3.1:**

* [2.2.1 Sequential Navigation Between Elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_221)
* [2.5.3 Configure Elements for Structural Navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_253)
* [Mouseless Browsing Firefox Extension](https://addons.mozilla.org/en-us/firefox/addon/mouseless-browsing/)

**2.3.2 Present Direct Commands from Rendered Content: 在呈现的内容中展示直接命令**

The user can have any [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) [direct commands](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-direct-command) in rendered content (e.g. accesskey, landmark) be presented with their associated elements (e.g. Alt+R to reply to a web email). (Level AA)

[Return to 2.3.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_232)

**Intent of Success Criterion 2.3.2:**

For many users, including those who use the keyboard or an input method such as speech, the keyboard is often a primary method of user agent control. It is important that direct keyboard commands assigned to user agent functionality be discoverable, including in rendered content. If direct commands are not presented in content, many users will not discover them.

**Examples for Success Criterion 2.3.2:**

* Fiona uses an audio browser. When the system reads form controls in the rendered content, it reads the label of the form followed by the accesskey (e.g. "name alt plus n").
* Mary cannot use the mouse or keyboard. She uses speech input. She is composing an email and wants to attach a file. She sees Ctrl+Shift+A next to the link to attach a file. Mary says the command to add the attachment.
* When reading email on her tablet, Mary touches a control which opens a toolbar with a setting to display the accesskeys and other direct commands that the author created. She sees that a 3-finger swipe will delete the current email.

**Related Resources for Success Criterion 2.3.2:**

* [2.1.5 Follow Text Keyboard Conventions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_232)
* [Mouseless & More](http://www.mouseless.de/index.php?/content/view/17/30/) for perceivable navigation and activation keys
* [Accesskey shortcuts in Google Chrome and more](http://aloiroberto.wordpress.com/2010/06/12/how-to-display-accesskey-shortcuts-in-google-chrome-and-much-more/)
* [Firefox hack to display "accesskey" keyboard shortcuts](http://blog.andrewbeacock.com/2007/06/firefox-hack-to-display-accesskey.html)

**2.3.3 Allow Direct Activation of Enabled Elements: 允许可用元素的直接激活**

The user can move directly to and activate any enabled element in rendered content. 用户能够在呈现的内容中直接移动和激活任意可用元素(Level A)

[Return to 2.3.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_233)

**Intent of Success Criterion 2.3.3:**

People who are blind or have mobility problems often find it difficult or impossible to use a mouse to move the viewport to, focus on, and activate important elements. Some other form of direct navigation – such as numbers or key combinations assigned to important elements – should be available. Direct navigation can be accessed via keyboard, which also supports other forms of input, such as gesture, speech and touch.

**Examples for Success Criterion 2.3.3:**

* Mary cannot use the mouse or keyboard due to a repetitive strain injury, instead she uses voice control technology with uses a mouse-less browsing plug-in to her browser. The plug-in overlays each hyperlink with a number that can then be used to directly select it (e.g. by speaking the command "select link twelve"). This prevents Mary from having to say the word 'tab' numerous times to get to her desired hyperlink.
* Mary cannot use the mouse or keyboard due to a repetitive strain injury. On her mobile phone, Mary uses a single speech command to launch the app, rather than having to use multiple commands to page through screens to find the app icon and activate it.

**Related Resources for Success Criterion 2.3.3:**

* [Configure Elements for Structural Navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_253)

**2.3.4 Present Direct Commands in User Interface: 在用户界面中展示直接命令**

The user can have any direct commands in the user agent user interface (e.g. keyboard shortcuts) be presented with their associated user interface controls (e.g. "Ctrl+S" displayed on the "Save" menu item and toolbar button). (Level AA)

[Return to 2.3.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_234)

**Intent of Success Criterion 2.3.4:**

For many users, including those who use the keyboard or and input method such as speech, the keyboard is often a primary method of user agent control. It is important that direct keyboard commands assigned to user agent functionality be discoverable as the user is exploring the user agent. If direct commands are not presented in content, many users will not discover them.

**Examples for Success Criterion 2.3.4:**

* Vlad is a keyboard-only user who uses a browser on the Mac operating system. When he needs to perform a new operation with the browser user interface, he searches for it in the menus, and notices that the menu item has a "⌘" label (e.g. "Copy ⌘+C"). This indicates the direct activation command he can use in the future to avoid having to traverse the menus.
* Amir uses ability switches to control an onscreen keyboard for the Windows operating system. When he presses the "Alt" key the available browser user interface accesskeys are shown as overlays on the appropriate user interface controls (e.g. "File with 'F' in an overlay").
* Neta has a repetitive strain injury. She relies on gestures and shortcuts to complete tasks. Using a specialized command on her mobile device, she can pull up an overlay of arrows and text showing all the commands that can be completed in that context. This allows her to learn new programs as efficiently as possible, making it less likely she will overtax her hands.

**Related Resources for Success Criterion 2.3.4:**

* None

**2.3.5 Allow Customized Keyboard Commands: 允许自定键盘命令**

The user can remap any [keyboard shortcut](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-shortcut) including recognized author supplied shortcuts (e.g. accesskeys) and user agent user interface controls, except for conventional bindings for the [operating environment](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-environment) (e.g. arrow keys for navigating within menus). (Level AA)

[Return to 2.3.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_235)

**Intent of Success Criterion 2.3.5**

People using a keyboard interface need the ability to remap the user agent's keyboard shortcuts in order to avoid keystroke conflicts with assistive technology, reduce number of keystrokes, use familiar keystroke combinations, and optimize keyboard layout (e.g. for one-handed use). This is important for people with dexterity issues where every keystroke can be time consuming, tiring or painful. It is also important for people using assistive technologies such as screen readers, where many keystrokes are already in use by the assistive technology. The goal of this success criterion is to enable the user to be in control of what happens when a given key is pressed, use the keyboard commands that meet specific needs, and save the modifications.

Content authors can use the Accesskey attribute to define shortcut keys that allow quick access to specific elements, actions, or parts of the web content. The author can select shortcuts that are different from what the user expects. Users who rely upon keyboard input may want consistent shortcut keys across the sites they visit.

Users should have the option to make any keyboard shortcuts be persistent across browsing sessions. The user should be able to save, import and export these settings.

User agents can also offer the user the option to automatically apply preferred key combinations for content that has author-specified accesskey bindings that are based on the associated text, label, or ARIA role. This overrides any author-specified keybinding.

**Examples for Success Criterion 2.3.5:**

* Frank has repetitive strain injuries and uses speech input. He has defined standard commands to access commonly used parts of a website. For example, when Frank says "site search", the cursor jumps to the search field. If the author has assigned an access key that is not standard, Frank can override it and set the access key to a standard mapping so he can say the same search command across all the sites he uses.
* Laura types with one hand and finds keys on the left side of the keyboard easier to press. She browses to a web page and notices that the author has assigned access keys using keys from the right side of the keyboard. She opens a dialog in the user agent and reassigns the access keys from the web page to the left side of the keyboard. She also uses the user agent's preference settings to redefine its keyboard shortcuts in the same way.
* Laura types with one hand. On her mobile device, Laura maps common website actions to numeric shortcut keys. For example, she prefers to have the 1 key to activate a site's search function. An author of a site visited daily by this user defines "S" as the accesskey for the search function. Laura overrides the author-specified accesskey of "S" with "1".
* Elaine has low vision and is using a screen magnification program. The program uses Alt+M to increase the size of the magnified area of the screen. She notices that in her web browser, Alt+M is a access key for activating a home button that stops her from being able to control her magnification software. She opens a access key reassignment feature in the user agent, and sets Alt+O to be the new access key for the home button. Her screen magnification software now works correctly.
* George uses a screenreader. The screenreader uses Ctrl+F to read the item with focus. Since this is a common user agent command for Find, the user agent allows George to reassign the Find command to a non-conflicting key binding. The user agent provides a list of user interface features and default keyboard assignments with options for the George to assign new key combinations. George saves the user keyboard customizations the same way he saves other user preferences.

**Related Resources for Success Criterion 2.3.5:**

* [Guideline 2.7 - Configure and store preference settings](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-store-prefs)

Implementing Guideline 2.4 - Provide text search [[Guideline 2.4](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-search-text)] 提供文本搜索

**Summary**: Users can search rendered content (2.4.1, Level A) forward or backward (2.4.2, Level A) and can have the matched content highlighted in the viewport (2.4.3, Level A). The user is notified if there is no match (2.4.4, Level A). Users can also search by case and for text within alternative content (2.4.5, Level AA).

**2.4.1 Text Search:文本搜索**

The user can perform a search within [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content), including rendered text alternatives and rendered generated content, for any sequence of printing characters from the [document character set](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-doc-char-set). (Level A)

[Return to 2.4.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_241)

**Intent of Success Criterion 2.4.1:**

The find or text search function in a user agent allows the user to easily locate desired information in rendered content. People who read or navigate slowly or with difficulty due to a disability rely more heavily on the ability to search for text, rather than scanning or reading an entire document to find it. The ability to search alternative content allows screen reader users to find content they heard on their speaker. Users with hearing impairments use Text Search as an efficient method of jumping to specific points in a video. Users who find it difficult to use the mouse or keyboard and have to limit their physical operations will save movements using search.

**Examples for Success Criterion 2.4.1:**

* Marvin has a dexterity impairment. He needs to move efficiently to specific text in the document.  The user agent provides a local search function that is available using speech commands. Marvin says "find box," a text box with a search button appears. He speaks the word he is looking for, and says "enter", which executes the search function.
* Betty, who has low vision, is attempting to create a user stylesheet for a site. She needs to know the 'class' attribute value for navigation headers. Betty gets the source view of the current page and searches for the specific phrase used in a navigation list to find the class associated with  the navigation element
* Joe, a user with a distraction disorder, is taking an online exam. He is working on the 6th question when he realizes he wrote something wrong in the essay on question 2. He uses the search function in the browser to find the text in error inside the textarea of question 2.
* Sam is a screen reader user. He has images off and the alternative content for images  is revealed. He wants to send the flow chart image on the page to a colleague. Sam searches for the word "flowchart" that he heard spoken as part of the 'alt' text for the image. He then uses the context menu to select the address of the image and sends it to a colleague.
* Agnes is deaf. She is watching a video with captions turned on. Agnes uses the search function to search through the captions and jump to the point in the video where the search term is located in the time line.
* Greta has a reading impairment. She is trying to efficiently locate some information in a large, detailed graphic. She is using a browser with native support for SVG. When Greta searches for a term, text within the SVG is searched along with the HTML content.

**Related Resources for Success Criterion 2.4.1:**

* None

**2.4.2 Search Direction: 搜索导向**

 The user can search forward or backward in rendered content. (Level A)

[Return to 2.4.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_242)

**Intent of Success Criterion 2.4.2:**

People who read slowly or with difficulty due to a disability rely more heavily on the ability to search for the text they're looking for, rather than scanning or reading an entire document to find it. Local find in a user agent allows the user to easily locate desired information in rendered content. The ability to search for alternative text content allows screen reader users to find content they heard on their speaker. Users with hearing impairment use text search as an efficient method of jumping to specific points in a video.

**Examples for Success Criterion 2.4.2:**

* A user has been reading through a web page and wants to quickly locate a phrase previously read. When opening the browser's page search feature, the user has options to search forward and backward from the current location. If the search reaches an endpoint in the document, the user is notified that the search has wrapped around, such as with an alert box or other indication.

**Related Resources for Success Criterion 2.4.2:**

* None

**2.4.3 Match Found: 匹配发现**

When a search operation produces a match, the matched content is [highlighted](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-highlight), the viewport is scrolled if necessary so that the matched content is within its visible area, and the user can search from the location of the match. (Level A)

[Return to 2.4.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_243)

**Intent of Success Criterion 2.4.3:**

It is important for the user to easily recognize that a search term has been found and that the term is revealed to the user in context. The user agent moves the viewport to include the found term and the term is highlighted in some fashion.The point of regard is the found element in the viewport. Any subsequent searches on the same term or other navigation tasks (e.g tabbing to the next anchor) begin from this point.

**Examples for Success Criterion 2.4.3:**

* Jules has low vision and uses a magnified screen. She frequently searches for terms that appear multiple times in a document that contains a lot of repetition. It is important that the viewport moves and if necessary her screen scrolls after each search so she can easily track where she is in the document.

**Related Resources for Success Criterion 2.4.3:**

* None

**2.4.4 Alert on Wrap or No Match:**

The user can choose to receive [notification](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-alert) when there is no match to a search operation. The user can choose to receive notification when the search continues from the beginning or end of content. (Level A)

[Return to 2.4.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_244)

**Intent of Success Criterion 2.4.4:**

It is important for users to get clear, timely feedback so they don't waste time waiting or, worse, issue a command based on a wrong assumption. It is important during a search that users are informed when there is no match or that the search has reached the beginning of the document.

**Examples for Success Criterion 2.4.4:**

* Dennis uses a screen reader. As soon as he gets a message that there is no match he goes on to search for something else, however, if he does not get a message he wastes time retrying the search to make sure the entire document has been searched.
* Dennis uses a screen reader. When he searches for a word in a web page and the search wraps to the beginning of the page, his computer beeps to alert him that the search has wrapped.

**Related Resources for Success Criterion 2.4.4:**

* None

**2.4.5 Alternative Content Search: 可替换的内容搜索**

The user can perform text searches within alternative content that is text (e.g. [text alternatives for non-text content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content), captions) even when the alternative content is not rendered onscreen. (Level AA)

[Return to 2.4.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_245)

**Intent of Success Criterion 2.4.5:**

Authors frequently provide alternative content to meet web content accessibility guidelines. Users with disabilities may experience this as part of the content. The purpose of this success criteria is to ensure that text search allows users to locate this content, even if it is not visibly rendered.

**Examples for Success Criterion 2.4.5:**

* Rhonda typically browses the web with images turned off so she sees alternative text rendered in place of images. She visits a favorite website on a friend's computer and wants to find a link to her favorite artist's photo gallery. She searches for the artist's name on her friend's computer. The alt text is not displayed, but the artist's image is highlighted because her 'text search' command has associated the alt text with the image.
* Dennis remembers a phrase in a captioned training movie that he wants to replay. He goes to the web page with the embedded movie. He types in the phrase in the "Find" box and the user agent moves to the point in the movie where the phrase is found.

**Related Resources for Success Criterion 2.4.5:**

* None

Implementing Guideline 2.5 - Provide structural navigation [[Guideline 2.5](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-nav-structure)] 提供结构导航

**Summary**: Users can view (2.5.1, Level AA), navigate (2.5.2, Level A), and configure the elements used in navigating (2.5.3, Level AAA) content hierarchy.

**2.5.1 Show Location in Hierarchy: 显示位置的层次结构**

When the user agent is presenting hierarchical information, but the hierarchy is not reflected in a standardized fashion in the [DOM](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-dom) or [platform accessibility services](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch), the user can view the path of nodes leading from the root of the hierarchy to a specified element. (Level AA)

[Return to 2.5.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_251)

**Intent of Success Criterion 2.5.1:**

Many users rely on assistive technology to interpret information and convey it in different ways. Sometimes when a user agent displays different pieces of information, it understands relationships between them that can't easily be conveyed through either the DOM or platform accessibility services. This makes it difficult or impossible for assistive technology to infer and interpret those relationships. In these cases the user agent should present those relationships in a simplified way that can be understood by the assistive technology, and by users who have difficulty interpreting complex visual presentations. This is also required when the hierarchy is included in the DOM in a non-standardized fashion (proprietary attributes or distributed extensibility microdata) because such proprietary information would not be useful to assistive technology.

**Examples for Success Criterion 2.5.1:**

* Armand is blind. His media player uses an XML data structure that describes the genre, artist, album, and track of each audio piece. It displays a column for each category. He can select one entry to filter the contents displayed in the subsequent columns (e.g. an entry in the Artist column causes the Albums and Tracks columns to only show items associated with the selected artist). The user agent understands the hierarchical nature of the columns, but since this hierarchical data cannot be conveyed in the HTML used for presentation, it is not in the DOM for access by his screen reader. When Armand moves the focus to an item in the Tracks column, the user agent updates a static text field that shows the path of genre, artist, and album associated with the track, and Armand can have this field read to him on request or automatically when it updates.

**Related Resources for Success Criterion 2.5.1:**

* None

**2.5.2 Provide Navigation by Heading and within Tables: 提供通过标题的导航以及表单内的导航**

The user agent provides at least the following types of [structural navigation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-structural-navigation), where the structure types exist: (Level AA)

* By heading
* Within tables

[Return to 2.5.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_252)

**Intent of Success Criterion 2.5.2:**

Users who find it difficult or impossible to use the mouse require an efficient way to jump among elements without having to navigate through intervening content. Navigating by heading is especially important when scanning a webpage to find a pertinent section. Navigating by table element is especially important when building or reading tables.

**Examples for Success Criterion 2.5.2:**

* Jamie is blind. When he reads the New York Times he scans the headlines to find interesting stories. To do this he needs a way to go from headline to headline.
* Billie is a paraplegic who uses speech input to write long legal documents. When she is proofing a document she needs to scan each section header. It takes far fewer speech commands to navigate the section headers when she can jump directly to them. Billie also makes extensive use of tables. It takes far fewer speech commands to navigate tables when she can jump directly among elements. Direct navigation to headlines and table elements allows her to do her job without overusing her vocal cords and within required time constraints.
* Celia has short-term memory issues and is easily distracted. When looking for any particular section of the document she finds it easier to scan through headings by jumping from heading to heading rather than having to scan through an entire page of potentially distracting text. Celia also finds it useful to be able to move from a subheading to a major heading and back to orient herself within the context without becoming confused.
* Armand is blind. When he uses the speech input to locate a web page on his smartphone, Armand navigates from heading to heading using touch commands.

**Related Resources for Success Criterion 2.5.2:**

* None

**2.5.3 Allow Elements to be Configured for Structural Navigation: 允许元素为结构导航而配置**

The user can configure a set of [important elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-important-elements) (including [element type](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element-type)) for structured navigation and hierarchical/outline view. (Level AAA)

[Return to 2.5.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_253)

**Intent of Success Criterion 2.5.3:**

Authors can visually convey relationships between elements by grouping them spatially, or giving them the same coloration or background. Users may not be able to perceive those attributes when using a screen reader, or when strong magnification makes it difficult to make a mental model of the screen layout. In cases like these, the user agent can assist by providing a view of the data that groups elements that that user agent perceives as implying relationships.

Often the user agent will choose by default the elements it considers important for structured navigation. These may not be relevant in all circumstances. For example, the user may wish to navigate via informal mechanisms such as microformats or or by a particular element.

