HTML5 Fly-by

HTML History

What's New

What's Next

HTML History

"Hypertext" concept has been around since at least the 40s

- I can still recall Mac Hypercards
- See http://goo.gl/rifuL2

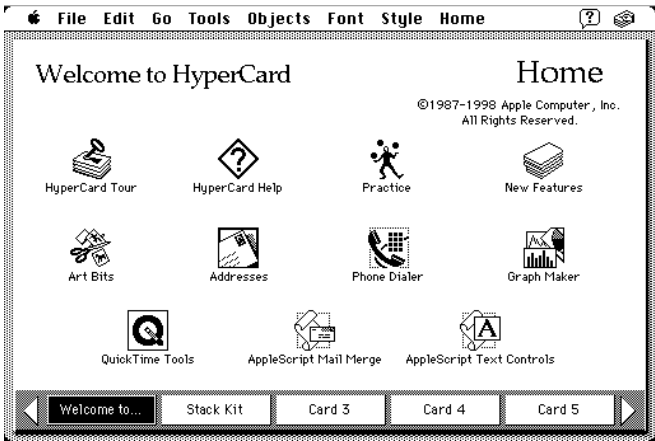
*Hyper*Text *M*arkup *L*anguage

- Sir Tim Berners-Lee "hacked" it together in 1989-90.
- Based it on SGML (Standard General Markup Language)
- Dave Ragget (HP) and Berners-Lee enhanced to HTML+ (92-94)
- Andressen created the IMG tag in Mosaic (the others didn't like it!)

Fast-forward:

- HTML 2.0 evolved ad hoc based on HTML+ (as did browsers)
- Netscape formed in Nov. 1994
- IETF on HTML formed in Sept. 1994
- W3C forms in late 1994

Competing forces on HTML standards



HTML History (cont.)

HTML3 – 1st "real" version of the standard (3.2 1/97)

- IETF draft 3/95, coalesced a number of offshoot features
- IE 1.0 (8/95) joins Netscape as major commercial browser
- Vendors moved faster than standards and pushed extensions
  - Forced the IETF group to disband
- CSS and "scripting" was still by informal agreement at best

Fast-forward #2: HTML 4.01 (May 2000)

- Adopted vendor extensions, including frame madness
- Deprecated formatting elements in favor of CSS
- Introduced mouse and keyboard events on most elements
- Largely stable revision until 2008 (first HTML5 published 1/08)
- But, XML came out in 1997, and folks like how clean it was

XHTML (epic fail): Trying to put the genie back in the bottle

- Tried to force well-formedness and semantic markup
- The WHATwg formed in 2004 to again promote practical considerations for evolving HTML over the strictness of XHTML

HTML 3.2 to HTML 4.0

These were the predominant versions for a long time.

Text Mixed with Markup Tags

- Tags Enclosed in Angle Brackets (<H1>Introduction</H1>)

What Does Markup Describe?

- Appearance
- Layout
- Content (Can't Enforce an Exact Look)

Changes in HTML 3.2 to HTML 4.0

- Standardization of frames
- Deprecation of formatting elements (vs. style sheets)
- Improved cell alignment and grouping in tables
- Mouse and keyboard events for nearly all elements
- Internationalization features

Hypertext Links

Links can contain images and other text-level elements (i.e., <A HREF…> ... </A>)

- Link to Absolute URL
  - Use a complete URL beginning with http://

```
Java discussed in
<A HREF=http://host/path/ch2.html>Chapter 2</A>.
```

- Link to Relative URL
  - Use a filename or relative path to filename
    - Interpreted wrt location of current file

```
Java discussed in <A HREF="ch2.html">Chapter 2</A>.
```

- Link to a Section of a URL
  - This traditionally was done using the # syntax, e.g.

```
Images discussed in <A HREF="ch1.html#Sec2”>Sec. 2 of Chap. 1</A>
```

  - But it is not done as much anymore

HTML5

The work of the WHATwg won out over the W3C backed work on XHTML

- *Diveintohtml5* suggests this is because it was error-forgiving.

So what is in HTML5?

- New semantic tags
- Updates to Forms
- New object types – video, audio, canvas, etc.
- Includes CSS and Javascript properties
- Updates to existing presentation tags (not included)

How do you know if your browser supports it?

- You can write Javascript to detect browser features
- You can use shims and polyfills (e.g. Modrnzr)

HTML5: Semantic Tags

From a markup perspective, it supports a set of semantic tags

- <main> - denotes the main content of the page
- <section>
- <article>
- <nav>
- <details> - Additional details user can view or hide
- <summary> - creates a visible header for the details
- <mark> - defines marked, or highlighted text
- <time> - Defines a date/time
- <aside> - sidebar-style content
- <header> - header content for a document or section
- <footer> - footer content for a document or section
- <figure> and <figcaption> - self-contained content

See
http://www.w3schools.com/html/html5_semantic_elements.asp

Food for (future) thought:
How does semantic tagging here compare with semantic approach of XML?

