HTML5 Overview

Michael(tm) Smith

W3C HTML WG Co-Chair

W3C Web Apps WG Co-Team Contact

mike@w3.org

with extensive borrowings from Anne van Kesteren and Simon Pieters

Full title

“HTML5: A vocabulary and associated APIs for HTML and XHTML”

original title: “Web Applications 1.0”

Focuses on Web Applications

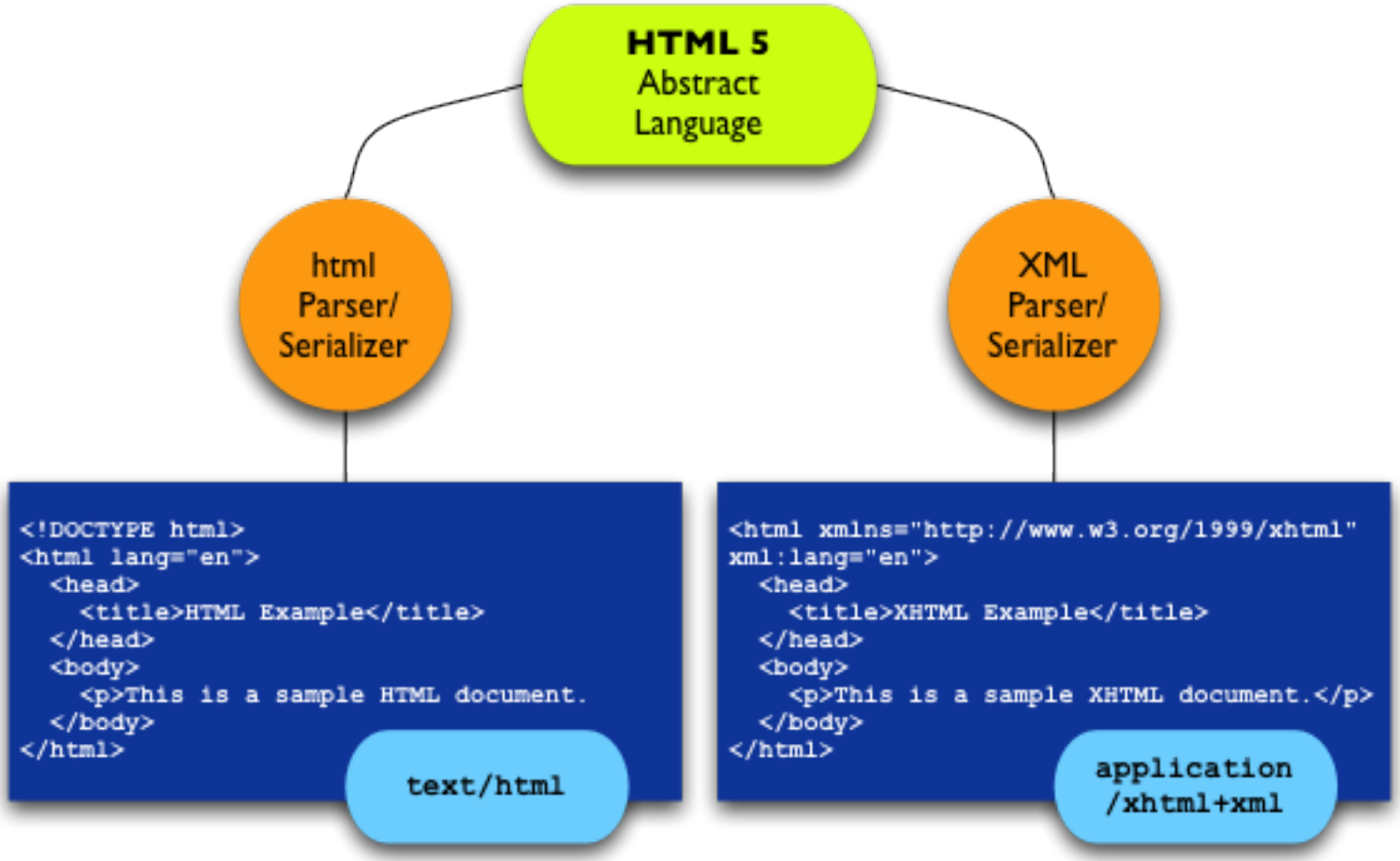
- 10 years since HTML4; in the mean time, among many other things, XHR/ Ajax happened
- what is now the W3C HTML5 draft has been in development since 2004 (by

WHATWG for 3 years before current HTML WG even existed)

HTML as an abstract language

(re)defines HTML as an abstract language

- Defined in terms of the DOM
- HTML serialization: `text/html`
- XML serialization: any XML MIME


HTML 5
Abstract
Language
html
Parser/
Serializer
XML
Parser/
Serializer
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML Example</title>
</head>
<body>
This is a sample HTML document.
</body>
</html>
text/html
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">
<head>
<title>XHTML Example</title>
</head>
<body>
This is a sample XHTML document.
</body>
</html>
application
/xhtml+xml

Other serializations possible?