**Examples for Success Criterion 2.5.3:**

* Fred is blind and wishes to navigate through the menu structure using the Tab key, but the menu is a set of nested list elements with an HTML class attribute denoting the menu-submenu relationship. Because Fred's user agent allows him to configure important elements, he can explicitly include the class name as an important element for navigation. He then assigns a keyboard shortcut to navigate to the next element with the same class name as the element that has the focus.
* When Fred is using his smartphone, he selects a control that allows him to change from navigation by heading to navigation by links.

**Related Resources for Success Criterion 2.5.3:**

* None

Implementing Guideline 2.6 - Provide access to event handlers [[Guideline 2.6](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-device-independent-handlers)]

**Summary**: Users can interact with web content by mouse, keyboard, voice input, gesture, or a combination of input methods. Users can discover what event handlers (e.g. onmouseover) are available at each element and activate an element's events individually (2.6.1).

**2.6.1 Allow Access and Activation of Input Methods: 允许访问并且激活输入方式**

The user agent provides a means for the user to determine [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) input methods explicitly associated with an element, and a means for the user to activate those methods in a modality independent manner. (Level AA)

[Return to 2.6.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_261)

**Intent of Success Criterion 2.6.1:**

Users interacting with a web browser may use one or more input technologies including keyboard, mouse, speech, touch, and gesture. Sometimes a web page is scripted so that it assumes and requires an input method that is not available to the user, such as handling drag events for a user relying solely on the keyboard. In those cases, the user needs to determine what methods are available and activate that element with a modality independent method.

In addition, any one input method should not hold back another. For instance, people who don't use a mouse shouldn't have to map their input methods to the same steps a mouse user would take. The user agent can constrain steps such as not allowing a mouseup before a mousedown.

**Examples for Success Criterion 2.6.1:**

* Jeremy cannot use a mouse. He needs to activate a flyout menu that appears when he hovers the mouse over a button. He navigates to the button and pulls up its context menu, which includes an "Inputs" submenu that lists the available input methods (e.g. events that were registered for this element: click, double-click, swipe left, and hover). Then he chooses the "Hover" command.
* Ken is a speech input user. In order to get his work done in a reasonable amount of time and without overtaxing his voice he uses a single speech command phrase to move the mouse up, left and click.
* Karen cannot use a mouse. She clicks a single key to activate both events for link that has an onmousedown and an onmouseup event link.
* Ingrid has low vision. When navigating a page with a smartphone, she can choose a bluetooth keyboard or gestures to operate all of the controls within the page.

**Related Resources for Success Criterion 2.6.1:**

* [Modality Independent Controls](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#intro-modality-independence)
* [1.3.1 Highlighted Items](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_131)

Implementing Guideline 2.7 - Configure and store preference settings [[Guideline 2.7](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-store-prefs)]

**Summary**: Users can restore preference settings to default (2.7.2, Level A), and accessibility settings persist between sessions (2.7.1, Level A). Users can manage multiple sets of preference settings (2.7.3, Level AA), and adjust preference setting outside the user interface so the current user interface does not prevent access (2.7.4, Level AA), and transport settings to compatible systems (2.7.5, Level AA).

**2.7.1 Allow Persistent Accessibility Settings: 允许持续性的无障碍设定**

User agent accessibility preference settings persist between sessions. 用户代理的无障碍设定可以会话之间保留(Level A)

[Return to 2.7.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_271)

**Intent of Success Criterion 2.7.1:**

When a user has customized settings within the user agent to maximize accessibility, customization is saved between browsing sessions. The user can automatically use those settings in subsequent browsing sessions.

**Examples for Success Criterion 2.7.1:**

* Lynn has moderately low vision. She sets the default zoom level, font size, and colors to make pages easier for her to read. Because those settings are persistent, she doesn't have to manually restore her settings every time she starts the browser.
* Brian has to adjust settings in his browser to make it fully compatible with his speech input system. It's difficult for him to adjust these settings, since he can't fully operate the browser until it's complete. Once his browser is configured this way, he relies on it staying in that configuration even if he upgrades the browser or restarts his system.
* Betty has low vision. She customizes her mobile browser's color and font settings to make text much easier to read. Her browser incorporates a cloud-based profile so she can retain her settings across her browsing sessions and her desktop and tablet browsers.

**Related Resources for Success Criterion 2.7.1:**

* None

**2.7.2 Allow Restore All to Default: 允许把所有设定恢复到默认**

The user can restore all preference settings to default values. (Level A)

[Return to 2.7.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_272)

**Intent of Success Criterion 2.7.2:**

For some users, it may be difficult to easily recall all modified settings. Others may find it difficult to navigate to each modified setting, especially if a particular setting impacts their ability to do so. Users who customize settings may find that their chosen settings are not suitable and decide to restore these settings to default values. This success criteria allows a user to easily restore all preference settings to default values using a single function or action.

**Examples for Success Criterion 2.7.2:**

* Ron is blind. He accidentally changes a browser setting that makes his browser incompatible with his screen reader, preventing him from changing it back. He restarts his browser using a command line option that starts in the default configuration. From there he can adjust the setting that caused him problems.
* Kathy has repetitive stress injuries which makes it painful to experiment with settings. She accidently turns on a zoom feature on her smartphone and cannot figure out how to turn it off. She gestures to navigate to the preferences menus and selects a command to reset preferences to default.

**Related Resources for Success Criterion 2.7.2:**

* [3.2.5 Settings Change Confirmation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_325)

**2.7.3 Allow Multiple Sets of Preference Settings: 允许多组性能设定**

The user can save and retrieve multiple sets of user agent preference settings. 在用户代理的表现设定中，用户可以存储和恢复多组设定(Level AA)

[Return to 2.7.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_273)

**Intent of Success Criterion 2.7.3:**

Some users may need to change their setting preferences under circumstances such as varying levels of user fatigue, changes in environmental noise, or changing lighting conditions. Providing an easy method for saving and switching between sets of preferences helps the user complete intended tasks.

**Examples for Success Criterion 2.7.3:**

* Hiroki has low vision. When he is carrying his tablet computer he operates it with the built-in touchscreen. When at his desk he links it to a Bluetooth keyboard and mouse, and redirects the display to a large computer monitor. The browser allows him to quickly switch between different configurations for different environments.
* Davy has moderately low vision and must adjust the contrast of media to different levels for day and night. Because this requires a multiple steps, he has two user profiles, one for each environment.
* Aaron has repetitive strain injury. He usually uses a keyboard and mouse, but when his repetitive strain injury is bothering him he prefers to use the keyboard and avoid using the mouse as much as possible. At those times he uses his browser's user preference profiles to load a different configuration that’s optimized for the keyboard.

**Related Resources for Success Criterion 2.7.3:**

* None

**2.7.4 Allow Preference Changes from outside the User Interface:**

The user can adjust any preference settings required to meet the User Agent Accessibility Guidelines (UAAG) 2.0 from outside the user agent user interface. (Level AAA)

[Return to 2.7.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_274)

**Intent of Success Criterion 2.7.4:**

Users with disabilities may not be able to use a user agent in a particular configuration. This can occur during setup when default settings don't meet their needs, or after someone changes an option. If users cannot change the settings from the user interface, they need a way to adjust or reset those options from outside the user agent. The user agent can accomplish this in multiple ways including detecting and implementing the platform accessibility settings, providing an external file to modify, providing access to settings from a separate utility program, providing accessibility options in the installation program, or providing command-line switches to change the user agent's behavior.

*Note:*User agents are encouraged to allow all user preferences to be adjusted.

**Examples for Success Criterion 2.7.4:**

* Aosa is blind. Her web browser is incompatible with her screen reader unless she changes one of the browser's advanced settings. Because she cannot activate and use the appropriate dialog box until the setting is changed, she needs to start the browser using a dedicated command-line switch that changes the behavior.
* Bintu is deaf and enjoys watching captioned videos. Since different video players may not have accessible settings, she sets her browser to always display captions, knowing the video player will respect the browser's request to display captions. In this way the options in one user agent (the embedded video player) are set outside its user interface by being set in another program (the hosting browser).
* Sasha requires high contrast to be able to discriminate the shape of letters. She has set the accessibility preferences on her computer to use high contrast mode. When she launches her browser, it detects that she is using high contrast and adjusts the font and color settings for its user interface to reflect those settings.
* Justin has an attention deficit disorder. He is setting up his new e-book reader and is interrupted while setting the default font colors. He accidentally sets his background and font color to white on white. He can't read the settings screen to recover his default settings, so he exits the reader and follows the instructions on the vendor's website to edit the "settings.ini" file to adjust the colors. He restarts the reader with the corrected color settings.
* Jan is easily confused by new interfaces. Using the screen reader capabilities on her mobile phone she changes the interface of the updated browser, then can't figure out how to undo them. She uses an app from the browser developer to reset the browser settings to default.

**Related Resources for Success Criterion 2.7.4:**

* None

**2.7.5 Make Preference Settings Transferable: 偏好设置可转移**

The user can transfer all compatible user agent preference settings between devices. 用户可以在不同设备之间转移所有可兼容的用户代理表现设定(Level AAA)

[Return to 2.7.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_275)

**Intent of Success Criterion 2.7.5:**

Configuring a user agent may be a complex and time-consuming task. Some users hire assistive technology professional trainers to do their system setup. Users who have spent time customizing accessibility preferences to meet their requirements need to migrate preference setting to other compatible devices. Schools and universities also need to maintain accessibility settings across multiple machines.

**Examples for Success Criterion 2.7.5:**

* Lori is a rehabilitation specialist who works with clients to customize electronics and assistive technology to their needs. She sets up a new system for Ray, and saves a backup of the accessibility setting for Ray and for herself. She keeps electronic backups of her clients' configurations that she can quickly email to them if they lose their settings. 可为用户保存无障碍设定，当用户丢失可以通过email方式找回其设定
* Ray is blind and uses a screenreader on his desktop and a voice application on his phone. He sets up web-based email application accessibility preferences on his phone. The preferences are automatically reflected in the desktop version of that web-based email application.
* Trisha is a 4th grader with low vision. When she changes classrooms, she carries a USB stick containing her accessibility settings so she and her teachers do not have to use class time customizing her computer.
* Betty is a low vision user and has a highly customized color palette defined in her browser. She saves her customizations to a cloud-based storage service, so her preferences can be transferred to the other desktop and mobile browsers that she uses.

**Related Resources for Success Criterion 2.7.5:**

* None

Implementing Guideline 2.8 - Customize display of graphical controls 自定义图形显示控件[[Guideline 2.8](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-configure-controls)]

**Summary**: It's recommended that users can add, remove, reposition, and assign shortcuts to user agent controls, and restore them to their default settings (2.8.1, Level AA).

**2.8.1 Customize Display of Controls for User Interface Commands, Functions, and Extensions: 用户界面命令、功能和扩展的显示界面可自定义**

The user can customize which user agent commands, functions, and extensions are displayed within the user agent user interface as follows: (Level AA)

* **Show:** The user can choose to display any controls available within the user agent user interface, including user-installed extensions. It is acceptable to limit the total number of controls that are displayed onscreen.
* **Simplify:** The user can simplify the default user interface by choosing to display only commands essential for basic operation (e.g. by hiding some controls).
* **Reposition:** The user can choose to reposition individual controls within containers (e.g. toolbars or tool palettes), as well as reposition the containers themselves to facilitate physical access (e.g. to minimize hand travel on touch screens, or to facilitate preferred hand access on handheld mobile devices).
* **Assign Activation Keystrokes or Gestures:** The user can choose to view, assign or change default keystrokes or gestures used to activate controls.
* **Reset:** The user has the option to reset the containers and controls to their default configuration.

[Return to 2.8.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_281)

**Intent of Success Criterion 2.8.1:**

The user needs to control which user interface elements are visible and usable, where elements are visually located on the screen, and where elements fall in the navigation order. In some cases adjusting whether an element is visible and usable may involve installing or uninstalling a component — or merely showing or hiding it — depending on the user agent and the specific component.

This can reduce keystrokes, bring buttons into view that are hidden by default or otherwise allow the user to interact with the user agent in a more efficient fashion. Users with dexterity impairments or mobility impairments may have problems making the large movements required to select between non-adjacent controls which they need to use frequently. Similarly, users with low vision may have to move their magnified view-port excessively to see frequently used controls. Enabling these controls to be situated together removes some of the strain faced by these users, and increases productivity as task completion times are decreased.

**Examples for Success Criterion 2.8.1:**

* Martin accesses the computer by pressing keys with a stick held in his mouth. He gets around the user agent with taps on the tab and arrow keys. The button for printing a web page is the last button on a toolbar. This button requires six presses of a right arrow key. This is the only button Martin uses on the toolbar. Using a preferences dialog, Martin is able to configure this toolbar to only show the Print button, reducing the number of time he must press with his mouthstick.
* Laura has one hand. When she holds her mobile phone in her left hand, she must use her thumb to press the controls. She configures her mobile apps so that the toolbars are at the left side or the bottom, so she can reach them.
* Caraway has a repetitive strain injury. She uses speech input. The speech program automatically clicks toolbar items that she speaks. In programs she uses a lot she removes toolbars that she doesn't use in order to reduce the probability that the speech program will interpret text input as a toolbar command and click something Caraway does not intend.
* Zelda has a brain injury that leaves her easily confused. She reorders the toolbars in her web-based word processing and layout programs so that the text color and highlight color icons are in the same order and she can rely on habit to click the correct button no matter what program she is in. Without this ability to reorder she finds herself constantly clicking the wrong button because they are configured differently by default in the different programs.
* Devon is easily distracted. In programs he uses frequently, he removes toolbars he does not use in order to cut down on distractions.
* Sally has memory issues that make it difficult to memorize keyboard shortcuts. She has aligned the keyboard shortcuts in several programs so she doesn't have to memorize as many shortcuts. She reminds herself of keyboard shortcuts by hitting a key that gives her a list of keyboard shortcuts in the current program.
* Linda has rheumatoid arthritis and finds it difficult to perform the pinch gesture that's commonly used to zoom on mobile phones. She changes the default gesture for zooming to a gesture she can more easily do. Linda's left hand is less damaged than her right hand. She moves a common control from the right side of the screen to the left side of the screen to make it easier to access with her left hand.
* Jennifer is blind. She sometimes configures apps on her friend Linda's mobile phone. When Jennifer picks up Linda's mobile phone, she turns on the built-in voice application so she so she can quickly find her way around Linda's phone. When Jennifer is done, she changes the controls back to Linda's original settings.

**Related Resources for Success Criterion 2.8.1:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

Implementing Guideline 2.9 - Allow time-independent interaction 允许不随时间变化的交互 [[Guideline 2.9](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-time-independent)]

**Summary**: Users can extend the time limits for user input when such limits are controllable by the user agent (2.9.1, Level A).

**2.9.1 Adjustable Time Limits: 时间限制可调节**

Where time limits for user input are [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) and controllable by the user agent, the user can extend the time limits. (Level A)

[Return to 2.9.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_291)

**Intent of Success Criterion 2.9.1:**

People who use assistive technology and those who require more time to read, understand, or act upon content (e.g. individuals with reading disabilities or non-native readers of the presented language) should be able to extend or override any content or author imposed presentation or interaction time limits.

**Examples for Success Criterion 2.9.1:**

* Lee has low vision and uses a screen magnifier. She goes to a news organization website with a home page region that cycles featured stories every 3 seconds. Lee requires more than three seconds to read the news item and select it. Lee uses a global setting to freeze all timed events. She uses another keyboard command to resume the timed presentation.
* Jennifer is blind and uses a screen reader. She is logged into a financial services website and is reading the site's detailed privacy policy. The site terminate sessions that have been inactive for 5 minutes. A prompt appears warning that Jennifer will be logged off. Jennifer selects an option in her browser that automatically responds to those prompts.

**Related Resources for Success Criterion 2.9.1:**

* None

Implementing Guideline 2.10 - Help users avoid flashing that could cause seizures 帮助用户避免可能会引起癫痫发作的闪烁 [[Guideline 2.10](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-prevent-flash)]

**Summary**: To help users avoid seizures, the default configuration prevents the browser user interface and rendered content from flashing more than three times a second above luminescence or color thresholds (2.10.1, Level A), or even below the thresholds (2.10.2, Level AAA).

**2.10.1 Three Flashes or Below Threshold: 闪光三次或低于**阈值

In its default configuration, the user agent does not display any user interface components that flashes more than three times in any one-second period, unless the flash is below general flash and red flash thresholds. (Level A)

[Return to 2.10.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2101)

**Intent of Success Criterion 2.10.1:**

Seizures due to photosensitivity can occur when there is a rapid series of general flashing, or a red flash. A potentially harmful flash occurs when there is a pair of significantly opposing changes in luminance. A transition to or from a saturated red is potentially harmful irrespective of luminance.

**Examples for Success Criterion 2.10.1:**

* Jamie has photosensitive epilepsy. He checks a game website that has just added a flashing red icon to advertise their newest product. Because his browser's default configuration protects him from harmful flashing, the potentially harmful flashing does not occur on his machine.
* Jamie visits a weather web site that uses flashing to indicate a tornado warning. In order to avoid triggering seizures, the flashing is limited to fewer than three times per second, and, to be extra cautious, it is not red.

**Related Resources for Success Criterion 2.10.1:**

* [ITC Guidance Note for Licensees on Flashing Images and Regular Patterns in Television](http://stakeholders.ofcom.org.uk/binaries/broadcast/guidance/gn_flash.pdf)
* [WCAG 2.0: 2.3.1 Three Flashes or Below Threshold](http://www.w3.org/TR/2008/REC-WCAG20-20081211/#seizure-does-not-violate)

**2.10.2 Three Flashes: 闪光三次**

In its default configuration, the user agent does not display any user interface components that flashes more than three times in any one-second period (regardless of whether not the flash is below the general flash and red flash thresholds). (Level AAA)

[Return to 2.10.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2102)

**Intent of Success Criterion 2.10.2:**

Seizures due to photosensitivity can occur when there is a rapid series of general flashing, or a red flash. People who are particularly sensitive to flashing can be harmed by any level of flashing. 2.10.2 has the same effect as 2.10.1, going further to ensure that more sensitive users can traverse the web without potentially harmful effects.

**Examples for Success Criterion 2.10.2:**

* Alfred has photosensitive epilepsy and is particularly sensitive to flashes. He checks a game website that has just added a flashing red icon to advertise their newest product. His browser's default configuration protects him from all flashing, so the potentially harmful flashing does not occur on his machine.

