HTML5 Standard and features

INF5750/9750 - Lecture 6 (Part I)

Problem Area

- The core language of the World Wide Web is HTML
- The WWW is accessed through a number of devices

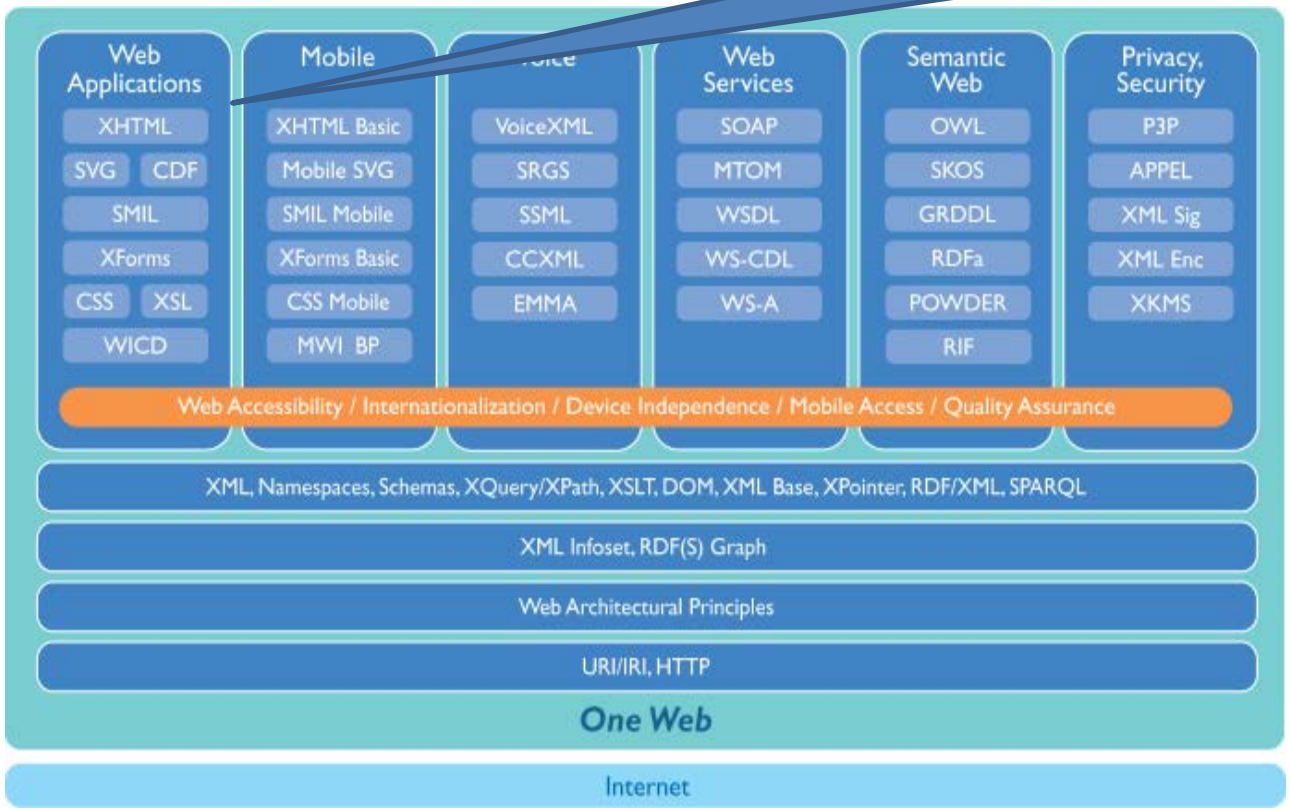
- Current HTML standard was designed for creating web pages that are accessed through browsers in computers with links to other web pages

- Today HTML is used to create Touch-screen websites, Games, 3D graphics, Audio-video conferencing, etc…

- Still we have to think whether our applications will work across devices

How to solve→ Standardization

- HTML is different from XHTML (WHATWG, 2004)
- WHATWG, a living HTML standard
- W3C uses version releases
(HTML5, 2014. HTML5.1 – 2016)



- Testing browser compatibility part of the standard to check adherence
- *Remember Acid2 and Acid3?*