HTML4 vs. HTML5

Structure of a Document

- The new tags do not impact presentation
- They were created to form more expressive document *structure*

- HTML4
  - Document structure done using tags
    - Creates a block structure to which you can apply formatting
    - Largely used for DOM manipulation or CSS application
      - *style*, *class*, and *id* attributes for CSS application and DOM location
      - *Think of div like curly braces {} in a program*
  - There is also the tag
    - Allows you to style the enclosed element but does not create structure
    - Unlike , will not insert newlines in rendered content
- HTML5
  - Tags like <section>,<article>, are intended to block structure the document semantically – the tagname describes the block type
  - No change to rendering at present, still have to apply CSS!

Digression: HTML Forms

```
<HTML>
<HEAD><TITLE>A Sample Form Using GET</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using GET</H2>

<FORM ACTION="http://localhost:8088/SomeProgram">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT"> <!-- Press this to submit form -->
  </CENTER>
</FORM>
</BODY>
</HTML>
```

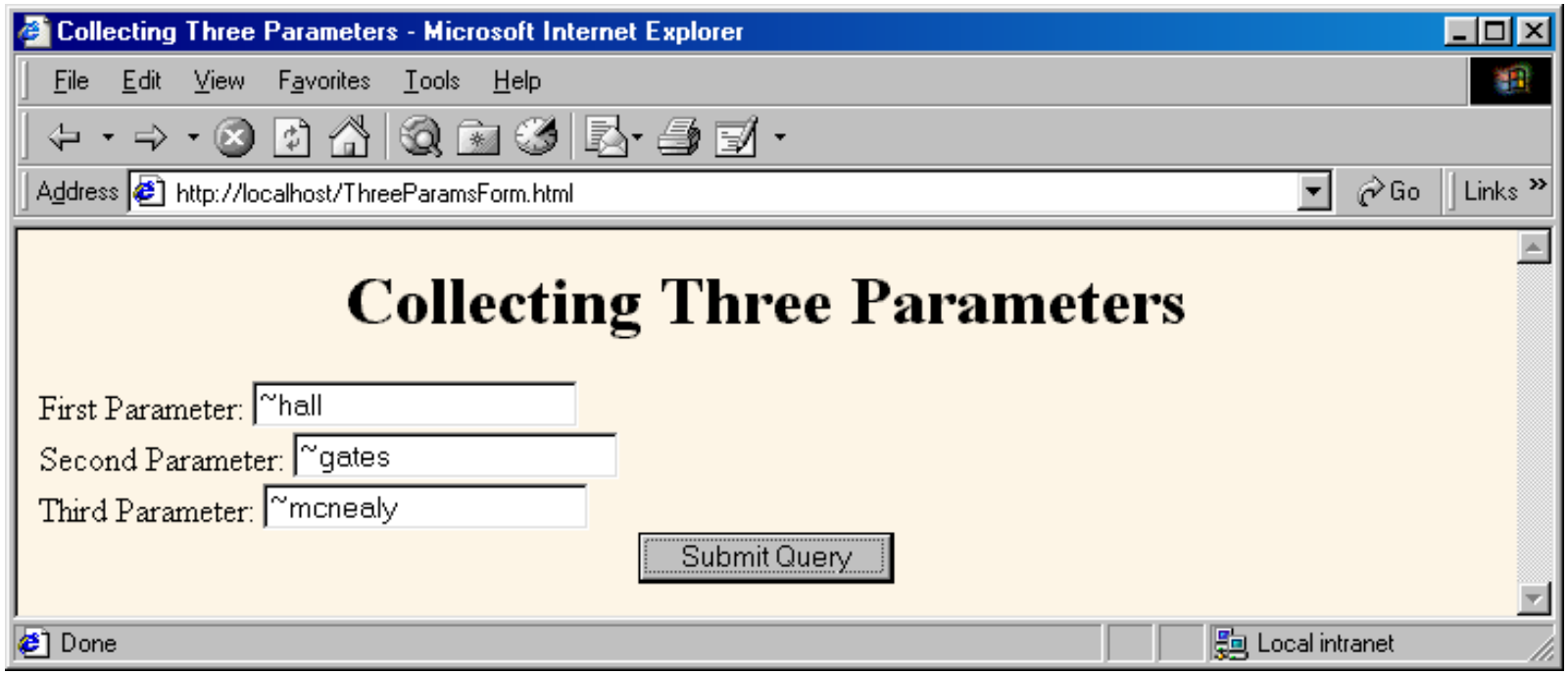
Sending POST Data

```
<HTML>
<HEAD><TITLE>A Sample Form Using POST</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using POST</H2>

<FORM ACTION=http://localhost:8088/SomeProgram METHOD="POST">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT">
  </CENTER>
</FORM>
</BODY></HTML>
```