**Related Resources for Success Criterion 2.10.2:**

* [WCAG 2.0: 2.3.2 Three Flashes](http://www.w3.org/TR/2008/REC-WCAG20-20081211/#seizure-three-times)
* [2.11.1 Time-Based Media Load-Only](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_2111)
* [2.11.5 Stop/Pause/Resume Time-Based Media](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_2115)

Implementing Guideline 2.11 - Provide control of time-based media 为时基媒体提供控制[[Guideline 2.11](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-control-inaccessible-content)]

**Summary**: The user can present placeholders for time-based media (2.11.1, Level A) and executable regions (2.11.2, Level A), or block all executable content (2.11.3, Level A), adjust playback (2.11.4, Level A), stop/pause/resume (2.11.5, Level A), navigate by time (2.11.6, Level A) or semantic structures such as chapter (2.1.7, Level AA), enable or disable tracks (2.11.8, Level AA), and adjust contrast and brightness of visual time-based media (2.11.9, Level AAA).

**2.11.1 Time-Based Media Load-Only: 仅加载的时基媒体**

The user can override the play on load of [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) time-based media content such that the content is not played until [explicit user request](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-request). (Level A)

[Return to 2.11.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2111)

**Intent of Success Criterion 2.11.1:**

Users who need to avoid signals that may trigger seizures, users who are easily distracted, and users who have difficulty interacting with the controls provided for playing media need to be able to load media in a paused state. The user agent provides a global control that sets a state equivalent to "paused waiting for user interaction" for all recognized media when a page loads. The user has a global option to set autoplay to off or paused until the user activates "play". The user agent provides a visual or auditory indicator that the video is in paused state and needs user interaction to start. This prevents media from playing without explicit request from the user.

When the media is not in the actual document, but rather has been created with document.createElement('audio'), the user agent does not recognize that the media exists and cannot give a visual indication by default. In this case, it is up to the author to provide the controls. See WCAG in resources below.

**Examples for Success Criterion 2.11.1:**

* Jill is blind. She browses the web using a screen reader to listen to the text of web pages. She navigates to her favorite shopping site and is greeted with trumpets blaring and an announcer shouting "Sale, sale, sale!" The audio is so loud that she can no longer hear the web page content. Jill closes her browser and changes a setting titled Play Audio on Request to yes and visits her shopping site again. This time she can read the content and when she is ready plays the audio and smiles, thinking of the deal's she is about to find.
* Jamie has epilepsy that's triggered by certain types of audio. She sets her browser so content does not play automatically so she can avoid audio that could trigger her epilepsy.
* Kendra has photo-epilepsy. She sets her browser so content does not play automatically so she can avoid flashing content that could trigger her photo epilepsy.

**Related Resources for Success Criterion 2.11.1:**

* [Guideline 2.10 - Help users avoid flashing that could cause seizures](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-prevent-flash).
* [HTML5 4.8.10.8 Playing the media resource](http://www.w3.org/TR/2012/CR-html5-20121217/embedded-content-0.html#playing-the-media-resource)
* [WCAG 2.0 - 1.4.2 Audio Control](http://www.w3.org/TR/WCAG20/#visual-audio-contrast)

**2.11.2 Execution Placeholder: 执行占位符**

The user can request a [placeholder](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-placeholder) instead of executable content that would normally be contained within an on-screen area (e.g. Applet, Flash), until [explicit user request](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-request) to execute. (Level A)

[Return to 2.11.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2112)

**Intent of Success Criterion 2.11.2:**

Documents that do things automatically when loaded can delay, distract, or interfere with user's ability to continue with a task. Replacing executable content like embedded objects, applets and media with a placeholder tells the user what has been blocked and provides a mechanism (e.g. a play button) for unblocking when the user is ready.

*Note:*A placeholder should take up the same space as the object it is replacing, so that the presentation doesn't need to be reflowed when the execution is started. However, people using mobile devices or screen enlargers, or those who have difficulty with scroll commands may benefit from having the option of a smaller placeholder.

**Examples for Success Criterion 2.11.2:**

* Jane has difficulty concentrating. In order to concentrate on the text of a document she hides multimedia content and only triggers execution of that content by clicking on the placeholder.
* Evan is blind. He sets the option in his browser so that when a web page loads it does not automatically run an executable object. This way any or music or speech they play won't interfere with his ability to hear his screen reader. When he is ready to hear it, he navigates to the placeholder and presses the Enter key to activate it.
* Evan has configured his mobile phone so that any audio or video file displays a placeholder with a triangle "play" icon. That allows him to control when the audio or video starts.

**Related Resources for Success Criterion 2.11.2:**

* None

**2.11.3 Execution Toggle: 执行触发**

The user can turn on/off the execution of dynamic or executable content (e.g. Javascript, canvas, media). 用户可以打开、关闭动态或者可执行内容的执行（例如，Javascript, canvas, media） (Level A)

[Return to 2.11.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2113)

**Intent of Success Criterion 2.11.3:**

Documents that do things automatically when loaded can delay, distract, or interfere with user's ability to continue with a task. The user needs to be able to specify that executable content (e.g. scripts) be blocked when a document loads, be told which content has been blocked, and be able to selectively execute the content at a later time.

*Note:*Although some web applications and document may be empty until scripts are run, it is important for users to have this level of control.

**Examples for Success Criterion 2.11.3:**

* Jane has difficulty concentrating. In order to concentrate on the text of a document she wants to prevent animations, media, or dynamic content from executing until she is ready. An icon on the status bar tells her that scripts have been blocked. She can click on it to select which scripts to run.
* Evan is blind. He sets an option in his browser so that when a web page loads it does not automatically start running scripts that might play sounds that would interfere with his ability to hear his screen reader. An icon on the status bar tells him that scripts have been blocked. He can activate it to select which scripts to run.

**Related Resources for Success Criterion 2.11.3:**

* None

**2.11.4 Adjustable Playback Rate for Prerecorded Content: 预录内容的重放速率可调整**

The user can adjust the playback rate of prerecorded time-based media content, such that all of the following are true: (Level AA)

* **Playback Rate:** The user can adjust the playback rate of the time-based media tracks to between 50% and 250% of real time.
* **Pitch:** Speech whose playback rate has been adjusted by the user maintains pitch in order to limit degradation of the speech quality.
* **Synchronization:** Audio and video tracks remain [synchronized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-synchronize) across this required range of playback rates.
* **Reset:** The user agent provides a function that resets the playback rate to normal (100%).

[Return to 2.11.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2114)

**Intent of Success Criterion 2.11.4:**

Users with sensory and cognitive impairments may have difficulty following or understanding spoken audio at the normal playback rate. If users can slow down the audio presentation of speech while maintaining the pitch or frequency characteristics, they are better able to follow the spoken content. Users with learning disabilities may be distracted or otherwise unable to follow complex animations or instructional video. With the presentation slowed, users can better observe the visual events of the animation. User may also want to slow down the media if they are taking notes, and do so slowly because of language or dexterity impairments.

Some users with visual impairments prefer faster speech rate on their screen readers or digital audio book players. The ability to speed up the audio while maintaining pitch allows those users to skim spoken audio without loss of understandability.

**Examples for Success Criterion 2.11.4:**

* Timo experienced a traumatic brain injury and has difficulty comprehending speech. When listening to episodes of his favorite podcast on the web, he slows down the audio by 50% and is able to understand the interviewer's and guest's question and answer session.
* Anu is a blind university student who has grown up with digital talking book players, and regularly listens to spoken audio at 200% of normal speaking rate. In studying for exams, she reviews the online lecture videos from her History course, adjusting the presentation rate to 200% on the web video player in order to quickly review the material. She slows the presentation to normal rate when she encounters material she needs to review carefully.
* Perttu has a learning disability and requires a longer time to follow instructions. He likes to cook and is watching a cooking demonstration on the web. When the instructions go by too quickly, he slows the video player to half speed.

**Related Resources for Success Criterion 2.11.4:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

**2.11.5 Stop/Pause/Resume Time-Based Media: 时基媒体的停止、暂停、恢复**

The user can stop, pause, and resume rendered audio and [animation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-animation) content (e.g video, animation, changing text) that lasts three or more seconds at the default playback rate. (Level A)

[Return to 2.11.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2115)

**Intent of Success Criterion 2.11.5:**

Users with sensory, attentional, or cognitive impairments may have difficulty understanding multimedia content. When users are able to control the presentation rate of time-based media by stopping, pausing, and resuming, they have enough time to understand or act upon presented content, or to stop potentially distracting or harmful content.

**Examples for Success Criterion 2.11.5:**

* Adam reads more slowly than average because of his dyslexia. He's watching a video of a lecture. When the video shows slides, he presses the space bar to pause the video so that he can read the text at his own speed. When he's ready to continue, he presses the space bar again to resume the video.
* Angelica can have audio-induced seizures. She uses a website to watch and listen to user-contributed podcasts. When she realizes the level of white noise in a particular soundtrack is likely to trigger a seizure. She quickly clicks on the player's "stop" button (or presses the equivalent keyboard command) and the noise is instantly discontinued.
* Allesandro finds it impossible to ignore visual changes. Unnecessary animations make it very difficult for him to read or interact with other content on the screen. When he's reading an article on a newspaper website and finds an animated advertisement or moving text of a news ticker distracting, he chooses the appropriate command to stop all animations.
* Amaryllis is blind and is listening to streaming audio on a web page. When she responds to an incoming email message she uses a keyboard command to pause the audio, which would otherwise interfere with her ability to hear her screen reader.

**Related Resources for Success Criterion 2.11.5:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

**2.11.6 Navigation of Time-Based media by Time: 时基媒体通过时间的导航**

If time-based media lasts three or more seconds at the default playback rate, the user can navigate it using a continuous scale and by [relative time units](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-relative-time-units). (Level A)

[Return to 2.11.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2116)

**Intent of Success Criterion 2.11.6:**

Users with sensory, cognitive or attentional impairments may find it difficult to understand or follow time-based media. This success criteria allows users to position within the timebase to review content or to skip content that may be distracting.

**Examples for Success Criterion 2.11.6:**

* Jared has a print disability that makes it laborious to read text. He is watching a technical training video which will display section objectives or summary questions as text. When the text flashes by too quickly for him to read, he presses a key command to skip back an increment so he can read the text. He pauses the video if more time is required.
* Debbie has difficulty with bright or flashing video. When she encounters a flashing transition in a video, she quickly presses a key command to forward the video past the flashing, then carefully uses the slider to adjust the video back to the start of the next section so she can avoid the flashing material.

**Related Resources for Success Criterion 2.11.6:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

**2.11.7 Navigation of Time-Based Media by Semantics: 时基媒体通过语义的导航**

The user can navigate by semantic structure within the time-based media, such as by chapters or scenes present in the media. 用户可以通过时基媒体内的语义结构进行导航，比如通过媒体内的章节或者场景展示(Level AA)

[Return to 2.11.7 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2117)

**Intent of Success Criterion 2.11.7:**

Users need to be able to navigate time-based media in ways that are more meaningful than arbitrary time increments.

**Examples for Success Criterion 2.11.7:**

* Marka is blind. She is listening to a video of an hour-long lecture. The section she is in has technical information that builds on material from an earlier section. A sighted user could pause the video and move the slider back to reach visually distinct content from earlier section. Marka uses a control to skip back section by section until she hears the section title name she wants to review. When she is finished, Marka uses the control to move forward section by section until she hears the title of the original topic.
* Wes has fatigue injury that limits the length of his computer sessions. He stops playback of a training video when he is tired. After resting, he restarts the video and navigates to the scene where he left off.

**Related Resources for Success Criterion 2.11.7:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

**2.11.8 Video Contrast and Brightness: 视频对比度和亮度**

Users can adjust the contrast and brightness of visual time-based media. 用户可以在时基媒体可视化中调节对比度和亮度(Level AAA)

[Return to 2.11.8 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2118)

**Intent of Success Criterion 2.11.8:**

Users with certain types of low vision need to control contrast and brightness of time-based media so they can discern the content. Users who are prone to seizures need to be able to reduce or dim contrast and brightness to protect themselves from seizures caused by flashing content.

**Examples for Success Criterion 2.11.8:**

* Frank has albinism. He requires a higher contrast to discern a video image. When Frank is watching an instructional video, he selects a menu item that allows him to increase the contrast of the video. This makes it easier for him to see the important content.
* Kelly has photo-epilepsy and is watching an amateur video taken on a sunny day near the water. Concerned that the video may contain flashing that could trigger a seizure, Kelly selects a menu item of video controls that allow her to reduce the brightness and contrast of the video. While some of the detail is lost, Kelly can safely watch the video.

**Related Resources for Success Criterion 2.11.8:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

Implementing Guideline 2.12 - Support other input devices 支持其它输入设备[[Guideline 2.12](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-other-devices)]

**Summary**: User agents support all of the platform's text input devices (2.12.1, Level A), and for all input devices the user can input text (2.12.3, Level AAA) and perform all other functions (2.12.2, Level AA).

**2.12.1 Platform Text Input Devices: 平台文本输入设备**

If the [platform](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-platform) supports text input using an input device, the user agent is compatible with this functionality. (Level A)

[Return to 2.12.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2121)

**Intent of Success Criterion 2.12.1:**

Some users rely entirely on pointing devices, touch, or gestures, and find these methods more convenient than keyboards. Other users rely on speech input. These users can operate applications more easily and efficiently if they can enter text with their chosen method of input. On systems that support alternative input methods, it is important that the user agent support these utilities.

**Examples for Success Criterion 2.12.1:**

* Ruth has extremely limited motor control and slurred speech. She operates her computer using a head pointer. Her desktop operating system includes a virtual keyboard utility. When active, the on-screen keyboard reserves the bottom portion of the screen for its own use, so the user agent respects this and does not cover that area even in modes that would normally take up the full screen. The user agent avoids communicating with the hardware keyboard through a low-level system API that would miss virtual keyboard input.
* Randall has repetitive stress injuries that make keyboard and mouse input painful. He carries out most computer operations using speech input, including completing web forms across multiple web sites. His user agent supports the APIs used by 3rd party speech input applications. This allows him to use his chosen input method whereever needed.
* Rick is paralysed from the neck down. He uses eyeblinks to control an onscreen keyboard. His user agent supports the APIs used by 3rd party eye blink applications.

**Related Resources for Success Criterion 2.12.1:**

* None

**2.12.2 Operation With Any Device: 任何设备的操作**

If an input device is supported by the [platform](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-platform), all user agent functionality other than text input can be operated using that device. (Level AA)

[Return to 2.12.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2122)

**Intent of Success Criterion 2.12.2:**

Some users rely entirely on pointing devices, touch, or gestures, and find these methods more convenient than keyboards. Other users rely on speech input. These users can operate applications more easily and efficiently if they can navigate and control applications. On systems that support alternative input methods, it is important that the user agent support these utilities.

**Examples for Success Criterion 2.12.2:**

* Ruth has extremely limited motor control and slurred speech, so operates her computer using a head pointer. The mouse pointer moves in response to the orientation of her head, and she clicks, double clicks, or drags using a sip-and-puff switch. It is much easier for her to point and click on a button or menu item than it is for her to simulate keyboard shortcuts using her on-screen keyboard. In fact, she prefers to customize her applications to make most functions available through toolbar buttons or menu items, even those that are by default available only through keyboard shortcuts.
* Randall has repetitive stress injuries. The web browser on his smart phone allows him to perform most operations using speech commands. Unfortunately, a few features are only available through the touchscreen, which Randall finds painful to use. In the next version of the browser, the remaining features are enabled using speech, and Randall finds the product safer and more convenient to use.

**Related Resources for Success Criterion 2.12.2:**

* [Success Criterion 2.12.1 Support Platform Text Input Devices](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_2121)

**2.12.3 Text Input With Any Device: 任何设备的文本输入**

If an input device is supported by the [platform](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-platform), all user agent functionality including text input can be operated using that device. (Level AAA)

[Return to 2.12.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_2123)

**Intent of Success Criterion 2.12.3:**

If the platform does not support text, the user agent provides a mechanism for text input as well as all other input device controls. Some users rely entirely on pointing devices, or find them much more convenient than keyboards. These users can operate applications much more easily and efficiently if they can carry out most operations with the pointing device, and only fall back on a physical or on-screen keyboard as infrequently as possible. If the platform provides the ability to enter arbitrary text using a device (such as large vocabulary speech recognition or an on-screen keyboard utility), the user agent is required to support it per [Success Criterion 2.12.1](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_2121). If the platform does not provide such a feature, the browser is encouraged to provide its own.

**Examples for Success Criterion 2.12.3:**

* Ruth has extremely limited motor control and slurred speech, so operates her computer using a head pointer. The mouse pointer moves in response to the orientation of her head, and she clicks, double clicks, or drags using a sip-and-puff switch. The operating system does not provide an on-screen keyboard, but in order to be maximally accessible, a small on-screen keyboard is available as an add-on for her browser.
* Randall has a web browser on his smart phone that allows him to perform most operations using speech commands. By offloading the speech recognition to an Internet server, it is able to perform large vocabulary speech recognition, so Randall can use his voice to compose email and fill in forms, as well as controlling the browser itself.

**Related Resources for Success Criterion 2.12.3:**

* None

**PRINCIPLE 3: Ensure that the user interface is understandable 确认用户是可理解的**

Implementing Guideline 3.1 - Help users avoid unnecessary messages [[Guideline 3.1](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#gl-avoid-unnecessary-messages)] 帮助用户避开无效的信息

**Summary**: Users can turn off non-essential messages from the author or user-agent (3.1.1, Level AA).

**3.1.1 Reduce Interruptions: 减少中断**

The user can avoid or defer: (Level AA)

1. [Recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) messages that are non-essential or low priority
2. Information in the user agent user interface that is being updated or changing
3. Rendered content that is being updated or changing

[Return to 3.1.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_311)

**Intent of Success Criterion 3.1.1:**

It's important that users be able to avoid unnecessary messages. Messages designed to inform the user can be a burden to users for whom pressing keys is time-consuming, tiring, or painful. Similarly, users with some cognitive impairments find it difficult to deal with constant distractions caused by updating visual or audio information, and would prefer to hide unnecessary updates.

**Examples for Success Criterion 3.1.1:**

* Randall has repetitive strain injuries. He finds pop-up messages difficult. Randall has set the option to receive updates only on request. This way, he can request updates when he has a friend helping him, so that if the update changes the way his computer behaves, he can get help to adjust the computer settings or roll the update back.
* Shirley has a cognitive disability and is distracted by the page flicker. A web page has a stock market ticker that is coded as having a WAI-ARIA low priority level, so she changes the browser's preference setting to indicate that regions with low priority level should not be automatically updated.

**Related Resources for Success Criterion 3.1.1:**

* [WAI-ARIA](http://www.w3.org/WAI/intro/aria)

Implementing Guideline 3.2 - Help users avoid and correct mistakes [[Guideline 3.2](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-avoid-mistakes)] 帮助用户回避和纠正错误

**Summary**: Users can have form submissions require confirmation (3.2.1, Level AA), go back after navigating (3.2.2, Level AA), have their text checked for spelling errors (3.2.3, Level AA), undo text entry (3.2.4, Level A), avoid or undo settings changes (3.2.5, Level A), and receive indications of progress activity (3.2.6, Level A).