S-expressions, … ?

What does HTML5 *not* do?

HTML5 does not treat HTML as SGML

HTML5 does not use DTDs

True: Some tools *do* handle HTML as SGML

- W3C Markup Validation service does process HTML as SGML
- http://validator.w3.org/

But: Browsers don't process HTML as SGML

Browsers do not have SGML parsers — they don't check DTDs or follow other SGML parsing rules.

Instead, they use custom parsers built specifically for parsing HTML

HTML5 *does* specify an XML serialization of HTML…

…but HTML5 does not *restrict* HTML to only a (well-formed) XML-based serialization

text/html: attribute syntax

All of the examples below are **conformant** HTML

- 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">`

The HTML DOCTYPE

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

Declaring a character encoding

```
<meta
 http-equiv="Content-Type"
 content="text/html;
          charset=utf-8">
<meta charset="utf-8">
```

Why *not* restrict HTML to well-formed XML (that is, XHTML)?

XML requires **draconian error handling** for all content served as XHTML…

…which means that for any XHTML page that contains even one single minor error…

…a browser must fail to display the page at all…

…and because the vast majority of HTML on the Web is not well-formed XML…

…we need to interoperably handle that "real world" (or "tag soup") HTML

Precisely specifying parsing of “real world” HTML

HTML5 includes a precise algorithm for exactly how conformant UAs/browsers must parse HTML (an algorithm that closely matches existing browser implementations)...

…including parsing of HTML that may not be well-formed XML and is served up as `text/html`

non-browser HTML5 parsers

- html5lib (Python, Ruby)
- the Validator.nu parser (Java)
- C++ library?
- can be plugged into other applications in the same way that XML parsers can be

Formal spec for error handling

Recognizing that many authors produce non-conformant documents and that apps need to deal with such documents, HTML5 precisely specifies handling of markup errors and other classes of errors — so that such errors will be handled in an interoperable way across UAs/browsers

In other words

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”.

see http://www.whatwg.org/news/start

And in yet other words

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”.

see http://esw.w3.org/topic/HTML/DraconianErrorHandling and Ian Hickson’s “Error handling and Web language design”, http://ln.hixie.ch/?start=1074730186

HTML5 conformance checking

- html5.validator.nu (X)HTML5 Validator
- does not use DTDs
- RELAX NG & Schematron for schema validation
- other means of non-schema checking
- a RESTful Web service API

So what's new/different in HTML5?

Support for de facto standards

- innerHTML
- Window object
- others…

Web Forms 2.0

- Client-side validation without script
- New form controls in browsers

<input>

- type=email:
- type=url:
- type=date:
- type=range: 0
- required:

New elements…

New elements for better document structure…

<section>

<nav>

<article>

<aside>

<header>

<footer>

`canvas` element: `img`, but scripted…

Used on Y! Pipes...

```
<canvas width="150" height="200" id="demo">
<!-- fallback content here -->
</canvas>

<script type="text/javascript">
 var canvas = document.getElementById("demo"),
     context = canvas.getContext("2d")
 context.fillStyle = "lime"
 context.fillRect(0, 0, 150, 200)
</script>
```

`video` and `audio` elements

- Extensive scripting API for loading and playing media resources

Along with new elements,
we also have new APIs

Cross-document messaging

- postMessage(): allows docs to communicate w/ each other across domains, without enabling cross-site scripting attacks
- implemented in Moz, Webkit, Opera, & IE8

New APIs: Another example

Persistent client-side data storage

- Client-side session and persistent storage of name/value pairs
- Gears-like client-side SQL database storage

Webkit implementation of HTML5 client-side SQL database API

- Example/demo is at http://webkit.org/misc/DatabaseExample.html
- Requires Safari 3.1 or latest nightly WebKit

Summary: How Will HTML5 help?

- Bring new features
- Make things better for Web developers
- Giving choice to users

Making things better for developers

- Precise, unambiguous spec makes interoperable implementation possible (less need for UA sniffing and multiple code paths)

- Enables (for example) Ajax to actually work the way it is supposed to — that is, interoperably
- Development of Web apps with features on par with Flash, Silverlight, and native applications

Giving choice to users

Standards-based Web applications work the same across browsers, so users are not locked into using any particular product from any particular vendor. Users get to **choose**.

How can you help HTML5?

- Participate in group (public-html mailing list)
- Something missing? Write a spec proposal!
- Volunteer as additional editor for (re)writing/editing a part of the spec

- Write and contribute test cases
- …

Questions? Comments?