

HTML ~~& CSS~~

CSC 337, Fall 2013

William H. Mitchell

Introduction

Big picture

HTML – Hypertext Markup Language
Specifies structure and meaning

CSS – Cascading Style Sheets
Specifies presentation

JavaScript
Specifies behavior

PHP – PHP: Hypertext Preprocessor
One of many back-end options

A history of hypertext and HTML

1941: Jorge Luis Borges writes *The Garden of Forking Paths*.

1945: Vannevar Bush describes Memex in *As We May Think*.

1960: Ted Nelson founds Project Xanadu and (1963?) coins term "hypertext".

1967: Brown U's Hypertext Editing System; used for Apollo.

1969: Charles Goldfarb, Ed Mosher, and Ray Lorie at IBM create the Generalized Markup Language. Leads to SGML.

1987: HyperCard, by Apple

History, continued

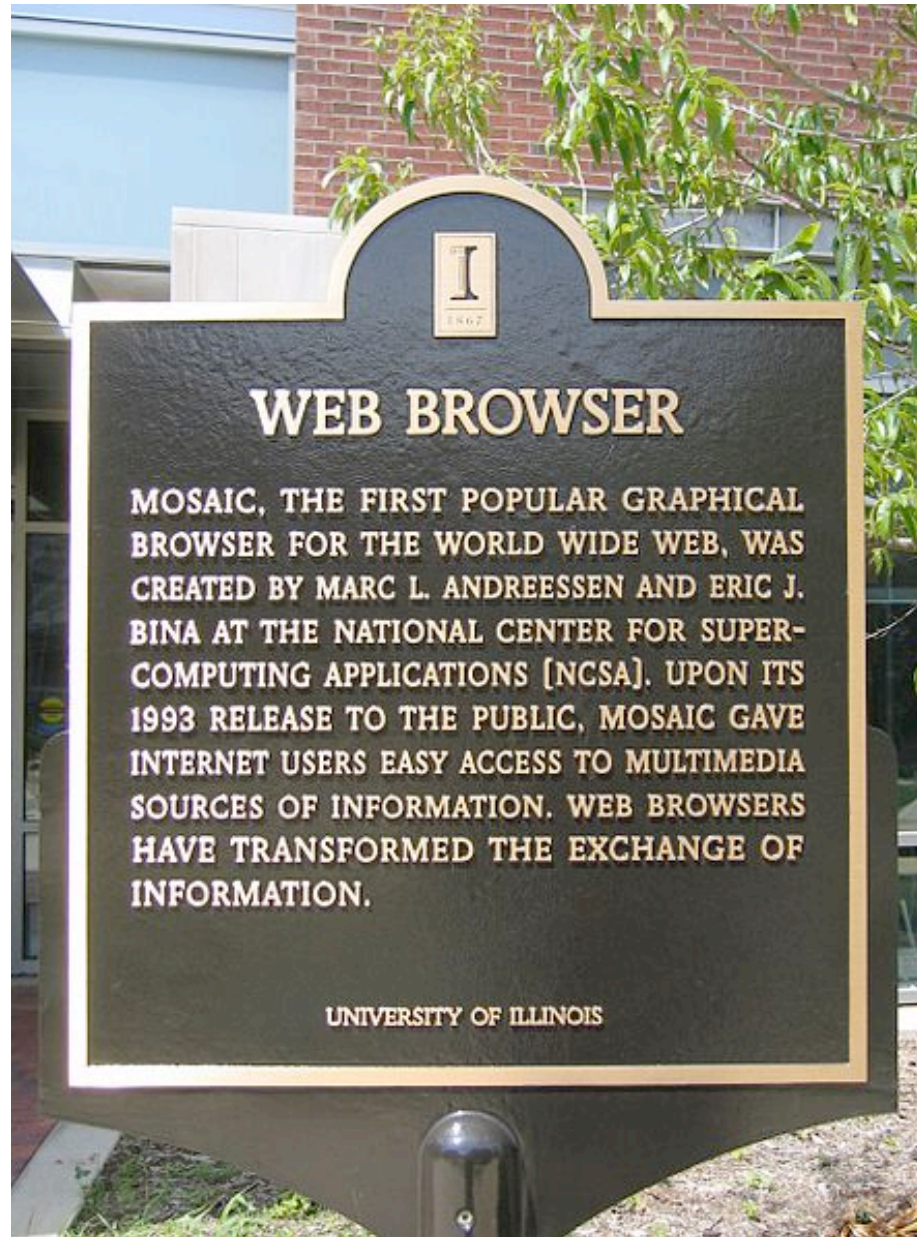
1990: Tim Berners-Lee and Robert Cailliau at CERN write *WorldWideWeb: Proposal for a HyperText Project*.