**3.2.1 Form Submission Confirm: 表单递交确认**

The user can specify whether or not [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) form submissions must be confirmed. (Level AA)

[Return to 3.2.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_321)

**Intent of Success Criterion 3.2.1:**

Users need to be protected against accidentally submitting a form. Some assistive technologies use the Enter key to advance to the next field. If the form is designed to submit on Enter, the user can unknowingly "submit on Enter" function.

**Examples for Success Criterion 3.2.1:**

* Marka is blind and uses a screenreader. When she installs a web browser she selects an option to disable form submission on Enter. This allows her to complete forms from the banking website knowing that the form won't be submitted until she selects the submit button.
* Ryan uses a mouse but his hand tremor makes him more likely to accidentally miss his intended target. Cancel and Submit (or OK) buttons are often next to each other, and Ryan has found that submitting a form inappropriately can cause serious complications, so he sets a browser option to always request confirmation of form submissions. As a result, when he clicks on a submit button the browser presents a message box asking "Are you sure you want to submit this form?" with larger, well-spaced buttons for Submit and Cancel.

**Related Resources for Success Criterion 3.2.1:**

* [2.1.1 Keyboard Operation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_211)
* [3.2.4 Text Entry Undo](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_324)

**3.2.2 Back Button: 返回按钮**

The user can reverse [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) navigation between web addresses (e.g. standard "back button" functionality). (Level AA)

[Return to 3.2.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_322)

**Intent of Success Criterion 3.2.2:**

Retracing a navigation step is important for users with cognitive issues that involve memory and attention. This is also important for users whose means of input is not 100% accurate, such as speech input users or users with fine motor challenges. It is also beneficial for users for whom navigation is time consuming, tiring, or painful, because it allows them to avoid having to re-enter long URLs.

**Examples for Success Criterion 3.2.2:**

* Joe is a quadraplegic who is using speech input in a relatively noisy room. The program hears an especially loud word from across the room and interprets it as Joe saying "Enter" to click a selected link. Joe says "Go Back" to go back to the page he was on.
* Mike's head injury leaves him easily distracted. He is in the middle of a search when the phone rings. When he comes back he says "Go Back" to go back to the original search page so he can reorient himself.
* Ellen had a head injury that affects her short-term memory. She clicks on a link, and is interrupted by a colleague. When she goes back to her task she has forgotten what she was originally doing. It's important that she be able to retrace her steps to reorient herself.
* Etta has moderately low vision and doesn't use assistive technology. She occasionally clicks on the wrong link. When she does so, she clicks the Back button in order to get back to where she had been.

**Related Resources for Success Criterion 3.2.2:**

* [1.8.10 Provide Viewport History](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_1810)

**3.2.3 Spell Check:  拼写检查**

User agents provide spell checking functionality for text created inside the user agent. (Level AA)

[Return to 3.2.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_323)

**Intent of Success Criterion 3.2.3:**

Users with various disabilities benefit from spell checkers. The ability to check spelling is particularly important for users with disabilities such as dyslexia that significantly increase the likelihood of misspelled words. Spellcheckers also alert blind and low vision users to errors in text entry.

**Examples for Success Criterion 3.2.3:**

* Amanda is dyslexic and frequently spells words incorrectly. She is able to correct words when alerted to the errors. She navigates to her web-based email application and composes a new message. The user agent alerts her to spelling errors as she is typing and she quickly corrects the mistakes and sends an error-free message.

**Related Resources for Success Criterion 3.2.3:**

* None

**3.2.4 Text Entry Undo: 撤销文本输入**

The user can reverse [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) text entry actions prior to submission. (Level A)

* *Note:* Submission can be triggered in many different ways, such as clicking a submit button, typing a key in a control with an onkeypress event, or by a script responding to a timer.

[Return to 3.2.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_324)

**Intent of Success Criterion 3.2.4:**

Users who are blind, have visual impairments, or have cognitive disabilities may have difficulty determining the location of the keyboard focus. This puts them at risk of entering text in an undesired window or location. Users with mobility problems may have difficulty selecting a form field and may not notice an incorrect selection until they have entered text information.  These users need to be able to reverse a text entry ("Undo") prior to submission.

**Examples for Success Criterion 3.2.4:**

* Billie is a paraplegic who uses speech input. When she is working from the main office, she is in a noisy location that can interfere with her speech input.  She is writing a blog entry and is almost finished when her speech input software incorrectly interprets some background noise as a "select all" command. This causes her to overwrite her entire blog entry with the next phrase she utters. She then uses the "undo" command to reverse the text entry and restore her blog entry.
* George is blind and uses a screenreader.  He is entering financial information into his banking billpaying account.  He types the first few letters of the payee from a long list. When George reviews the selection prior to pressing "submit", he hears that he had selected the wrong payee. George uses the "undo" command and selects the correct payee.

**Related Resources for Success Criterion 3.2.4:**

* None

**3.2.5 Settings Changes can be Reversed or Confirmed: 设定改变能够被撤销或确认**

If the user agent provides mechanisms for changing its user interface settings, it either allows the user to reverse the setting changes, or the user can require user confirmation to proceed. (Level A)

[Return to 3.2.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_325)

**Intent of Success Criterion 3.2.5:**

The description of some user interface settings can be confusing to less technical users. Settings changes can also have unintended consequences. In addition, some disabilities make it more likely that a user can make an unintended selection on a preference screen.  Users need to be able to reverse changes to the user interface.

**Examples for Success Criterion 3.2.5:**

* Davy has moderately low vision. He is adjusting the contrast of the background on his mobile phone when he accidentally selects a white background with the previously selected white text. This causes all the icon labels to disappear.  He can see a highlighted rectangle on the screen that usually contains the word "undo" when he makes a change on his phone. He selects that box and the dark background returns, so he can now read the text.  He then changes the background to a color with sufficient contrast for comfortable reading.

**Related Resources for Success Criterion 3.2.5:**

* [Guideline 2.7 - Configure and store preference settings](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#gl-store-prefs)

**3.2.6 Retrieval Progress: 检索进程**

By default, the user agent shows the state of content retrieval activity. (Level A)

[Return to 3.2.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_326)

**Intent of Success Criterion 3.2.6:**

Users need to know that their actions are producing results even if there is a time delay. Users with limited ability to interact with a device need a passive indicator of progress. Users who cannot see visual indications need to have feedback indicating a time delay and have an idea of where they are in the retrieval process. This reduces errors and unnecessary duplicate actions.

**Examples for Success Criterion 3.2.6:**

* Larry has severe repetitive strain injuries and is limited to typing for only a short period of time every day. He downloads a long document, in which he wants to search. He periodically checks the progress bar to make sure it is fully downloaded before running his search.
* Jennifer is blind and uses a screen reader. She clicks on a link that is downloading a large file. The user agent displays a programmatically available progress bar. If the progress stops, the user agent displays a message that it has timed out.
* Jennifer purchases an item from a online shopping cart and is waiting for confirmation. The user agent displays a programmatically available message that it is waiting for a response from the server. If the process times out, the user agent displays a message that the purchase has timed out.

**Related Resources for Success Criterion 3.2.6:**

* [4.1.6 Expose Accessible Properties](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_416)

Implementing Guideline 3.3 - Document the user agent user interface including accessibility features 记录用户代理的用户界面，包括无障碍特性[[Guideline 3.3](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-doc-access-features)]

**Summary**: User documentation is available in an accessible format (3.3.1, Level A), it includes accessibility features (3.3.2, Level A), delineates differences between versions (3.3.3, Level AA), provides a centralized view of conformance UAAG2.0 (3.3.4, Level AAA).

**3.3.1 Accessible Documentation: 可访问记录**

Product [documentation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-documentation) is available in a format that meets success criteria of WCAG 2.0 level "A" or greater.(Level A)

[Return to 3.3.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_331)

**Intent of Success Criterion 3.3.1:**

People with disabilities need documentation in a format that is accessible. If provided as web content, it must at least conform to WCAG 2.0 level "A." If the document is not provided as web content, it must be in conformance to a published accessibility benchmark.

**Examples for Success Criterion 3.3.1:**

* Randall has repetitive strain injuries. His injuries make it painful to sit at the computer for long periods of time. When exploring a new program, he will often print out the documentation to read while away from his desk.This allows him to be more efficient, and learn new programs without having to spend as much computer time.
* Lee has low vision. When she upgrades her user agent, she wants to see if any features she relies on have changed. She opens the HTML help documentation and pleased to see that the documentation is using her preferred settings.

**Related Resources for Success Criterion 3.3.1:**

* None

**3.3.2 Describe Accessibility Features: 无障碍特征介绍**

For each user agent feature that is used to meet UAAG 2.0, at least one of the following is true: (Level A)

1. **Described in the Documentation**: Use of the feature is explained in the user agent's documentation; or
2. **Described in the Interface**: Use of the feature is explained in the user agent user interface; or
3. **Platform Service**: The feature is a service provided by an underlying platform; or
4. **Not Used by Users**: The feature is not used directly by users (e.g., passing information to a platform accessibility service).

[Return to 3.3.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_332)

**Intent of Success Criterion 3.3.2:**

Some users with disabilities will need help in determining how to use the accessibility features that user agents provide. There are four possibilities:

1. This information is provided in the documentation of the user agent (e.g. help system, context sensitive help, etc.);
2. The user interface element is self-explanatory (e.g. a zoom % drop-down menu);
3. The accessibility feature is actually a service of the platform (e.g. high contrast mode), which therefore has the responsibility to document the feature; or
4. The feature is not used directly by users (e.g., passing information to a platform accessibility service). In this case, user documentation is not required, although developer documentation (e.g. how accessibility APIs are used, the user agent's own plug-in API) would still be recommended.

**Examples for Success Criterion 3.3.2:**

* Lee has low vision. When she installs a new user agent, she checks the accessibility documentation to see if the features she needs are available. She finds section entitled "Browser Features Supporting Accessibility", which provides a detailed description of user agent features that provide accessibility, describes how they function, and lists any supported third party assistive technologies that may be supported or required.
* Lee is exploring the menus of her user agent and finds a feature named "Use My Style Sheet". She clicks "help" to learn that this feature allows custom CSS stylesheets to be created to help make web content more accessible.
* Neta has a repetitive strain injury. She relies on gestures and shortcuts to complete tasks. Using a specialized command, she pulls up a list of all the gesture commands available including descriptions.

**Related Resources for Success Criterion 3.3.2:**

* [3.3.4 Centralized View](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_334)

**3.3.3 Changes Between Versions: 版本变更**

Changes to features that meet UAAG 2.0 success criteria since the previous user agent release are documented.(Level AA)

[Return to 3.3.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_333)

**Intent of Success Criterion 3.3.3:**

Users need to be informed about accessibility features implemented in new versions of the user agent.

**Examples for Success Criterion 3.3.3:**

* Martha goes to an app store on her computer and installs an update for her web browser. After it installs, she sees a welcome page describing the new features in this release. One of the welcome page links is titled "What's New For Accessibility". Following this link, Martha reads about the accessibility improvements and discovers a feature that allows her to have tooltips displayed for elements when she is using caret browsing. The text also informs Martha that this feature is off by default and that she should go to accessibility settings to turn it on.

**Related Resources for Success Criterion 3.3.3:**

* None

**3.3.4 Centralized View:  集中视图**

There is a dedicated section of the documentation that presents a view of all features of the user agent necessary to meet the requirements of User Agent Accessibility Guidelines 2.0. (Level AAA)

[Return to 3.3.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_334)

**Intent of Success Criterion 3.3.4:**

Users need to know about accessibility features and how to operate them. A centralized view of accessibility features makes it easier for people with disabilities and people who are evaluating software to quickly become familiar with the features such as keyboard shortcuts, how to zoom the viewport, and where to find accessibility configuration settings. Nested user agents or add-ons may provide separate centralized documentation. It is also useful to document accessibility features in context (such as displaying keyboard shortcuts next to their menu command).

**Examples for Success Criterion 3.3.4:**

* Bob is blind and uses a screen reader that is part of his phone's operating system. He downloads a new web browser on his mobile phone. The browser's online help includes a section on accessibility that point him to pages on non-visual access, such as interaction with screen readers, helpful hints such as an explanation of the screen layout, and a list of supported touch gestures.
* Marissa is assessing new user agents for clients who are blind. The first thing she does in looking at each user agent is to turn to the section in the documentation (local or online) that details accessibility features. This gives her a quick overview of the accessibility features across different products.

**Related Resources for Success Criterion 3.3.4:**

* None

Implementing Guideline 3.4 - Make the user agent behave in predictable ways [[Guideline 3.4](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-predictable-operation)] 确使用户代理以可预期的方式表现

**Summary**: Users can prevent non-requested focus changes (3.4.1, Level A).

**3.4.1 Avoid Unpredictable Focus: 避免无法预期的聚集**

The user can prevent focus changes that are not a result of explicit user request. (Level A)

[Return to 3.4.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_341)

**Intent of Success Criterion 3.4.1:**

Users need to know that navigation in a web page is going to start in a predictable location and move in a predictable fashion. If a page moves the initial focus to somewhere other than the beginning of the page, the user may skip over content without realizing it. If the focus moves and remains unnoticed, users can make unintentional changes, such as entering data in an incorrect field. Focus changes can cause a window to scroll unexpectedly, confusing users. This is particularly problematic for users who can only see a small portion of the document, because they must use more effort to determine where the cursor has moved. Such users also are more likely to continue typing, not immediately realizing that the context has changed. Users who find navigation time consuming, tiring, or painful (including those using screen readers or with impaired dexterity) may also need more steps to return to the area where they want to work. It may improve accessibility for some users on some pages to have the page set focus to specific links or fields when the page loads, but this can be detrimental for other users, and therefore users need to control this behavior.

**Examples for Success Criterion 3.4.1:**

* Jerome has repetitive strain injuries. He tries to limit the number of keyboard and mouse entries he makes each day, because it is painful. He loads a page that has its default focus in a search box. Jerome wants to read the content of the page, rather than search. This requires him to make additional clicks to get to the content of the page. He intends to use this page frequently, so he adjusts his browser's settings to disable the automatic focus change to prevent the extra clicking in the future.
* Jessica uses a screen enlarger. She loads a page that contains instructions followed by a form. If the page automatically moves the keyboard focus to the form, she may not realize there were instructions. To avoid this problem, she sets an option to prevent default focus changes.
* James uses speech recognition. He speaks his credit card number by saying several digits followed by speaking "tab". He needs to know ahead of time whether it is necessary to include the "tab" command in the phrase.
* Joey is blind and uses a screenreader. She is filling in a web form that asks for her phone number using three separate fields. When she types the three digits of her area code into the first field, the browser automatically moves the focus to the second field. Because Joey does not realize that the focus has moved for her, she presses the Tab key to move it manually, not realizing that this now puts the focus on the third field rather than the second. To avoid this problem, Joey goes to her browser's Preferences dialog box and checks the option that prevents focus changes that she has not explicitly requested.
* Justine has trouble using a mouse. She uses a keyboard macro in her web email program to execute a command to start a new message and put a block of text in the new message. The focus moves without her control, causing the macro to execute in the Inbox instead of the Compose window. With the Inbox in focus several single-key shortcuts execute, changing the column sorting and deleting several messages.

**Related Resources for Success Criterion 3.4.1:**

* [1.3.1 Highlighted Items](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_131)
* [1.8.11 Allow Top-Level Viewport Open on Request](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_1811)
* [1.8.12 Allow Top-Level Viewport Focus Control](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_1812)

**PRINCIPLE 4: Facilitate programmatic access 实现程序化访问**

Implementing Guideline 4.1 - Facilitate programmatic access to assistive technology [[Guideline 4.1](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-AT-access)] 为辅助技术实现程序化访问

**Summary**: The user agent supports platform accessibility services (4.1.1, Level A) that are quick and responsive (4.1.7, Level A), including providing information about all controls and operation (4.1.2, Level A & 4.1.6, Level AA), access to DOMs (4.1.4, Level A). Controls can be adjusted programmatically (4.1.5, Level A). Where something can't be made accessible, provide an accessible alternative version, such as a standard window in place of a customized window (4.1.3, Level A).

**Note**: UAAG 2.0 assumes that a platform accessibility API will be built on top of underlying security architectures that will allow user agents to comply with both the success criteria and security needs.

**4.1.1 Support Platform Accessibility Services: 支持平台上的无障碍服务**

The user agent supports relevant [platform accessibility services](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch). 用户代理支持相应的平台上的无障碍服务 (Level A)

[Return to 4.1.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_411)

**Intent of Success Criterion 4.1.1:**

Platform accessibility services provide common functionality across the well-behaved applications running on the platform. This reduces exceptions that assistive technology has to implement for the hundreds of applications it may need to support. Most major operating environments provide platform accessibility services that allow applications to work with assistive technologies. These platform accessibility services must be supported by both the user agent and the assistive technology. Specifics of what constitutes a platform accessibility service will differ on each platform, but basic features common to these services are addressed by other success criteria under Guideline 4.1.

Most web-based user agents support this requirement automatically because they run inside host user agents. The host is responsible for exposing all content, including nested user agents, via platform accessibility services. As long as the nested user agent's user interface is entirely web-based and complies with Web Content Accessibility Guidelines (e.g. providing alternative text and supporting WAI-ARIA where needed) the host will understand it well enough to provide a bridge between the content and platform accessibility services.

**Examples for Success Criterion 4.1.1:**

* Jamie is blind and uses a screenreader. He is testing a new type of button bar for a browser. When coding the new component, the developer supported the relevant platform accessibility services so that Jamie's screenreader can recognize it as a toolbar, and Jamie can identify, navigate, and activate the bar and its buttons.