An HTML Form With Three Parameters

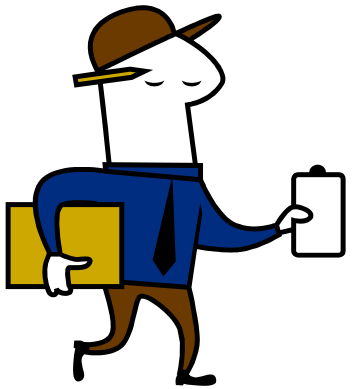
```
<FORM ACTION="/servlet/coreservlets.ThreeParams">
  First Parameter:  <INPUT TYPE="TEXT" NAME="param1"><BR>
  Second Parameter: <INPUT TYPE="TEXT" NAME="param2"><BR>
  Third Parameter:  <INPUT TYPE="TEXT" NAME="param3"><BR>
  <CENTER><INPUT TYPE="SUBMIT"></CENTER>
</FORM>
```



HTML5 (cont.)

HTML5 includes support for additional form inputs:

1. <input type="search"> for search boxes
2. <input type="number"> for spinboxes
3. <input type="range"> for sliders
4. <input type="color"> for color pickers
5. <input type="tel"> for telephone numbers
6. <input type="url"> for web addresses
7. <input type="email"> for email addresses
8. <input type="date"> for calendar date pickers
9. <input type="month"> for months
10. <input type="week"> for weeks
11. <input type="time"> for timestamps
12. <input type="datetime"> for precise, absolute date+time stamps
13. <input type="datetime-local"> for local dates and times



… and form elements:

1. <datalist> - predefined options on an input list (no more “select”)
2. <keygen> - generates public/private keys for authentication (stay tuned)
3. <output> - result of a calculation performed by a script

New HTML5 Media Object Types

New media object types provide standard tagging for objects now popular to deliver on the Web:

- <video src="foo.ext#t=5,10" type='video/ext;codecs="WebM,MP4"'/>
- <audio>
  - video and audio support a <source> nested element that presents an ordered list for the browser to choose from for compatibility.
  - After that list of <source> tags is a fallback content object if no match

- <canvas> - creates a container area on the page that Javascript can then get a reference to, and execute drawing commands
  - Lot of stuff you can do in a canvas, see http://goo.gl/F1MTwW
- <svg> - Scalable Vector Graphics
  - Also supports various drawing primitives
  - Vector graphics can be panned/zoomed without distortion (unlike raster) due to the way they are defined (XML graphics primitives)
- Note – you can still use <object> and <embed> if you like to add untyped object extensions you expect the browser to handle via a plug-in of some sort.

CSS and Javascript

HTML5 now “includes” CSS and Javascript

- Actually CSS has been there a long time
- Javascript has too (see History), but in non-standard ways
- But now there are a number of new features
  - Geolocation, Drag ‘n Drop, Storage, App Cache, Web Workers, & SSEs
  - We will revisit these after we do some Javascript
- Their inclusion attempts to address the abundance of shims

- What are *shims* and *polyfills*?
  - Short answer: hacks
  - Longer answers:
    - Shims are a think compatibility adapter (Go4 Adapter/Façade)
    - Polyfills augment browsers for missing features (Go4 Decorator)
  - While necessary, the emergence and application is ad hoc

- How do you know if a browser supports your new HTML5?
  - You can ask the browser using Javascript
  - Or, you can use Modernizr (or others) to check for you

HTML5 Summary

HTML5 is the result of a long effort to evolve browsers with richer features

Adoption delayed by outside forces:

1. The fractioning of the community between WHATwg and XHTML
2. The sense that the browser wars were “over”
3. The explosion of content meant (and still means) people do not have huge appetite for rewriting their content and rendering systems again

And yet more forces that represented constant change

1. The rate of change in mobile technologies
   - You folks ever hear of MIDP, J2ME, or WAP?
   - Processing power, RAM, display capabilities, and greater bandwidth has made mobile rendering a hard landscape to keep up on.
2. The rate of change in gaming and social media
   - Again, capabilities of the devices make web-based gaming real
   - Younger users, BSA, where does it end?
3. And we didn’t even mention WebComponents and ShadowDOM!

It doesn’t end – your phone as an always-on device, your universal experience, The privacy of your data on the cloud, and ubiquitous sensors (IoT)…