HTML-syntax & XML-syntax

```
<!doctype html>
<html>
   <head>
      <meta charset="UTF-8“>
      <title>Example document</title>
   </head>
   <body>
      <p>Example paragraph</p>
   </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <title>Example document</title>
   </head>
   <body>
      <p>Example paragraph</p>
   </body>
</html>
```

Character Encoding & Doctype

```
<meta charset="UTF-8">
```

Instead of

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

No more DTDs. (earlier used for HTML quirks mode) and since HTML was SGML-based

```
<!DOCTYPE html>
```

Instead of

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

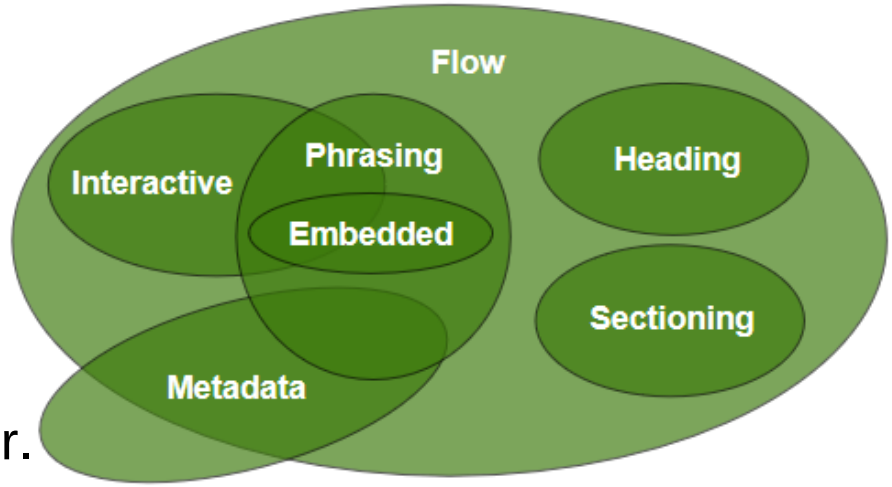
STRICT OR TRANSITIONAL

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

What’s different in HTML5

- 28+ new elements
- Many new attributes
- Many changed attributes
- Obsolete elements
- Obsolete attributes
- HTML5 does not use the terms "block-level" or "inline“
- New APIs
- Extensions

Content Model - Categories



- Metadata content, e.g. link, script.
- Flow content, e.g. span, div, text.

  This is roughly like HTML4's

  "block-level" and "inline" together.
- Sectioning content, e.g. aside, section.
- Heading content, e.g. h1.
- Phrasing content, e.g. span, img, text. This is roughly like HTML4's "inline". Elements that are phrasing content are also flow content.
- Embedded content, e.g. img, iframe, svg.
- Interactive content, e.g. a, button, label. Interactive content is not allowed to be nested.

New elements (commonly used ones)

- section – to define an application section. Used with h1, h2..
- main – only used once, to highlight main content
- header - a group of introductory or navigational aids
- footer – footer to a section, may contain author, copyright…
- nav – navigational part of the HTML
- figure/figcaption – self-contained flow content
- video/audio – multimedia content
- progress – to monitor progress of long processing tasks
- canvas – Used to render dynamic bitmap graphics
- input types – tel, search, url, email, datetime, month, week, time, datetime-local, range, color

See examples for new elements

- http://jsfiddle.net/sunbiz/4zJnE/
- http://jsfiddle.net/sunbiz/4JXkY/
- http://jsfiddle.net/sunbiz/KqymQ/
- http://sunnam.us/~saptarshi/
- http://jsfiddle.net/sunbiz/DqAF5/
- http://jsfiddle.net/sunbiz/xX5QB/
- http://www.senchalabs.org/philogl/demos.html

New attributes (commonly used ones)

- download – hyperlink to download a resources [a, area]
- charset – to specify character set [meta]
- autofocus – focused after page loads [input, select, button,…]
- placeholder – hint/tooltip shown [input, textarea]
- form – link element to form [input, select, fieldset, object…]
- required – data should be filled [input, select, textarea]
- many new attributes to input – autocomplete, min, max, …
- async – asynchronous loading [script]
- manifest – application cache manifest – offline data [html]
- sizes – different sizes icons [link: type=“icon”]
- crossorigin – getting image from another domain [img]

Obsolete elements

Better done through CSS:

- *basefont*, *big*, *center*, *font*, *strike*, *tt*

Removed because they are bad from usability

- *frame*, *frameset*, *noframes* - (DO NOT CONFUSE iframe)

Confusing or duplicate ones removed

- *acronym* is removed. Instead use *abbr* for abbreviations.
- *applet* removed. Instead use *object*.
- *isindex* usage can be replaced by usage of form controls.
- *dir* removed. Instead use *ul*.
- *noscript* only in HTML and not XML

New APIs

- *video*/*audio* have APIs for playback, with subtitles
- API for offline Web applications, using "*application cache*"
- Drag & drop API in combination with a *draggable* attribute
- *Location* interface exposing document's URL & navigation
- *History* interface exposing session history
- *atob()*, *btoa()* – Base64 encoding
- *setTimeout()*, *setInterval()* – timer based callbacks
- *postMessage()*, *MessageChannel* interface for cross-document and channel messaging
- *Worker*, *SharedWorker* interface for background scripts
- *localStorage*, *sessionStorage* – for offline data
- *WebSocket, EventSource* – client/server communication

API examples

- Video API - http://jsfiddle.net/sunbiz/P5DrC/4/
- Custom Validation - http://jsfiddle.net/sunbiz/uy9Ae/1/
- Drag Drop & Base64 - http://jsfiddle.net/sunbiz/b2FxK/2/
- Worker & MessageChannel - http://threadcomm.appspot.com/

Highly Innovative Changes

• Among the APIs, we can see some highly disruptive changes, that modify the way we've seen the WWW till date

- WebSocket
- LocalStorage, SessionStorage, IndexedDB **(Covered in Part II)**
- WebRTC

WebSockets

- WWW is largely based on so-called request/response paradigm of HTTP

- How does a server send information to client, when it wants to?? “Push data to client”

- HACK1: With long polling, the client opens an HTTP connection to the server which keeps it open until sending response

- Yuck2: Use Flash or Java applet and sockets from those

- Comes “Comet” and then standardized WebSocket to rescue!

Code snippets

```
// Defining a connection. Currently SOAP and XMPP are supported sub-protocol
var connection = new WebSocket('ws://folk.uio.no/~saptarsp/echo', ['soap', 'xmpp']);

// When the connection is open, send some data to the server
connection.onopen = function () {
    connection.send('Ping'); // Send the message 'Ping' to the server
};

// Sending String
connection.send('your message');

// Sending canvas ImageData as ArrayBuffer
var img = canvas_context.getImageData(0, 0, 400, 320);
var binary = new Uint8Array(img.data.length);
for (var i = 0; i < img.data.length; i++) {
    binary[i] = img.data[i];
}
connection.send(binary.buffer);

// Sending file as Blob
var file = document.querySelector('input[type="file"]').files[0];
connection.send(file);
```

WebSocket Demo

http://labs.dinahmoe.com/plink/

http://html5demos.com/web-socket

- Using WebSocket creates a whole new usage pattern for server side applications
- Such architectures are usually designed around either threading or so called non-blocking IO.

WebRTC

• **WebRTC is a new front in the long war for an open and unencumbered web.**

— Brendan Eich, inventor of JavaScript

• Imagine a world where your phone, TV and computer could all communicate on a common platform.

WebRTC implements three APIs:

• MediaStream (aka getUserMedia)

• RTCPeerConnection

• RTCDataChannel

• https://apprtc.appspot.com/

The War Goes On…

- On 2nd Oct, 2013, the W3C accepted proposal to include DRM as part of in HTML5
- The purpose of DRM is to give content providers leverage against creators of playback devices. (W3C, Ian Hickson)

- Mozilla's bug report - Pledge never to implement HTML5 DRM - https://bugzilla.mozilla.org/show_bug.cgi?id=923590

Resources

- W3C HTML5 Candidate Release
  - http://www.w3.org/html/wg/drafts/html/CR/
- WHATWG HTML Living Standard
  - http://www.whatwg.org/specs/web-apps/current-work/multipage/
- Discussion around HTML5 DRM
  - https://plus.google.com/107429617152575897589/posts/iPmatxBYuj2
  - http://www.theguardian.com/technology/blog/2013/mar/12/tim-berners-lee-drm-cory-doctorow