**Related Resources for Success Criterion 4.1.1:**

* [4.1.2 Name, Role, State, Value, Description](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_412) and [4.1.6 Expose Accessible Properties](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#sc_416) for specific information that should be exposed through platform accessibility services.
* [Microsoft Active Accessibility (MSAA)](http://msdn.microsoft.com/en-us/library/ms971310.aspx)
* [Microsoft UI Automation](http://msdn.microsoft.com/en-us/library/ms747327.aspx)
* [Gnome Accessibility Toolkit (ATK)](http://library.gnome.org/devel/atk/)
* [GNOME Accessibility Architecture (ATK and AT-SPI)](http://accessibility.kde.org/developer/atk.php)
* [Accessibility Programming Guidelines for Mac](https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/Accessibility/)
* [Accessibility Programming Guide for iOS](https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility/)
* [Iaccessible2](http://www.linuxfoundation.org/collaborate/workgroups/accessibility/iaccessible2)
* [Open Source Accessibility for Windows](http://www-03.ibm.com/able/open_computing/open_source_windows.html)
* [Accessibility API Cross reference](http://www.mozilla.org/access/platform-apis.html)

**4.1.2 Expose Basic Properties: 公开基本属性**

For all user interface components, including [user agent user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui), [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content), and generated content, the user agent makes available the following via a [platform accessibility service](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch): 对于所有用户界面组件，包括用户代理用户界面，呈现内容、以及生成内容，用户代理应该可以通过平台无障碍服务使如下部分是可用的 (Level A)

* Name 名称
* Role 角色
* State 状态
* Value 值
* Selection 选择
* Focus 焦点

[Return to 4.1.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_412)

**Intent of Success Criterion 4.1.2:**

Modern web content is highly interactive. Including people who use assistive technology in the interactive experience requires that the user agent provide information about user interface components in a standardized manner.This information includes:

* Name (component name) 组件名
* Role (purpose, such as alert, button, checkbox) 目标，比如说警告、按钮、检查框
* State (current status, such as busy, disabled, hidden) 当前状态，比如状态忙、不可用、隐藏
* Value (information associated with the component such as, the data in a text box, the position number of a slider, the date in a calendar widget)
* Focus (has focus, focusable)
* Selection (selected, selectable)

Every component developed for the user agent must pass this information to the appropriate accessibility platform architecture or application program interface (API). Embedded user agents, like media players can pass Name, Role, State, and Value via the WAI-ARIA techniques. Since not every element supports every property, follow the platform accessibility services convention for cases where an element type does not support one of the listed properties.

**Examples for Success Criterion 4.1.2:**

* Jamie is blind and uses a screenreader. He adjusts the volume on a web site media player using a slider control. This is easy for him to do because the developer has coded the slider component with WAI-ARIA techniques to pass the following information to the accessibility API:  
  Name: Volume control  
  Role: Slider  
  States & Values:
  + aria-valuenow is the slider’s current value.
  + aria-value-min is the minimum of the volume range
  + aria-value-max is the maximum of the volume range

**Related Resources for Success Criterion 4.1.2:**

* http://msdn.microsoft.com/en-us/library/ms697187
* http://www.mozilla.org/access/windows/msaa-server

**4.1.3 Provide Equivalent Accessible Alternatives: 提供等效的可用替换**

If a component of the user agent user interface cannot be exposed through [platform accessibility services](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch), then the user agent provides an equivalent alternative that is exposed through the platform accessibility service.(Level A)

[Return to 4.1.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_413)

**Intent of Success Criterion 4.1.3:**

Like everyone else, users who rely on assistive technology need to be able to carry out all tasks provided by the user agent. When a particular user interface component cannot support the platform accessibility service, and thus can't be made compatible with assistive technology, the user agent should let the user achieve the same goal using another component that is fully accessible.

**Examples for Success Criterion 4.1.3:**

* Doug uses a mouse-stick on his smart phone. He uses the assistive touch option on his mobile phone to control an app for 3D design drawing. The app provides a single, complex control for 3-dimensional manipulation of a virtual object. This custom control cannot be represented in the platform accessibility service, so the app provides Doug the option to achieve the same functionality through an alternate user interface: a panel that adjusts the yar, spin, and roll independently using arrow keys.

**Related Resources for Success Criterion 4.1.3:**

* None

**4.1.4 Make DOMs Programmatically Available: DOMs可被程序化激活**

If the user agent implements one or more [Document Object Models](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-dom) (DOM), they must be made [programmatically available](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-programmatically-avail) to assistive technologies. 如果用户代理实现一个以上的DOM（文档对象模型），它们应该可以被辅助技术程序化激活。 (Level A)

[Return to 4.1.4 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_414)

**Intent of Success Criterion 4.1.4:**

Assistive technologies need all possible information. Applications such as user agents and assistive technologies use a combination of DOMs, accessibility Application Programming Interfaces ([API](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-api)), native platform APIs, and hard-coded heuristics to provide an accessible user interface and accessible content. It is the user agent's responsibility to expose the DOM to assistive technology which in many cases is the richest source of information on web content.

**Examples for Success Criterion 4.1.4:**

* Joey is blind and uses a screenreader. He is reading a long technical article where code samples are labeled using CSS. In the past, he has had trouble reading technical articles because the CSS text is not exposed to the the platform accessibility API, and is therefore missing from his text. His new browser exposes the CSS DOM information to the platform accessibility services. This allows him to read and search on all of the code sample labels.
* Joey is reading a compound document containing HTML, MathML, and SVG. Each has a separate DOM. As Joey moves through the document, he is moving through multiple DOMs. He has no trouble reading it because the transition between DOMs is seamless and transparent.

**Related Resources for Success Criterion 4.1.4:**

* www.w3.org/TR/SVG/svgdom.html
* www.w3.org/TR/MathML/chapter8.html
* www.w3.org/TR/DOM-Level-2-HTML/
* www.w3.org/TR/DOM-Level-2-Style/
* [Gecko DOM Reference](https://developer.mozilla.org/en-US/docs/Gecko_DOM_Reference)
* <http://developer.apple.com/mac/library/documentation/AppleApplications/Conceptual/SafariJSProgTopics/Tasks/DOM.html>
* <http://msdn.microsoft.com/en-us/library/ms533050%28VS.85%29.aspx>
* www.adobe.com/devnet/acrobat/pdfs/access.pdf
* www.w3.org/2004/CDF/
* dev.w3.org/2006/cdf/cdi-framework/
* www.w3.org/TR/CDR/
* <http://accessibility.linuxfoundation.org/a11yspecs/atspi/adoc/a11y-dom-apis.html>

**4.1.5 Make Write Access Programmatically Available: 确保写访问可被程序化激活**

If the user can modify the state or value of a piece of content through the user interface (e.g. by checking a box or editing a text area), the same degree of write access is [programmatically available](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-programmatically-avail). 如果用户可以修改通过用户界面修改某一部分内容的状态或者数值时（比如通过在单选框中选择或者在文本区域做修改），那么相同程序的写访问应该可被程序化激活(Level A)

[Return to 4.1.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_415)

**Intent of Success Criterion 4.1.5:**

If users can control the user interface using any form of input, they can control it through programmatic access. It is often more reliable for assistive technology to use the programmatic method of access versus attempting to simulate mouse or keyboard input.

**Examples for Success Criterion 4.1.5:**

* Billie is a quadrapalagic who uses speech input. When she says "Volume 35" her speech input utility programmatically sets the value of the volume slider to 35%. This allows Billie to set the volume in a single command to a specific value rather than simulating mouse clicks.
* Billie uses a direct speech command to set the value of a check box in the user agent preference settings to block pop up windows. Even though the control cycles through “checked”, and “unchecked” using the spacebar or mouse, the speech command uses a programatic command to check the box regardless of whether the box is already checked.

**Related Resources for Success Criterion 4.1.5:**

* None

**4.1.6 Expose Additional Properties: 公开附加的属性**

For all user interface components, including the [user agent user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui), rendered content, and generated content, the user agent makes [available](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-programmatically-avail) the following, via a [platform accessibility service](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch), if the properties are supported by the service:(Level AA)

* Bounding dimensions and coordinates
* Font family of text
* Font size of text
* Foreground and background color for text
* Change state/value notifications
* Highlighting
* [Keyboard commands](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-command)

[Return to 4.1.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_416)

**Intent of Success Criterion 4.1.6:**

These properties are used by assistive technology to create alternative views of the user agent user interface and rendered content. They also provide alternative means for the user to interact with these items. These properties are exposed in the user agent user interface (e.g. menus and dialog boxes) and to recognized aspects of the user content (e.g. HTML script using ARIA to indicate focus on custom controls). Keyboard commands include direct keyboard commands (e.g. Control-f to select the find box) and keyboard sequences (e.g. Alternate-f, to call up the File menu and select Save As).

**Examples for Success Criterion 4.1.6:**

* Kiara is blind and uses a screenreader. She loads a new version of a popular web browser for the first time. She puts her screen reader into "explore mode", which lets her review what appears on the screen. Using this mode, she identifies the menu bar because her screen reader uses the bounding rectangle of each element to identify that those items all appear on the same horizontal line below the window's title bar.
* Kiara is using a screen reader at her job at a telephone call center. She needs to know the caller name and banking status to immediately respond to each customer appropriately. The web application displays caller names in different colors to indicate banking status, instead of indicating the status semantically (which would be more accessible). However, Kiara's browser exposes the font color to her screenreader, which reads her the customer name followed by the banking status color.
* Max uses a screen magnifier that shows him only a small amount of screen. Because his browser exposes accessible properties to his screen magnifier, he can pan through different portions of a web page and quickly pan back to positions of interest, such as the text matched by a recent Search operation, text that he previously selected by dragging the mouse, or the text caret rather than having to manually pan through the document searching for them.

**Related Resources for Success Criterion 4.1.6:**

* None

**4.1.7 Make Programmatic Exchanges Timely: 确使程序化及时交换**

For APIs implemented to satisfy the requirements of UAAG 2.0, ensure that programmatic exchanges proceed at a rate such that users do not perceive a delay. (Level A)

[Return to 4.1.7 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_417)

**Intent of Success Criterion 4.1.7:**

Conveying information for accessibility may involve communication between a user agent, an accessibility API, document object model, assistive technology and end user interaction. The goal of this success criterion is that when interacting with a user agent, the end user perceives the same amount of delay as someone who is not using assistive technology.

**Examples for Success Criterion 4.1.7:**

* Bonita has polio. She accesses her web browser with a speech input program. She navigates to a web page and speaks the name of a link she wants to click. The link is activated with the same speed as it would be if a mouse had been used to click the link.
* Arthur is blind. He is browsing a web page with a screen reader. As he tabs from link to link, the text of each link promptly appears on his braille display.

**Related Resources for Success Criterion 4.1.7:**

* None

**PRINCIPLE 5: Comply with applicable specifications and conventions 符合适用的规范和惯例**

Implementing Guideline 5.1 - Comply with applicable specifications and conventions [[Guideline 5.1](http://www.w3.org/TR/2013/WD-UAAG20-20131107/Overview.html#gl-obs-env-conventions)] 遵守适用规范和惯例

**Summary**: When the browser's controls are authored in HTML or similar standards, they need to meet W3C's Web Content Accessibility Guidelines (5.1.1, Levels A, AA, AAA). The user agent supports the accessibility features of content formats (5.1.2, Level A) and of the platform (5.1.3, Level A), allows handling of unrendered technologies (5.1.4, Level A) including by alternative viewers (5.1.5, Level AA), and allows users to report accessibility issues (5.1.6, Level AAA).

**5.1.1 Comply with WCAG: 遵守WCAG**

Web-based user agent user interfaces meet the WCAG 2.0 success criteria. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; and Level AAA to meet WCAG 2.0 Level A, AA, and AAA success criteria)

* *Note:* This success criterion does not apply to non-web-based user agent user interfaces, but does include any parts of non-web-based user agents that are web-based (e.g. help systems).

[Return to 5.1.1 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_511)

**Intent of Success Criterion 5.1.1:**

User agent user interfaces that are web based must meet the same accessibility guidelines as web content. The Web Content Accessibility Guidelines (WCAG) 2.0 are an internationally accepted guideline for accessible web content.

**Examples for Success Criterion 5.1.1:**

* Lee has low vision. She needs to customize the appearance of her web based media player. WCAG 2.0 success criteria 2.7.1 requires that a user agent enable a user to change settings that impact accessibility. Her web-based media player enables her to change accessibility settings to display high contrast captions.

**Related Resources for Success Criterion 5.1.1:**

* WAI-ARIA 1.0 User Agent Implementation Guide
* W3C Web Design and Applications Activity
* WCAG 2.0

**5.1.2 Implement Accessibility Features of Content Specifications: 实现内容规范的无障碍特性**

Implement the accessibility features of content specifications. Accessibility features are those that are either 实现内容规范的无障碍特性，如下无障碍特性中的某一种(Level A):

* Identified as such in the content specifications or
* Allow authors to satisfy a requirement of WCAG 2.0
* *Note 1*: If a conformance claim is filed, cite the implemented specifications in the conformance claim.
* *Note 2*: When a rendering requirement of another specification contradicts a requirement of UAAG 2.0, the user agent may disregard the rendering requirement of the other specification and still satisfy this guideline.

[Return to 5.1.2 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_512)

**Intent of Success Criterion 5.1.2:**

Most content specifications include features important to users with disabilities. Users may find it difficult or impossible to use a product that fails to support those features. Most content specifications include features important to users with disabilities. Users may find it difficult or impossible to use a product that fails to support those features. Conformance claim documents list the technologies that the user agent fully supports, such as WAI-ARIA, so that users can make informed decisions about whether or not they will be able to use, and therefore should install, a new product or version of that product.

**Examples for Success Criterion 5.1.2:**

* Jordy is blind and uses a screenreader. He often uses a website that is coded with WAI-ARIA to identify the functions of custom controls. He is downloading a new version of a web browser. If the browser doesn't support WAI-ARIA and expose that information to assistive technology, the website would be unusable. Before he installs the new browser, Jordy makes sure it fully supports WAI-ARIA. He determines this by reading product documentation and UAAG conformance claims posted on the web.

**Related Resources for Success Criterion 5.1.2:**

* [[WCAG20]](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-WCAG20)
* [HTML](http://www.w3.org/TR/#tr_HTML)- - listing of the HTML documents
* [CSS](http://www.w3.org/TR/#tr_CSS) - listing of the CSS documents
* [WAI-ARIA Suite Overview](http://www.w3.org/WAI/intro/aria.php)

**5.1.3 Implement Accessibility Features of the Platform: 实现平台的无障碍特性**

If the user agent contains non-web-based user interfaces, then those user interfaces follow user interface accessibility guidelines for the [platform](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-platform). (Level A)

* *Note:* When a requirement of another specification contradicts a requirement of UAAG 2.0, the user agent may disregard the rendering requirement of the other specification and still satisfy this guideline.

[Return to 5.1.3 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_513)

**Intent of Success Criterion 5.1.3:**

User agent user interfaces that are not web applications need to be accessible to people with disabilities. Accessibility guidelines already exist for many platforms. Most operating systems have conventions and expectations that aid accessibility, such as keyboard behavior, support of an accessibility API, user interface design. User agents need to comply with the basic accessibility requirements of the platform in use. Developers have the flexibility to conform with the appropriate accessibility guidelines or legislation for their platform or markets.

The user should be able to easily discover detailed information about the user agent's adherence to accessibility standards, platform standards such as MSAA or JAA, and third-party standards such as ISO 9241-171, and should be able to do so without installing and testing the accessibility features.

*Note:* In the [conformance claim](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#conformance), list the requirements you fully comply with, list the requirements you partially comply with and explain, and list the requirements you do not comply with and explain. Where applicable, these explanations can be general and cover several sections at once.

**Examples for Success Criterion 5.1.3:**

* Lee has low vision and uses a screen magnifier on a Linux system. She loads a new Gnome application that works with her screen magnifier because the developer followed the [Gnome Accessibility Developers Guide](http://developer.gnome.org/accessibility-devel-guide/). For example, the Keyboard Focus section states: "Show current input focus clearly at all times. Remember that in controls that include a scrolling element, it is not always sufficient to highlight just the selected element inside that scrolling area, as it may not be visible. " Lee can use this app because it conforms to this accessibility guideline for focus.
* Martin uses a mouth stick to control his mobile browser. Even though he cannot use a pinch gesture, he controls the zoom in his mobile browser with a custom gesture. He can do this because the app developer followed the guidance provided in the "Accessibility Programming Guide for iOS".

**Related Resources for Success Criterion 5.1.3:**

**[APPLE-ACCESS]**

"[Introduction to Accessibility Overview](http://developer.apple.com/mac/library/documentation/Accessibility/Conceptual/AccessibilityMacOSX/index.html)," Apple Computer Inc.

**[CARBON-ACCESS]**

"[Introduction to Accessibility Programming Guidelines for Carbon](http://developer.apple.com/legacy/mac/library/documentation/Carbon/Conceptual/MakingAppsAccessible/index.html)," Apple Corporation.

**[COCOA-ACCESS]**

"[Introduction to Accessibility Programming Guidelines for Cocoa](http://developer.apple.com/mac/library/documentation/Cocoa/Conceptual/Accessibility/index.html)," Apple Corporation.

**[EITAAC]**

"[EITAAC Desktop Software standards](http://trace.wisc.edu/docs/eitaac_desktop_software_standards/desktop_software_standards.htm)," Electronic Information Technology Access Advisory (EITAAC) Committee.

**[GNOME-ACCESS]**

"[GNOME Accessibility for Developers](http://developer.gnome.org/accessibility-devel-guide/)," C. Benson, B. Cameron, B. Haneman, S. Snider, P. O'Briain, The GNOME Accessibility Project.

**[GNOME-API]**

"[Gnome Accessibility Toolkit API](http://developer.gnome.org/atk/)"

**[GNOME-KDE-KEYS]**

"[Gnome/KDE Keyboard Shortcuts](http://www.novell.com/coolsolutions/tip/2289.html)," Novell Corporation.

**[IBM-ACCESS]**

"[Software Accessibility](http://www-03.ibm.com/able/guidelines/accesssoftware.html)," IBM Special Needs Systems.

**[IEC-4WD]**

IEC/4WD 61966-2-1: Colour Measurement and Management in Multimedia Systems and Equipment - Part 2.1: Default Colour Space - sRGB. May 5, 1998.

**[ISO-TS-16071]**

"[Ergonomics of human-system interaction -- Guidance on accessibility for human-computer interfaces](http://www.iso.org/iso/catalogue_detail.htm?csnumber=30858)". International Organization for Standardization.

**[JAVA-ACCESS]**

"[IBM Guidelines for Writing Accessible Applications Using 100% Pure Java](http://www.ibm.com/developerworks/library/j-access/)," R. Schwerdtfeger, IBM Special Needs Systems.

**[JAVA-API]**

"[Java Accessibility Package](http://download.oracle.com/javase/1.5.0/docs/api/javax/accessibility/package-summary.html)"

**[JAVA-CHECKLIST]**

"[Java Accessibility Guidelines and Checklist](http://www-306.ibm.com/able/guidelines/java/accessjava.html)," IBM Special Needs Systems.

**[MACOSX-KEYS]**

"[Mac OS X keyboard shortcuts](http://support.apple.com/kb/HT1343)," Apple Corporation.

**[MS-ENABLE]**

"[Designing Accessible Applications](http://msdn.microsoft.com/en-us/library/aa291864%28v=vs.71%29.aspx)," Microsoft Corporation.

**[MS-WIN7-ACCESS]**

["Engineering Software For Accessibility"](http://www.microsoft.com/downloads/details.aspx?familyid=6E71A995-45E0-4170-8AC2-C1C0E635548D&displaylang=en), Microsoft Corporation.

