

Introduction to HTML5

Michael(tm) Smith

mike@w3.org

<http://people.w3.org/mike>

sideshowbarker on Twitter, etc.

*I work for the W3C
in Japan, based
at Keio University
near Tokyo*

My official W3C title is:

Special Missions

Subsection Junior Interim

Floor Manager

*Where to start learning
about HTML5?*

HTML5 differences from HTML4

<http://w3.org/TR/html5-diff/>

*Tutorials and
how-to guidance*

Dive into HTML5

<http://diveintohtml5.org/>

The HTML5 Doctors

<http://html5doctor.com/>

*Alternative
authoring references*

HTML5 (Author Edition)

[http://dev.w3.org/html5/
spec-author-view](http://dev.w3.org/html5/spec-author-view)

HTML5: The Markup Language

[http://dev.w3.org/html5/
markup](http://dev.w3.org/html5/markup)

HTML5 differences

- Syntax simplifications
- MathML and SVG integration
- New elements/attributes added
- Removed old elements/attributes
- Added new APIs for scripting

Syntax simplifications

```
<!DOCTYPE html PUBLIC  
"-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/  
DTD/xhtml1-transitional.dtd">
```

```
<!doctype html>
```



```
<meta http-equiv="Content-Type"  
content="text/html;  
charset=Shift_JIS">
```

```
<meta charset="Shift_JIS">
```

New elements

- `<video>` & `<audio>` (no plugins)
- `<canvas>` (scriptable image)
- `<ruby>` (annotations)
- `<article>`, `<section>`, `<header>`
- `<details>`, `<progress>`, more...

New attributes

- **draggable** (drag-and-drop)
- **contenteditable** (editable pages)
- **spellcheck** (catch spelling errors)
- new form attributes (for **client-side validation**, plus new form controls such as **date picker**, etc.)

Elements and attributes removed or obsoleted

- `<frame>`, `<frameset>`
- `<a name>`
- `more...`

New APIs for scripting

- API for `<video>` & `<audio>`
- 2D drawing API for `<canvas>`
- `getElementsByClassName()`
- `innerHTML`
- `more...`

HTML design principles

<http://w3.org/TR/html-design-principles/>

HTML design principles

- Support existing content
- Ensure interoperability
- Precisely define UA behavior
- Handle errors (*non-draconian*)
- Evolution not revolution

Important point:

HTML5 includes XHTML

Frequently Asked Questions (FAQ) about the future of XHTML

<http://w3.org/2009/06/xhtml-faq.html>

About error handling...

Which of these are errors?

- Well-formed XML:

```
<input disabled="disabled" />
```

- Empty attribute: `<input disabled>`

- Without quotes: `<input value=yes>`

- Single quotes: `<input type='checkbox' />`

- Double quotes: `<input name="be evil" />`

This is a real error

<i>misnested tags</i>

HTML5 defines how
browsers can **handle real
errors interoperably and
gracefully.**

*Why is it important to
handle errors?*

More than 93% of
Alexa Top 500 sites
contain are **not**
conformant XHTML

We need to **specify error handling behavior to ensure interoperability** “even in the face of documents that do not comply to the letter of the specifications”.

Authors will write invalid content regardless of what we spec. So the spec states “what authors must not do, and then tells implementors what they must do when an author does it anyway”.

HTML5 Validator

<http://validator.nu>

The Open Web Platform.

Formats:

HTML, CSS, SVG,

ARIA, even JavaScript

The server side can be considered a **black box**.

The open Web platform
also includes **APIs**.

Device-independent,
language-independent
APIs.

APIs that **browsers**
expose to Web
applications
on the *client-side*

Open Web Platform

=

formats + *APIs*

(formats=HTML, CSS, SVG, etc.)

HTML5:

The Web platform as an
*application development/
runtime environment*

Other app-dev environments:

iPhone SDK, Android

SDK, J2ME/MIDP,

BREW, Symbian, Flash

APIs spun off HTML5

- Canvas 2D Context API
- (Canvas 3D API: WebGL*)
- Web Storage API
- Indexed Database API (non-SQL)

APIs spun off HTML5

- Web Workers API
- Web Messaging APIs (cross-doc/postmsg + channel messaging)
- WebSocket API + Protocol; & Server-Sent Events

Formats related to HTML5

- Microdata + Microdata API & RDFa + RDFa API
- WAI-ARIA 1.0 (add accessibility to Web apps)

APIs spun off HTML5

- Canvas 2D Context API
- Canvas 3D API: WebGL *
- Web Storage API
- Indexed Database API (non-SQL)

Other APIs

- Geolocation
- Device Orientation +
Compass
- File API (w/ HTML5 Drag &
Drop)

Other APIs

- **Selectors API**
- **DOM Level 3 Events**
- **DOM4 + Element Traversal**

Other APIs

- **XMLHttpRequest** 1.0 & 2.0
- **Cross-Origin** Resource Sharing

HTML5, CSS3, etc.,
feature detection

<http://modernizr.com/>

HTML5 milestones 2010

- We aim to have HTML5 *start* **Last Call** (LC) in 2010
- To start Last Call:
 - spec is *feature complete*
 - *all* HTML WG issues resolved

HTML5 milestones 2010

- To get to *end* of Last Call:
 - collect comments from community
 - respond to all comments

That's it.

Thank you.