**[MS-KEYS]**

"[Keyboard shortcuts for Windows](http://support.microsoft.com/default.aspx?scid=kb;en-us;q126449)," Microsoft Corporation.

**[NOTES-ACCESS]**

"[Lotus Notes application accessibility](http://www-03.ibm.com/able/guidelines/notes/accessr5.html)," IBM Corporation.

**[ORACLE-DESIGN]**

"[Designing for Accessibility](http://www.oracle.com/us/sun/index.htm)," Eric Bergman and Earl Johnson. This paper discusses specific disabilities including those related to hearing, vision, and cognitive function.

**[Editors' Note: Resource links from Jim - compare and expand]**

http://www.microsoft.com/windowsxp/using/accessibility/default.mspx

http://www.apple.com/accessibility/

http://www.linux.org/docs/ldp/howto/Accessibility-HOWTO/linuxos.html

http://www.linuxfoundation.org/collaborate/workgroups/accessibility/iaccessible2

http://developer.apple.com/ue/accessibility/

http://msdn.microsoft.com/en-us/library/dd373592%28VS.85%29.aspx

http://msdn.microsoft.com/en-us/windows/ee815673.aspx

**5.1.5 Allow Content Elements to be Rendered in Alternative Viewers: 允许内容元素在替代视图中被呈现**

The user can select content elements and have them rendered in alternative viewers. (Level AA)

[Return to 5.1.5 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_515)

**Intent of Success Criterion 5.1.5:**

When accessing media or specialized content (e.g. MathML) on the web, users with disabilities may have a richer or more accessible experience using a third-party application, plug-in, or extension, rather than using the browser's built-in facilities. Users need to be able to enable or activate a browser plug-in or extension to interact with the content. Alternately, they may elect to save that content to disk and launch it in a third-party application. If streaming video cannot be saved to disk, the browser launches the external viewer, passing it the URI to the online video.

**Examples for Success Criterion 5.1.5:**

* George is blind. His browser supports the VIDEO tag and adds its own play and pause controls, but George prefers to view the video content in a third-party application that provides much more sophisticated navigation controls such as bookmarks, skip-forward and backwards, and the ability to speed playback without increasing pitch of the audio track. In the browser, he right-clicks on the video to display a context menu, and from that chooses "Open in…", and then chooses his preferred video player. The browser launches the player to show that video file.
* Jukka is blind and uses a screenreader. He is a scientist whose uses mathematical models. Many of the journals he reads online use MathML to display equations. Jukka finds the native support for MathML accessibility in his web browser to be generally compatible with his screen reader, but it can become unreliable for extremely complex equations. In those cases, Jukka selects an alternate rendering plugin to make the MathML understandable to his screen reader.

**Related Resources for Success Criterion 5.1.5:**

* None

**5.1.6 Enable Reporting of User Agent Accessibility Faults: 启用用户代理的无障碍故障报告**

The user agent provides a mechanism for users to report user agent accessibility issues. (Level AAA)

[Return to 5.1.6 in Guidelines](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#sc_516)

**Intent of Success Criterion 5.1.6:**

People who use assistive technologies such as screen readers may find that a technology isn't fully compatible with a web browser, or that the content authored to meet WCAG still is not accessible when rendered in the browser. This causes information loss and inconvenience. When this happens, users benefit from being able to easily file a report with the user agent vendor to report the incompatibility, similar to the way users can file bug reports or provide feedback.

**Examples for Success Criterion 5.1.6:**

* Alice is a visually impaired college student who frequently uses a refreshable braille display with her web browser. She experiences difficulty with a drop-down list boxes because the text is only partially rendered on her braille display. When Alice notices this incompatibility, she navigates to the feedback section of her browser. After providing some basic information (such as AT software and computer hardware used), Alice describes the problem and submits the report to the browser vendor.
* Fred has low vision and uses screen magnification software to enlarge displayed information and alter color contrast. While reading content on a website in compliance with WCAG, Fred discovers that the scroll bars are not visible and he cannot scroll further down the page. Unable to read the information he needs, Fred selects the Feedback option from his browser's Tools menu, and is presented with a dialog where he can select which information to transmit to the browser vendor. He includes the current URI of the page he is trying to view, system information including OS and screen magnifier version, and enters a description of the problem he is having. Though not required, Fred chooses to provide his email address so that the vendor may contact him for further details or to provide a workaround.

**Related Resources for Success Criterion 5.1.6:**

* None

Conformance

This section is normative.

Conformance means that the user agent satisfies the success criteria defined in the guidelines section. This section lists requirements for conformance and conformance claims.

Conformance Requirements

In order for a web page to conform to UAAG 2.0, one of the following levels of conformance is met in full.

* Level A: For level A conformance (the minimum level of conformance), the user agent satisfies all the Level A success criteria.
* Level AA: For level AA conformance, the user agent satisfies all level A and level AA Success Criteria.
* Level AAA: For level AAA conformance, the user agent satisfies all level A, level AA and level AAA Success Criteria.

Note: Although conformance can only be achieved at the stated levels, developers are encouraged to report (in their claim) any progress toward meeting success criteria from all levels beyond the achieved level of conformance.

Conformance Claims

User agents can conform to UAAG 2.0 without making a claim. If a conformance claim is made, the conformance claim must meet the following conditions and include the following information:

**Conditions on Conformance Claims**

If a conformance claim is made, the conformance claim must meet the following conditions:

* At least one version of the conformance claim must be published on the web as a document meeting level "A" of WCAG 2.0. A suggested metadata description for this document is "UAAG 2.0 Conformance Claim".
* Whenever the claimed conformance level is published (e.g. product information website), the URI for the on-line published version of the conformance claim must be included.
* The existence of a conformance claim does not imply that the W3C has reviewed the claim or assured its validity.
* There are no restrictions on who can make a claim.
* Claimants are solely responsible for the accuracy of their claims.
* Claimants are encouraged to claim conformance to the most recent version of the User Agent Accessibility Guidelines Recommendation.

**Components of UAAG 2.0 Conformance Claims**

1. Claimant name and affiliation
2. Claimant contact information
3. Date of the claim
4. Conformance level satisfied
5. **User agent information**:
   1. Name and manufacturer
   2. Version number or version range
   3. Required patches or updates, human language of the user interface and documentation
   4. Configuration changes to the user agent that are needed to meet the success criteria (e.g. ignore author foreground/background color, turn on Carat Browsing)
   5. Plugins or extensions (including version numbers) needed to meet the success criteria (e.g. mouseless browsing)
6. **Platform:** Provide relevant information about the software and/or hardware platform(s) that the user agent relies on for conformance. This information may include:
   1. Name and manufacturer
   2. Version of key software components (e.g. operating system, other software environment)
   3. Hardware requirements (e.g. audio output enabled, minimum screen size: 2", bluetooth keyboard attached)
   4. Operating system(s) (e.g. Windows, Android, iOS, GNOME)
   5. Other software environment (Java, Eclipse)
   6. Host web browser when the conforming user agent is web-based (e.g. JW Player on Firefox)
   7. Configuration changes to the platform that are needed to meet the success criteria (e.g. turn on Sticky Keys, use High Contrast Mode)
7. **Platform Limitations:** If the platform (hardware or operating system) does not support a capability necessary for a given UAAG 2.0 success criterion, list the success criterion and the feature (e.g. a mobile operating system does not support platform accessibility services, therefore the user agent cannot meet success criterion 4.1.2). For these listed technologies, the user agent can claim that the success criteria do not apply.
8. **Web Content Technologies:** List the web content technologies rendered by the user agent that are included in the claim. If there are any web content technologies rendered by the user agent that are excluded from the conformance claim, list these separately. Examples of web content technologies include web markup languages such HTML, XML, CSS, SVG, and MathML, image formats such as PNG, JPG and GIF, scripting languages such as JavaScript/EcmaScript, specific video codecs, and proprietary document formats.
9. **Declarations:** For each success criterion, provide a declaration of either
   1. whether or not the success criterion has been satisfied; or
   2. declaration that the success criterion is not applicable and a rationale for why not

**Limited Conformance for Extensions**

This option may be used for a user agent extension or plug-in with limited functionality that wishes to claim UAAG 2.0 conformance. An extension or plugin can claim conformance for a specific success criterion or a narrow range of success criteria as stated in the claim. All other success criteria may be denoted as Not Applicable. The add-in must not cause the combined user agent (hosting user agent plus installed extension or plug-in) to fail any success criteria that the hosting user agent would otherwise pass.

**Optional Components of an UAAG 2.0 Conformance Claim**

A description of how the UAAG 2.0 success criteria were met where this may not be obvious.

Disclaimer

Neither W3C, WAI, nor UAWG take any responsibility for any aspect or result of any UAAG 2.0 conformance claim that has not been published under the authority of the W3C, WAI, or UAWG.

Appendix A: Glossary

This glossary is [normative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-normative).

[a](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#a) · b · [c](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#c) · [d](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#d) · [e](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#e) · [f](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#f) · [g](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#g) · [h](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#h) · [i](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#i) · j · k · l · m · [n](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#n) · [o](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#o) · [p](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#p) · q · [r](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#r) · [s](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#s) · [t](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#t) · [u](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#u) · [v](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#v) · [w](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#w) · x · y · z

***activate***

To carry out the behaviors associated with an [enabled element](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-enabled-element) in the [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content) or a component of the [user agent user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui).

***alternative content***

[Web content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) that user agents can programmatically determine is usable in place of other content that some people are not able to access. Alternative content fulfills essentially the same function or purpose as the original content. There are several general types of alternative content:

* ***text alternative for non-text content***: Text that is programmatically associated with [non-text content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-non-text-content) or referred to from text that is programmatically associated with non-text content. For example, an image of a chart might have two text alternatives: a short text alternative and a described-by relationship to a nearby paragraph that more fully describes the content of the chart.
* ***alternative for time-based media***: [Web content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) that serves the same function or purpose as one or more tracks in a time-based media presentation. This includes alternatives for audio such as [captions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-captions) and sign language interpretation, and alternatives for video such as[audio descriptions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-description) and [extended audio descriptions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-Auditory-Description-Extended). Another form of time-based media alternative is a correctly sequenced text description of time-based visual and auditory information that also is capable of achieving the outcomes of any interactivity in the time-based presentation.
* ***text alternatives for text content***: Text that is programmatically associated with certain types of text content to provide expanded information. For example, an abbreviation (or acronym) may provide an expansion of the shortened word or initialized words (<abbr title="User Agent Accessibility Guidelines">UAAG</abbr>).
* ***media alternative for text***: Media that presents no more information than is already presented in text (directly or via text alternatives). A media alternative for text is provided for people who benefit from alternate representations of text. Media alternatives for text may be audio-only, video-only (including sign-language video), or audio-video.

**Note**: According to WCAG 2.0, alternative content may or may not be programmatically determinable (e.g., a short description for an image might appear in the image's description attribute or within text near the image). However, UAAG 2.0 adds the [programmatically available](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-programmatically-avail)condition because this is the only type of alternative content that user agents can recognize.

***animation***

Graphical [content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) rendered to automatically change over time, giving the user a visual perception of movement. Examples include [video](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-video), animated images, scrolling text, programmatic animation (e.g. moving or replacing rendered objects).

***application programming interface (API)***

A mechanism that defines how communication may take place between applications.

***assistive technology***

For the purpose of UAAG 2.0 conformance, assistive technology meets the following criteria:

1. Relies on services (such as retrieving [web resources](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-web-resource) and parsing markup) provided by one or more host user agents.
2. Communicates data and messages with host user agents by monitoring and using [APIs](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-api).
3. Provides services beyond those offered by the host user agents to meet the requirements of users with disabilities. Additional services include alternative renderings (e.g. as synthesized speech or magnified content), alternative input methods (e.g. voice), additional navigation or orientation mechanisms, and content transformations (e.g. to make tables more accessible).

Examples of assistive technologies that are important in the context of UAAG 2.0 include the following:

* Screen magnifiers, which are used by people with visual disabilities to enlarge and change colors on the screen to improve the visual readability of rendered text and images.
* Screen readers, which are used by people who are blind or have reading disabilities to read textual information through synthesized speech or braille displays.
* Voice recognition software, which is used by some people who have physical disabilities to simulate the keyboard and mouse.
* Alternative keyboards, which are used by some people with physical disabilities to simulate the keyboard and mouse.
* Alternative pointing devices, which are used by some people with physical disabilities to simulate mouse pointing and button activations.

***audio***

The technology of sound transmission. Audio can be created synthetically (including speech synthesis), streamed from a live source (e.g. a radio broadcast), or recorded from real world sounds. There may be multiple [audio tracks](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-track) in a presentation.

***audio description***

A type of [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) that takes the form of narration added to the [audio](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio) to describe important visual details that cannot be understood from the main soundtrack alone. Audio description of [video](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-video) provides information about actions, characters, scene changes, on-screen text, and other visual content. In standard audio description, narration is added during existing pauses in dialogue.

* ***extended audio description***: An [audio description](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-description) in which the [video](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-video) is sometimes paused so that there is time to add additional description.

***audio track***

All or part of the [audio](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio) portion of a presentation (e.g. each instrument may have a track, or each stereo channel may have a track).

***author***

A person who works alone or collaboratively to create [content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) (e.g. content author, designer, programmer, publisher, tester).

***captions***

A type of [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) that takes the form of text presented and synchronized with time-based media to provide not only the speech, but also non-speech information conveyed through sound, including meaningful sound effects and identification of speakers. In some countries, the term "subtitle" is used to refer to dialogue only and "captions" is used as the term for dialogue plus sounds and speaker identification. In other countries, "subtitle" (or its translation) is used to refer to both.

* ***open captions***: Captions that are always rendered with a visual track; they cannot be turned off.
* ***closed captions***: Captions that may be turned on and off. The captions requirements of UAAG 2.0 assume that the user agent can [recognize](http://www.w3.org/TR/IMPLEMENTING-UAAG20/" \l "def-recognize" \o "definition: Recognize)the captions as such.

*Note:* Other terms that include the word "caption" may have different meanings. For instance, a "table caption" is a title for a table, often positioned graphically above or below the table.

***command*s**

Actions made by users to control the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). These include:

* ***direct commands***: Commands that apply to a specified item (e.g. button) or action (e.g. save function), regardless of the current focus location:
  + ***direct navigation commands***: Commands that move focus to a specified item.
  + ***direct activation commands***: Commands that activate the specified item (and may also move focus to it) or action.
* ***sequential navigation commands*** (sometimes called "logical navigation commands" or "linear navigation commands"): Commands that move focus forwards and backwards through a list of items. The element list being navigated may be the list of all elements or just a subset (e.g. the list of headers, the list of links).
* ***spatial commands*** (sometimes called "directional commands"): Commands that require the user to be aware of the spatial arrangement of items on the screen:
  + ***spatial navigation commands***: Commands that move from one item to another based on direction on the screen.
  + ***spatial manipulation commands***: Commands that resize or reposition an item on the screen.
* ***structural navigation commands***: Commands that move forwards, backwards, up and down a hierarchy.

***content (web content)***

Information and sensory experience to be communicated to the user by means of a [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent), including code or markup that defines the content's structure, presentation, and interactions.

***continuous scale***

When interacting with a time-based media presentation, a continuous scale allows user (or programmatic) action to set the active playback position to any time point on the presentation timeline. The granularity of the positioning is determined by the smallest resolvable time unit in the media timebase.

***default***

see [***properties***](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-Properties-and-Values)

***disabled element***

see [**element**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element)

***document character set***

The internal representation of data in the [source](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text-source) content by a [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent).

***document object*,*****Document Object Model (DOM)***

A platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. Overview of DOM-related materials: <http://www.w3.org/DOM/#what>.

***documentation***

Any information that supports the use of a [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). This information may be provided electronically or otherwise and includes help, manuals, installation instructions, tutorials, etc. Documentation may be accessed in various ways (e.g. as files included in the installation, available on the web).   
Note: The level of technical detail in documentation for users should match the technical level of the feature. For example, user documentation for a browser's zoom function should not refer users to the source code repository for that browser.

***element*,*****element type***

Primarily, a syntactic construct of a document type definition (DTD) for its application. This is the sense employed by the XML 1.0 specification ([*[XML]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-XML), section 3). This sense is also relevant to structures defined by XML schemas. UAAG 2.0 also uses the term "element" more generally to mean a type of content (such as video or sound) or a logical construct (such as a header or list).

* ***enabled element***: An [element](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element) with associated behaviors that can be activated through the user interface or through an [API](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-api). The set of elements that a user agent enables is generally derived from, but is not limited to, the set of [elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element) defined by implemented markup languages.
* ***disabled element***: A potentially enabled element that is not currently available for activation (e.g. a "grayed out" menu item).

***events and scripting, event handler, event type***

User agents often perform a task when an event having a particular "event type" occurs, including a user interface event, a change to content, loading of content, or a request from the [operating environment](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-environment). Some markup languages allow authors to specify that a script, called an***event handler***, be executed when an event of a given type occurs. An event handler is ***explicitly associated with an element*** through scripting, markup or the [DOM](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-dom).

***enabled element***

see [**element**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element)

***explicit user request***

An interaction by the user through the [user agent user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui), the [focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-focus), or the [selection](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-selection). User requests are made, for example, through[user agent user interface controls](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ui-control) and [keyboard commands](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-command). Some examples of explicit user requests include when the user selects "New viewport," responds "yes" to a [prompt](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-prompt) in the user agent's user interface, configures the user agent to behave in a certain way, or changes the selection or focus with the keyboard or pointing device. *Note:*Users can make errors when interacting with the user agent. For example, a user may inadvertently respond "yes" to a prompt instead of "no." This type of error is still considered an explicit user request.

***extended audio description***

see [**audio description**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-description)

***focus, input focus***

The location where input will occur if a [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport) is active. Examples include:

* ***keyboard focus***: The screen location where [keyboard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard) input will occur if a viewport is active.
* ***pointing device focus***: The screen location where [pointer](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-pointer) input will occur if a viewport is active. There can be multiple pointing device foci for example when using a screen sharing utility there is typically one for the user's physical mouse and one for the remote mouse.

The ***active input focus*** is in the active viewport. The ***inactive input focus*** is in the inactive viewport. Focus is typically indicated by a[focus cursor](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-cursor).

***focus cursor***

Visual indicator that [highlights](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-highlight) a user interface element to show that it has [input focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-input-focus) (e.g. the dotted line around a button, outline around a pane, or brightened title bar on a window).

* ***keyboard focus cursor***: Indicator showing the enabled element with current input focus where focus has been moved by the keyboard (e.g. the dotted line around a button).
* ***text cursor***: Indicator showing where keyboard input will occur in text (e.g. the flashing vertical bar in a text field, also called a caret).
* ***pointer***: Indicator showing where pointing device input will occur. The indicator can be moved with a pointing device or emulator such as a mouse, pen tablet, keyboard-based mouse emulator, speech-based mouse commands, or 3-D wand. A pointing device click typically moves the[input focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-input-focus) to the pointer location. The indicator may change to reflect different states. When touchscreens are used, the "pointing device" is a combination of the touchscreen and the user's finger or stylus. On most touchscreen systems there is no pointer (on-screen visual indication).

Cursors are active when in the active viewport, and inactive when in an inactive viewport.

***focusable element***

Any [element](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-element) capable of having [input focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-input-focus) (e.g. a link, text box, or menu item). In order to be accessible and fully usable, every focusable element should take [keyboard focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-focus), and ideally would also take [pointer focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-pointing-device-focus).

***globally, global configuration***

A setting is one that applies to the entire [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) or all content being rendered by it, rather than to a specific feature within the user agent or a specific document being viewed.

***graphical***

Information (e.g. text, colors, graphics, images, or [animations](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-animation)) rendered for visual consumption.

***highlight, highlighted, highlighting***

Emphasis indicated through the user interface. For example, [user agents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) highlight content that is selected, focused, or matched by a search operation. Graphical highlight mechanisms include dotted boxes, changed colors or fonts, underlining, adjacent icons, magnification, and reverse video. Synthesized speech highlight mechanisms include alterations of voice pitch and volume ( i.e. speech prosody). User interface items may also be highlighted, for example a specific set of foreground and background colors for the title bar of the active window. Content that is highlighted may or may not be a [selection](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-selection).

***image***

Pictorial content that is static (i.e. not moving or changing). Also see [animation](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-animation).

***important elements***

Elements determined as important by the user to facilitate the user's navigation of the content. UAAG 2.0 intentionally does not identify which important elements must be navigable because this will vary by user needs and technologies being used.

***informative (non-normative)***

see [normative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-informative)

***keyboard***

The letter, symbol and command keys or key indicators that allow a user to control a computing device. Assistive technologies have traditionally relied on the keyboard interface as a universal, or modality independent interface. In this document references to keyboard include keyboard emulators and [keyboard interfaces](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-interface) that make use of the keyboard's role as a modality independent interface (see [Modality Independent Controls](http://www.w3.org/TR/2013/WD-UAAG20-20131107/#intro-modality-independence)). Keyboard emulators and interfaces may be used on devices which do not have a physical keyboard, such as mobile devices based on touchscreen input.

***keyboard interface***

Keyboard interfaces are programmatic services provided by many platforms that allow operation in a device independent manner. A keyboard interface can allow keystroke input even if particular devices do not contain a hardware keyboard (e.g. a touchscreen-controlled device can have a keyboard interface built into its [operating system](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-system) to support onscreen keyboards as well as external keyboards that may be connected).   
*Note:* Keyboard-operated mouse emulators, such as MouseKeys, do not qualify as operation through a keyboard interface because these emulators use pointing device interfaces, not keyboard interfaces.

***keyboard command* (*****keyboard binding*,*****keyboard shortcuts*, *accesskey, access key,******accelerator keys*, *direct keyboard command*)**

A key or set of keys that are tied to a particular UI control or application function, allowing the user to navigate to or activate the control or function without traversing any intervening controls (e.g. CTRL+"S" to save a document). It is sometimes useful to distinguish keyboard commands that are associated with controls that are rendered in the current context (e.g. ALT+"D" to move focus to the address bar) from those that may be able to activate program functionality that is not associated with any currently rendered controls (e.g. "F1" to open the Help system). Keyboard commands can be triggered using a physical keyboard or keyboard emulator (e.g. on-screen keyboard or speech recognition). (See [Modality Independent Controls](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#intro-modality-independence)). ***Sequential keyboard commands*** require multiple keystrokes to carry out an action (e.g. a series of Tab or arrow presses followed by Enter, or a sequence like ALT-F, V to drop down a File menu and choose Print Preview).

***non-text content (non-text element, non-text equivalent)***

see [***text***](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text)

***normative*,*****informative (non-normative)***

Required (or not required) for conformance. Abilities identified as "normative" are required for [conformance](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#conformance) (noting that one may conform in a variety of well-defined ways to UAAG 2.0). Abilities identified as "informative" (or, "non-normative") are never required for conformance.

***notify***

To make the user aware of events or status changes. Notifications can occur within the user agent user interface (e.g. a status bar) or within the content display. Notifications may be passive and not require user acknowledgment, or they may be presented in the form of a [prompt](http://www.w3.org/TR/IMPLEMENTING-UAAG20/" \l "def-prompt)requesting a user response (e.g. a confirmation dialog).

***obscure***

To render a visual element in the same screen space as a second visual element in a way that prevents the second visual element from being visually perceived.   
Note: The use of transparent backgrounds for the overlaying visual element (e.g., video captions) is an acceptable technique for reducing obscuration, if space is available.

***operating environment***

The software environment that governs the user agent's operation, whether it is an [operating system](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-system) or a programming language environment such as Java.

***operating system (OS)***

Software that supports a device's basic functions, such as scheduling tasks, executing applications, and managing hardware and peripherals.   
*Note:*Many operating systems mediate communication between executing applications and [assistive technology](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-assistive-technology) via a [platform accessibility service](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-access-platform-arch).

***override***

When one configuration or behavior preference prevails over another. Generally, the requirements of UAAG 2.0 involve user preferences prevailing over author preferences and user agent default settings and behaviors. Preferences may be multi-valued in general (e.g. the user prefers blue over red or yellow), and include the special case of two values (e.g. turn on or off blinking text content).

***placeholder***

Content generated by the user agent to replace author-supplied content. A placeholder may be generated as the result of a user preference (e.g. to not render images) or as [repair content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-repair-content) (e.g. when an image cannot be found). A placeholder can be any type of content, including text, images, and audio cues. A placeholder should identify the technology of the replaced object.

***platform***

The software and hardware environment(s) within which the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) operates. Platforms provide a consistent operational environment. There may be layers of software in an hardware architecture and each layer may be considered a platform. Non-web-based platforms include desktop[operating system](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-system) (e.g. Linux, Mac OS, Windows, etc.), mobile operating systems (e.g. Android, Blackberry, iOS, Windows Phone, etc.), and cross-OS environments (e.g. Java). Web-based platforms are other user agents. User agents may employ server-based processing, such as web content transformations, text-to-speech production, etc.  
*Note 1*: A user agent may include functionality hosted on multiple platforms (e.g. a browser running on the desktop may include server-based pre-processing and web-based documentation).  
*Note 2*: Accessibility guidelines for developers exist for many platforms.

***platform accessibility service***

A programmatic interface that is engineered to enhance communication between mainstream software applications and assistive technologies (e.g. MSAA, UI Automation, and IAccessible2 for Windows applications, AXAPI for Mac OSX applications, Gnome Accessibility Toolkit API for GNOME applications, Java Access for Java applications). On some platforms it may be conventional to enhance communication further by implementing a[DOM](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-dom).

***plug-in***

see [**user agent**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent)

***point of regard***

The position in [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content) that the user is presumed to be viewing. The dimensions of the point of regard may vary. For example,it may be a two-dimensional area (e.g. content rendered through a two-dimensional graphical viewport), or a point (e.g. a moment during an audio rendering or a cursor position in a graphical rendering), or a range of text (e.g. focused text), or a two-dimensional area (e.g. content rendered through a two-dimensional graphical viewport). The point of regard is almost always within the viewport, but it may exceed the spatial or temporal [dimensions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#viewport-dimension) of the viewport (see the definition of [rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content) for more information about viewport dimensions). The point of regard may also refer to a particular moment in time for content that changes over time (e.g. an audio-only presentation). User agents may determine the point of regard in a number of ways, including based on viewport position in content, [keyboard focus](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-keyboard-focus), and [selection](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-selection).

***pointer***

see [**focus cursor**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-pointer)

***profile***

A named and persistent representation of user preferences that may be used to configure a [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). Preferences include input configurations, style preferences, and natural language preferences. In [operating environments](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-operating-environment) with distinct user accounts, profiles enable users to reconfigure software quickly when they log on. Users may share their profiles with one another. Platform-independent profiles are useful for those who use the same user agent on different devices.

***programmatically available***

Information that is encoded in a way that allows different software, including [assistive technologies](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-assistive-technology), to extract and use the information relying on published, supported mechanisms, such as, platform accessibility services, APIs, or the document object models (DOM). For [web-based user interfaces](http://www.w3.org/WAI/AU/ATAG20/#def-Web-Based-UI), this means ensuring that the [user agent](http://www.w3.org/WAI/AU/ATAG20/#def-User-Agent) can pass on the information (e.g. through the use of WAI-ARIA). Something is programmatically available if the entity presenting the information does so in a way that is explicit and unambiguous, in a way that can be understood without reverse-engineering or complex (and thus potentially fallible) heuristics, and only relying on methods that are published, and officially supported by the developers of the software being evaluated.

***prompt***

Any [user agent](http://www.w3.org/WAI/AU/ATAG20/#def-User-Agent)-initiated request for a decision or piece of information from a user.

***properties, values, and defaults***

A user agent renders a document by applying formatting algorithms and style information to the document's elements. Formatting depends on a number of factors, including where the document is rendered (e.g. on screen, on paper, through loudspeakers, on a braille display, on a mobile device). Style information (e.g. fonts, colors, synthesized speech prosody) may come from the elements themselves (e.g. certain font and phrase elements in HTML), from stylesheets, or from user agent settings. For the purposes of these guidelines, each formatting or style option is governed by a property and each property may take one value from a set of legal values. Generally in UAAG 2.0, the term "[property](http://www.w3.org/TR/1998/REC-CSS2-19980512/conform.html)" has the meaning defined in [CSS 2.1 Conformance](http://www.w3.org/TR/CSS2/conform.html#property) ([*[CSS21]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-CSS2), ). A reference to "styles" in UAAG 2.0 means a set of style-related properties.

* ***default value***: The value given to a property by a user agent at installation.

***recognize***

[Authors](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author) encode information in many ways, including in markup languages, style sheet languages, scripting languages, and protocols. When the information is encoded in a manner that allows the user agent to process it with certainty, the user agent can "recognize" the information. For instance, HTML allows authors to specify a heading with the H1 element, so a user agent that implements HTML can recognize that content as a heading. If the author creates a heading using a visual effect alone (e.g. just by increasing the font size), then the author has encoded the heading in a manner that does not allow the user agent to recognize it as a heading. Some requirements of UAAG 2.0 depend on content roles, content relationships, timing relationships, and other information supplied by the author. These requirements only apply when the author has encoded that information in a manner that the user agent can recognize. See the section on [conformance](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#conformance) for more information about applicability. User agents will rely heavily on information that the author has encoded in a markup language or style sheet language. Behaviors, style, meaning encoded in a [script](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-script), and markup in an unfamiliar XML namespace may not be recognized by the user agent as easily or at all.

***reflowable content***

[Web content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) that can be arbitrarily wrapped over multiple lines. The primary exceptions to reflowable content are graphics and [video](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-video).

***relative time units***

Time intervals for navigating media relative to the current point (e.g. move forward 30 seconds). When interacting with a time-based media presentation, a user may find it beneficial to move forward or backward via a time interval relative to their current position. For example, a user may find a concept unclear in a video lecture and elect to skip back 30 seconds from the current position to review what had been described. Relative time units may be preset by the user agent, configurable by the user, and/or automatically calculated based upon media duration (e.g. jump 5 seconds in a 30-second clip, or 5 minutes in a 60-minute clip). Relative time units are distinct from absolute time values such as the 2 minute mark, the half-way point, or the end.

***rendered content***

The presentation generated by the user agent based on the author supplied code. This includes text characters, images, stylesheets, scripts, and any other content that, once processed, may be perceived.

* ***rendered text***: [Text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text) content that is rendered in a way that communicates information about the characters themselves, whether visually or as synthesized speech.

***repair content*,*****repair text***

Content generated by the user agent to correct an error condition. "Repair text" refers to the [text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text) portion of repair content. Error conditions that may lead to the generation of repair content include:

* Erroneous or incomplete content (e.g. ill-formed markup, invalid markup, or missing [alternative content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-conditional-content) that is required by format specification);
* Missing resources for handling or rendering content (e.g. the user agent lacks a font family to display some characters, or the user agent does not implement a particular scripting language).

*Note:* UAAG 2.0 does not require user agents to include repair content in the [document object](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-document-object). Repair content inserted in the document object should conform to the Web Content Accessibility Guidelines 2.0 [*[WCAG20]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-WCAG20). For more information about repair techniques for web content and software, refer to "Techniques for Authoring Tool Accessibility Guidelines 1.0" [*[ATAG10-TECHS]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-ATAG10-TECHS).

***script***

Instructions to create dynamic web content that are written in a programming (scripting) language. In guidelines referring to the written (natural) language of content, as referenced in Unicode [*[UNICODE]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-UNICODE)), script can also refer to "a collection of symbols used to represent textual information in one or more writing systems". Information encoded in (programming) scripts may be difficult for a user agent to[recognize](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize). For instance, a user agent is not expected to recognize that, when executed, a script will calculate a factorial. The user agent will be able to recognize some information in a script by virtue of implementing the scripting language or a known program library (e.g. the user agent is expected to recognize when a script will open a viewport or retrieve a resource from the web).

***selection***

A [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) mechanism for identifying a (possibly empty) range of content that will be the implicit source or target for subsequent operations. The selection may be used for a variety of purposes, including for cut-and-paste operations, to designate a specific element in a document for the purposes of a query, and as an indication of [point of regard](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-point-of-regard) (e.g. the matched results of a search may be automatically selected). The selection should be [highlighted](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-highlight) in a distinctive manner. On the screen, the selection may be highlighted in a variety of ways, including through colors, fonts, graphics, and magnification. When rendered using synthesized speech, the selection may be highlighted through changes in pitch, speed, or prosody.

***source text***

Text that the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) renders upon user request to view the source of specific [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport) content (e.g. selected content, frame, page).

***style properties***

Properties whose values determine the presentation (e.g. font, color, size, location, padding, volume, synthesized speech prosody) of content elements as they are rendered (e.g. onscreen, via loudspeaker, via braille display) by [user agents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). Style properties can have several origins:

* ***user agent default styles***: The default style property values applied in the absence of any author or user styles. Some [web content technologies](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-Web-Content-Technology) specify a default rendering; others do not.
* ***author styles***: Style property values that are set by the author as part of the content (e.g. in-line styles,[author style sheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author-stylesheet)).
* ***user styles***: Style property values that are set by the user (e.g. via user agent interface settings, [user style sheets](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-stylesheet)).

***style sheet***

A mechanism for communicating [style property](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-style-properties) settings for web content, in which the style property settings are separable from other content resources. This separation allows author style sheets to be toggled or substituted, and user style sheets defined to apply to more than one resource. Style sheet [web content technologies](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-Web-Content-Technology) include Cascading Style Sheets (CSS) and Extensible Stylesheet Language (XSL).

* ***user style sheet***: Stylesheets specified by the user, resulting in [user styles](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-styles).
* ***author style sheet***: Style sheets specified by the author, resulting in [author styles](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author-styles).

***synchronize***

The act of time-coordinating two or more presentation components (e.g. a [visual track](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-visual-track) with captions, several tracks in a multimedia presentation). For authors, the requirement to synchronize means to provide the data that will permit sensible time-coordinated rendering by a user agent. For example, web content developers can ensure that the segments of caption text are neither too long nor too short, and that they map to segments of the visual track that are appropriate in length. For user agent developers, the requirement to synchronize means to present the content in a sensible time-coordinated fashion under a wide range of circumstances including technology constraints (e.g. small text-only displays), user limitations (e.g. slow reading speeds, large font sizes, high need for review or repeat functions), and content that is sub-optimal in terms of accessibility.

***technology (web content technology)***

A mechanism for encoding instructions to be rendered, played or executed by [user agents](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent). Web content technologies may include markup languages, data formats, or programming languages that [authors](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-author) may use alone or in combination to create end-user experiences that range from static web pages to multimedia presentations to dynamic web applications. Some common examples of web content technologies include HTML, CSS, SVG, PNG, PDF, Flash, and JavaScript.

**text**

A sequence of characters that are [programmatically available](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-programmatically-avail), where the sequence is expressing something in human language.

* ***non-text content***: Any content that is not a sequence of characters that can be programmatically determined or where the sequence is not expressing something in human language Note: This includes ASCII Art (which is a pattern of characters), emoticons, leetspeak (which uses character substitution), and images representing text.
* ***blocks of text***: More than one sentence of text [ based on [WCAG definition: blocks of text](http://www.w3.org/TR/WCAG20/#blockstextdef)]

***text transcript***

A type of alternative content that takes the form of text equivalents of audio information (e.g. an audio-only presentation or the [audio track](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-audio-track)of a movie or other animation). A text transcript provides text for both spoken words and non-spoken sounds such as sound effects. Text transcripts make audio information accessible to people who have hearing disabilities and to people who cannot play the audio. Text transcripts are usually created by hand but may be generated on the fly (e.g. by voice-to-text converters).

***top-level viewport***

see [**viewport**](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport)

***user agent***

Any software that retrieves, renders and facilitates end user interaction with web content. UAAG 2.0 identifies four user agent architectures:

* ***platform-based user agent, non-web-based user agent***: User agents that run on non-web platforms (operating systems and cross-OS platforms, such as Java) and perform content retrieval, rendering and end-user interaction facilitation themselves (e.g. Firefox, Internet Explorer, Chrome, Opera, Windows Media Player, QuickTime Pro, RealPlayer).
* ***embedded user agent, plug-in***: User agents that "plug-in" to other user agents or applications (e.g. media player plug-in for a web browser, web view component). Embedded user agents may establish direct connections with the platform (e.g. communication via platform accessibility services).
* ***web-based user agent***: User agents that have user interfaces that are implemented using web content technologies and are accessed by users via a user agent. web-based user agents transform content into web content technologies that the host user agent can render (e.g. Google Docs, Bing Translator, Yahoo Mail).

*Note:* Many web applications retrieve, render and facilitate interaction with very limited data sets (e.g. online ticket booking). In such cases, WCAG 2.0, without UAAG 2.0, may be appropriate for assessing the application's accessibility.   
Examples of software that are generally considered user agents under UAAG 2.0:

* Desktop web browsers (e.g. Internet Explorer, Firefox, Chrome, Safari, Opera)
* Mobile web browsers (e.g. Firefox, Chrome, Safari, Android Browser, Opera Mini, Atomic Web, Puffin)
* Browser plug-ins (e.g. QuickTime Plug-in for Firefox, Acrobat Reader Plug-in for Internet Explorer, Shockwave Plug-in for Chrome)
* Web view components (e.g. Webkit Webview component, Web Tools Platform Plug-in for Eclipse, UIWebView for iOS)
* Authoring tools that render the web content being edited (e.g. Word, Dreamweaver, HTML-Kit)

Examples of software that are not considered user agents under UAAG 2.0 (in all cases, WCAG 2.0 still applies if the software is web-based):

* Operating environments or software bundles that include platform-based user agents (e.g. Windows, OS X, KDE, iOS), though the included user agents themselves are covered by UAAG 2.0.
* General-purpose platforms or toolkits that don't use web technologies, even though they may be used by user agents for other purposes (e.g. GNOME, KDE, .NET Framework/CLR).
* Narrow-purpose platform-based or web applications (e.g. online ticket booking applications).
* Authoring tools that only display a source view of the web content being edited (e.g. Notepad, Vim).

***user agent extension (add-in)***

Software installed into a [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) that adds one or more additional features that modify the behavior of the user agent. Two common capabilities for user agent extensions are the ability to:

* modify the [content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content) before the user agent renders it (e.g. to add highlights if certain types of alternative content are present), and
* modify the user agent's own [user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui) (e.g. add a headings view).

***user interface***

For the purposes of UAAG 2.0, the user interface includes both:

* ***user agent user interface***: The [controls](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ui-control) (e.g. menus, buttons, [prompts](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-prompt), native audio/video player controls, and other components for input and output) and mechanisms (e.g. selection and focus) provided by the [user agent](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-user-agent) that are not created on the basis of the author-supplied [content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content). The user agent user interface may include [extensions](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-extension) that become part of the user agent user interface (e.g. toolbars, additional menus).
* ***content user interface***: The user interface that emerges from the user agent rendering of the author-supplied content. It includes all[rendered content](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-rendered-content) (e.g. [text](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-text), headings, [enabled elements](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-enabled-element), disabled elements, author-supplied audio/video controls).   
  *Note:* There may be a mix of [recognized](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-recognize) and unrecognized user interface controls depending on the author-supplied content.

This document distinguishes user agent user interface and content user interface only where required for clarity.

***user interface control***

A component of the [user agent user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-ua-ui) or the [content user interface](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-content-ui), distinguished where necessary.

***video***

The technology of moving pictures or images. Video can be made up of animated or photographic images, or both.

***view***

A user interface function that lets users interact with web content. UAAG 2.0 recognizes a variety of approaches to presenting the content in a view, including:

* ***rendered view***: A view where content is presented such that it is rendered, played or executed. There are two sub-types:
  + In conventionally rendered views the content is rendered, played or executed according to the web content technology specification. This is the default view of most user agents.
  + In unconventionally rendered views the content is rendered quite differently than specified in the technology specification (e.g. rendering an audio file as a graphical wavefront).
* ***source view***: A view where the web content is presented without being rendered, played or executed. The source view may be plain text (i.e. "View Source") or it may include some other organization (e.g. presenting the markup in a tree).
* ***outline view***: A view where only a subset of the rendered content is presented, usually composed of labels or placeholders for important structural elements. The important structural elements will depend on the web content technology, but may include headings, table captions, and content sections.

*Note:*A view can be visual, audio, or tactile.

***viewport***

A mechanism for presenting only part of a visual or tactile [view](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-view) to the user via a screen or tactile display. There may be multiple viewports on to the same underlying view (e.g. when a split-screen is used to present the top and bottom of a document simultaneously) and viewports may be nested (e.g. a scrolling frame located within a larger document). When the viewport is smaller than the view it is presenting, some of the view will not be presented. Mechanisms are typically provided to move the view or the viewport such that all of the view can be brought into the viewport (e.g. scrollbars).

**Note**: In UAAG 1.0 viewports were defined as having a temporal dimension. In UAAG 2.0, this is not the case. Since audio content is inherently time-based, audio viewports are excluded.

* ***top-level viewport***: A viewport that is not contained within another viewport of a platform-based user agent. Web-based user agents are always displayed inside another viewport, and therefore are never top-level viewports. A popular browser implementation is to provide a window that includes some user agent user interface elements (e.g., menus) and a series of tabbed panels, each of which contains additional user agent user interface elements (e.g., address bar, bookmarks, back/forward buttons) and a top-level viewport for rendering a view of the addressed web resource.

***viewport dimensions***

The onscreen size of a [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport), or the temporal duration of a viewport displaying time-based media. When the dimensions (spatial or temporal) of rendered content exceed the dimensions of the viewport, the user agent provides mechanisms such as scroll bars and advance and rewind controls so that the user can access the rendered content "outside" the viewport (e.g. when the user can only view a portion of a large document through a small graphical viewport, or when audio content has already been played).

***visual-only***

Content consisting exclusively of one or more [visual tracks](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-visual-track) presented concurrently or in series (e.g. a silent movie is an example of a visual-only presentation).

***visual track***

Content rendered through a graphical [viewport](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-viewport). Visual objects include graphics, text, and visual portions of movies and other animations. A visual track is a visual object that is intended as a whole or partial presentation. A visual track does not necessarily correspond to a single physical object or software object.

***voice browser***

A device (hardware and software) that interprets voice markup languages to generate voice output, interpret voice input, and possibly accept and produce other modalities of input and output. Definition from "Introduction and Overview of W3C Speech Interface Framework"[*[VOICEBROWSER]*](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#ref-VOICEBROWSER).

***web resource***

Anything that can be identified by a Uniform Resource Identifier (URI).

Appendix B: How to refer to UAAG 2.0 from other documents

This section is [informative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-informative).

For the most up-to-date information, see "Referencing and Linking to WAI Guidelines and Technical Documents" at <<http://www.w3.org/WAI/intro/linking.html>>.

There are two recommended ways to refer to the "User Agent Accessibility Guidelines 2.0" (and to W3C documents in general):

1. References to a specific version of "User Agent Accessibility Guidelines 2.0." For example, use the "this version" URI to refer to the current document:  
   <http://www.w3.org/TR/2010/WD-UAAG20-20100617/>
2. References to the latest version of "User Agent Accessibility Guidelines 2.0." Use the "latest version" URI to refer to the most recently published document in the series:   
   <http://www.w3.org/TR/UAAG20/>.

The top of UAAG 2.0 includes the relevant catalog metadata for specific references (including title, publication date, "this version" URI, editors' names, and copyright information).

An XHTML 1.0 paragraph including a reference to this specific document might be written:

<p>  
<cite><a href="http://www.w3.org/TR/2010/WD-UAAG20-20100617/">  
"User Agent Accessibility Guidelines 2.0,"</a></cite>  
J. Allan, K. Ford, J. Spellman, eds.,  
W3C Recommendation, http://www.w3.org/TR/ATAG20/.  
The <a href="http://www.w3.org/TR/ATAG20/">latest version</a> of this document is available at http://www.w3.org/TR/ATAG20/.</p>

For very general references to this document (where stability of content and anchors is not required), it may be appropriate to refer to the latest version of this document. Other sections of this document explain how to build a [conformance claim](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#conformance).

Appendix C: References

This section is [informative](http://www.w3.org/TR/IMPLEMENTING-UAAG20/#def-informative).

For the **latest version** of any W3C specification please consult the list of [W3C Technical Reports](http://www.w3.org/TR/) at http://www.w3.org/TR/. Some documents listed below may have been superseded since the publication of UAAG 2.0.

**Note:** In UAAG 2.0, bracketed labels such as "[WCAG20]" link to the corresponding entries in this section. These labels are also identified as references through markup.

**[ATAG10]**

[*"Authoring Tool Accessibility Guidelines 1.0,"*](http://www.w3.org/TR/2000/REC-ATAG10-20000203/) J. Treviranus, C. McCathieNevile, I. Jacobs, and J. Richards, eds., 3 February 2000. This W3C Recommendation is http://www.w3.org/TR/2000/REC-ATAG10-20000203/.

**[ATAG10-TECHS]**

[*"Techniques for Authoring Tool Accessibility Guidelines 1.0,"*](http://www.w3.org/TR/2002/NOTE-ATAG10-TECHS-20021029/) J. Treviranus, C. McCathieNevile, J. Richards, eds., 29 Oct 2002. This W3C Note is http://www.w3.org/TR/2002/NOTE-ATAG10-TECHS-20021029/.

**[CHARMOD]**

[*"Character Model for the World Wide Web,"*](http://www.w3.org/TR/2002/WD-charmod-20020430/) M. Dürst and F. Yergeau, eds., 30 April 2002. This W3C Working Draft is http://www.w3.org/TR/2002/WD-charmod-20020430/. The [latest version](http://www.w3.org/TR/charmod/) is available at http://www.w3.org/TR/charmod/.

**[CSS21]**

[*"Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification,"*](http://www.w3.org/TR/2011/REC-CSS2-20110607/) B. Bos, T. Celik, I. Hickson, H. Lie, eds., 07 June 2011. This W3C Recommendation is http://www.w3.org/TR/2011/REC-CSS2-20110607/.

**[DOM2HTML]**

[*"Document Object Model (DOM) Level 2 HTML Specification,"*](http://www.w3.org/TR/2002/PR-DOM-Level-2-HTML-20021108/) J. Stenback, P. Le Hégaret, A. Le Hors, eds., 8 November 2002. This W3C Proposed Recommendation is http://www.w3.org/TR/2002/PR-DOM-Level-2-HTML-20021108/. The [latest version](http://www.w3.org/TR/DOM-Level-2-HTML/) is available at http://www.w3.org/TR/DOM-Level-2-HTML/.

**[HTML4]**

[*"HTML 4.01 Recommendation,"*](http://www.w3.org/TR/1999/REC-html401-19991224/) D. Raggett, A. Le Hors, and I. Jacobs, eds., 24 December 1999. This W3C Recommendation is http://www.w3.org/TR/1999/REC-html401-19991224/.

**[RFC2616]**

[*"Hypertext Transfer Protocol — HTTP/1.1,"*](http://www.ietf.org/rfc/rfc2616.txt) J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, June 1999.

**[RFC3023]**

[*"XML Media Types,"*](http://www.ietf.org/rfc/rfc3023.txt) M. Murata, S. St. Laurent, D. Kohn, January 2001.

**[SMIL]**

[*"Synchronized Multimedia Integration Language (SMIL) 1.0 Specification,"*](http://www.w3.org/TR/1998/REC-smil-19980615/) P. Hoschka, ed., 15 June 1998. This W3C Recommendation is http://www.w3.org/TR/1998/REC-smil-19980615/.

**[SMIL20]**

[*"Synchronized Multimedia Integration Language (SMIL 2.0) Specification,"*](http://www.w3.org/TR/2001/REC-smil20-20010807/) J. Ayars, et al., eds., 7 August 2001. This W3C Recommendation is http://www.w3.org/TR/2001/REC-smil20-20010807/.

**[SVG]**

[*"Scalable Vector Graphics (SVG) 1.0 Specification,"*](http://www.w3.org/TR/2001/REC-SVG-20010904/) J. Ferraiolo, ed., 4 September 2001. This W3C Recommendation is http://www.w3.org/TR/2001/REC-SVG-20010904/.

**[UAAG10]**

"[User Agent Accessibility Guidelines 1.0](http://www.w3.org/TR/2002/REC-UAAG10-20021217/)," I. Jacobs, J. Gunderson, E. Hansen, eds.17 December 2002. This W3C Recommendation is available at http://www.w3.org/TR/2002/REC-UAAG10-20021217/.

**[UAAG10-CHECKLIST]**

An appendix to UAAG 2.0 lists all of the checkpoints, sorted by priority. The checklist is available in either [tabular form](http://www.w3.org/TR/2002/REC-UAAG10-20021217/uaag10-chktable) or [list form](http://www.w3.org/TR/2002/REC-UAAG10-20021217/uaag10-chklist).

**[UAAG10-ICONS]**

Information about [UAAG 1.0 conformance icons](http://www.w3.org/WAI/UAAG10-Conformance) and their usage is available at http://www.w3.org/WAI/UAAG10-Conformance.

**[UAAG10-SUMMARY]**

An appendix to UAAG 2.0 provides a [summary](http://www.w3.org/TR/2002/REC-UAAG10-20021217/uaag10-summary) of the goals and structure of User Agent Accessibility Guidelines 1.0.

**[UAAG10-TECHS]**

[*"Techniques for User Agent Accessibility Guidelines 1.0,"*](http://www.w3.org/TR/UAAG10-TECHS/) I. Jacobs, J. Gunderson, E. Hansen, eds. The latest draft of the techniques document is available at http://www.w3.org/TR/UAAG10-TECHS/.

**[UNICODE]**

The Unicode Consortium. [*The Unicode Standard, Version 6.1.0*](http://www.unicode.org/versions/Unicode6.1.0/), (Mountain View, CA: The Unicode Consortium, 2012. ISBN 978-1-936213-02-3)   
<http://www.unicode.org/versions/Unicode6.1.0/>

**[VOICEBROWSER]**

[*"Introduction and Overview of W3C Speech Interface Framework,"*](http://www.w3.org/TR/2000/WD-voice-intro-20001204/) J. Larson, 4 December 2000. This W3C Working Draft is http://www.w3.org/TR/2000/WD-voice-intro-20001204/. The [latest version](http://www.w3.org/TR/voice-intro/) is available at http://www.w3.org/TR/voice-intro/. UAAG 2.0 includes references to additional W3C specifications about voice browser technology.

**[W3CPROCESS]**

[*"World Wide Web Consortium Process Document,"*](http://www.w3.org/Consortium/Process-20010719/) I. Jacobs ed. The 19 July 2001 version of the Process Document is http://www.w3.org/Consortium/Process-20010719/. The [latest version](http://www.w3.org/Consortium/Process/) is available at http://www.w3.org/Consortium/Process/.

**[WCAG20]**

[*"Web Content Accessibility Guidelines (WCAG) 2.0"*](http://www.w3.org/TR/WCAG20/) B. Caldwell, M. Cooper, L. Guarino Reid, G. Vanderheiden, eds., 8 December 2008. This W3C Recommendation is http://www.w3.org/TR/2008/REC-WCAG20-20081211/. The [latest version](http://www.w3.org/TR/WCAG20/) is available at http://www.w3.org/TR/WCAG20/. Additional format-specific techniques documents are available from this Recommendation.

**[WCAG20-TECHS]**

[*"Techniques for Web Content Accessibility Guidelines 2.0,"*](http://www.w3.org/TR/WCAG10-TECHS/) B. Caldwell, M. Cooper, L. Guarino Reid, G. Vanderheiden, eds., 8 December 2008. This W3C Note is http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/. The [latest version](http://www.w3.org/TR/WCAG20-TECHS/) is available at http://www.w3.org/TR/WCAG20-TECHS/. Additional format-specific techniques documents are available from this Note.

**[WCAG-EM]**

["Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0"](http://www.w3.org/TR/2013/WD-WCAG-EM-20130226/) E. Velleman, S. Abou-Zahra, eds., 26 February 2013. This is an informative draft of a Working Group Note. The [latest version](http://www.w3.org/TR/WCAG-EM/) is available at http://www.w3.org/TR/WCAG-EM/

**[WEBCHAR]**

[*"Web Characterization Terminology and Definitions Sheet,"*](http://www.w3.org/1999/05/WCA-terms/01) B. Lavoie, H. F. Nielsen, eds., 24 May 1999. This is a W3C Working Draft that defines some terms to establish a common understanding about key Web concepts. This W3C Working Draft is http://www.w3.org/1999/05/WCA-terms/01.

**[XAG10]**

[*"XML Accessibility Guidelines 1.0,"*](http://www.w3.org/TR/2002/WD-xag-20021003) D. Dardailler, S. Palmer, C. McCathieNevile, eds., 3 October 2001. This W3C Working Draft is http://www.w3.org/TR/2002/WD-xag-20021003. The [latest version](http://www.w3.org/TR/xag) is available at http://www.w3.org/TR/xag.

**[XML]**

["Extensible Markup Language (XML) 1.0 (Second Edition),"](http://www.w3.org/TR/1998/REC-xml-19980210) T. Bray, J. Paoli, C.M. Sperberg-McQueen, eds., 6 October 2000. This W3C Recommendation is http://www.w3.org/TR/2000/REC-xml-20001006.

**[XHTML10]**

[*"XHTML[tm] 1.0: The Extensible HyperText Markup Language,"*](http://www.w3.org/TR/2000/REC-xhtml1-20000126/) S. Pemberton, et al., 26 January 2000. This W3C Recommendation is http://www.w3.org/TR/2000/REC-xhtml1-20000126/.

**[XMLDSIG]**

[*"XML-Signature Syntax and Processing,"*](http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/) D. Eastlake, J. Reagle, D. Solo, eds., 12 February 2002. This W3C Recommendation is http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.

**[XMLENC]**

[*"XML Encryption Syntax and Processing,"*](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/) D. Eastlake, J. Reagle, eds., 10 December 2002. This W3C Recommendation is http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/.

Appendix D: Acknowledgments

Participants active in the UAWG prior publication:

* Jim Allan (Co-Chair, Texas School for the Blind and Visually Impaired)
* Kelly Ford (Co-Chair, Microsoft)
* Mark Hakkinen (Educational Testing Service)
* Eric Hansen (Educational Testing Service)
* Simon Harper (University of Manchester)
* Greg Lowney (Invited Expert)
* Kimberly Patch (Invited Expert)
* Jan Richards ( Inclusive Design Institute, OCAD University)
* Jeanne Spellman (W3C Staff Contact)

**Previous Editor:**

Jan Richards, Inclusive Design Institute, OCAD University

Additional Contributors of Mobile Examples

* Kathy Walhbin
* Mark Sadecki

Other previously active UAWG participants and other contributors to UAAG 2.0:

* Judy Brewer (W3C)
* Alan Cantor (Invited Expert)
* Wayne Dick (Invited Expert)
* Bim Egan (Royal National Institute of Blind People)
* Sean Hayes (Microsoft)
* Dean Hudson (Apple)
* Patrick Lauke (Opera Software)
* Cathy Laws (IBM)
* Peter Parente (IBM)
* David Poehlman (Invited Expert)
* Simon Pieters (Opera Software)
* Henny Swan (Opera Software)
* Gregory Rosmaita (Invited Expert)
* David Tseng (Apple)

UAAG 2.0 would not have been possible without the work of [those who contributed to UAAG 1.0](http://www.w3.org/TR/WAI-USERAGENT/acks.html#Acknowledgments).

This publication has been funded in part with Federal funds from the U.S. Department of Education, National Institute on Disability and Rehabilitation Research (NIDRR) under contract number ED-OSE-10-C-0067. The content of this publication does not necessarily reflect the views or policies of the U.S. Department of Education, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.