1990: Berners-Lee creates first browser: WorldWebWeb, later renamed Nexus, on a NeXT computer.

1991: *HTML Tags* by Berners-Lee. Influenced by SGML.

1993: Marc Andreessen and Eric Bina at NCSA create Mosaic. First developed with X11 then released for Windows and Mac. First browser to show images in-line.

1994: Mosaic Communications Corporation founded by Jim Clark and Marc Andreessen. Later renamed Netscape.



WEB BROWSER

MOSAIC, THE FIRST POPULAR GRAPHICAL BROWSER FOR THE WORLD WIDE WEB, WAS CREATED BY MARC L. ANDREESSEN AND ERIC J. BINA AT THE NATIONAL CENTER FOR SUPER-COMPUTING APPLICATIONS [NCSA]. UPON ITS 1993 RELEASE TO THE PUBLIC, MOSAIC GAVE INTERNET USERS EASY ACCESS TO MULTIMEDIA SOURCES OF INFORMATION. WEB BROWSERS HAVE TRANSFORMED THE EXCHANGE OF INFORMATION.

UNIVERSITY OF ILLINOIS

History, continued

1994: World Wide Web Consortium founded.

1995: HTML 2.0 published by Internet Engineering Task Force (RFC 1866)

1995ish: Browser wars begin—Internet Explorer vs. Netscape Navigator (others, too)

January 1997: HTML 3.2 published as *W3C Recommendation*. Incorporates various proprietary extensions, including visual markup.

History, continued

December 1997: HTML 4.0 published by W3C.
Deprecates visual markup in favor of style sheets.

1998: XML 1.0 Recommendation by W3C

1999: HTML 4.01 published

2000: XHTML 1.0 published

2004: Web Hypertext Application Technology
Working Group (WHATWG) begins work on HTML5

2008: First W3C *Working Draft* of HTML5

HTML5

In practice, "HTML5" has two meanings:

- The fifth major version of this markup language

- An umbrella for many web technologies

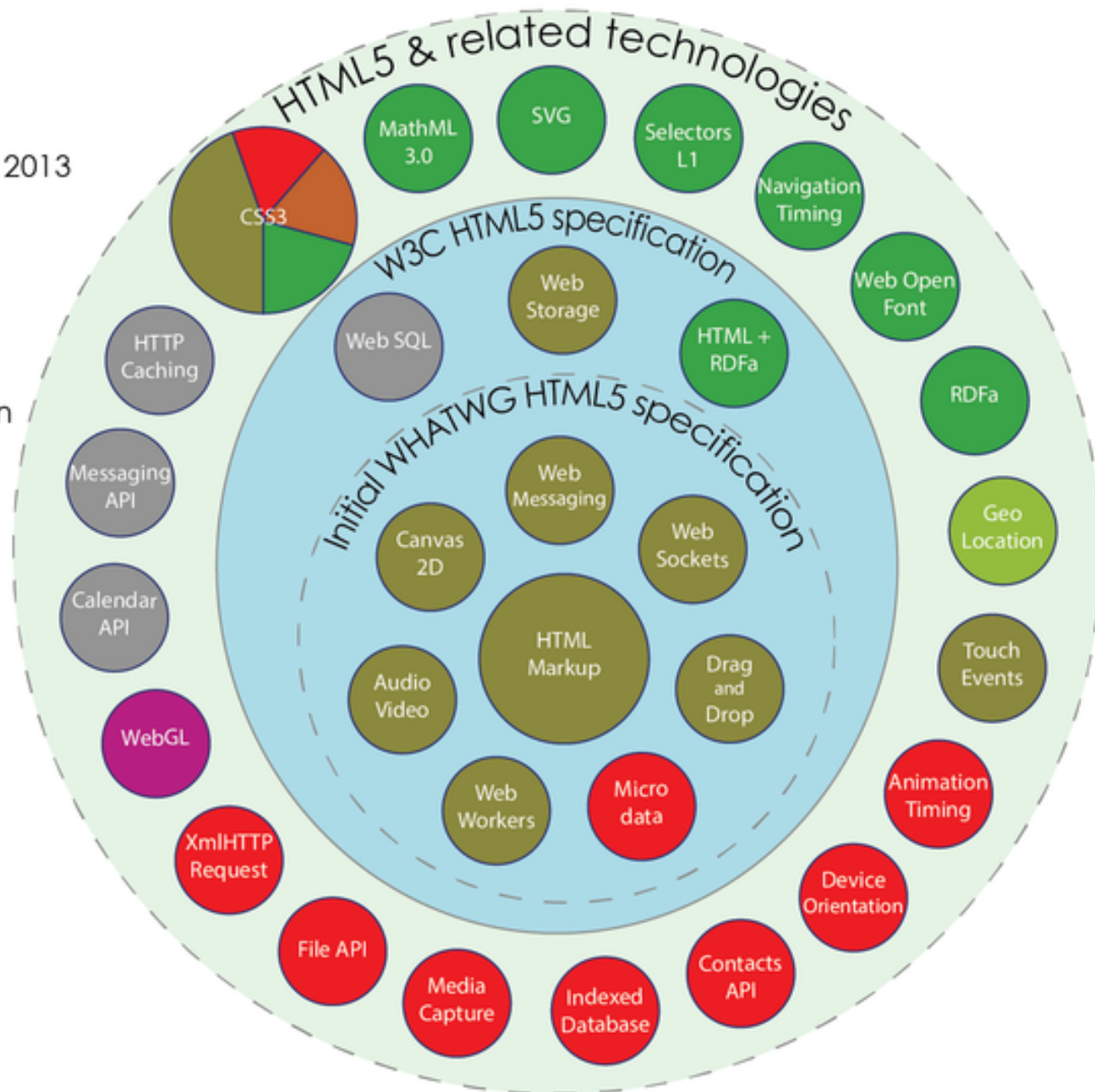
Current W3C status for the markup language is "Candidate Recommendation".

Current goal for reaching "W3C Recommendation", i.e., a standard, is 2014.

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody (CC) BY · SA

<http://en.wikipedia.org/wiki/File:HTML5-APIs-and-related-technologies-by-Sergey-Mavrody.png>

"HTML" for us

We'll consider "HTML" to mean the markup language specified by the HTML5 Candidate Recommendation, www.w3.org/TR/html5/

It's largely HTML 4.01 plus a number of new elements for *semantic markup* and media support.

Resources for HTML

Head First HTML & CSS, 2nd edition by Elisabeth Robson and Eric Freeman. On Safari.

Mozilla Developer Network:

developer.mozilla.org

W3C:

www.w3.org/TR/html5/

Web Hypertext Application Technology Working Group:

whatwg.org

HTML Basics

HTML specifies structure

Here is some text marked-up with HTML:

```
<h1>Introduction</h1>
```

```
<p>
```

```
A paragraph...
```

```
</p>
```

```
<ul>
```

```
  <li>First item</li>
```

```
  <li>Second item</li>
```

```
</ul>
```

```
<p>
```

```
Second paragraph...</p>
```

What's structure good for?

We can format text and apply styles based on types and structure.

- First-level headings

- An image in a footer

- Paragraphs in a particular class of lists of lists

Programs analyzing pages can derive meaning.

- An article

- Headers, rows, and columns in table

- Ordered vs. unordered lists

Element anatomy

This is a title *element*:

```
<title>My page</title>
```

- "title" is the *tag name*
- <title> is the *opening tag*
- </title> is the *closing tag*
- The element's *content* is "My page"

Sometimes we'll write "<a>" as a shorthand for "the 'a' element".

Omitting closing tags

Closing tags can be omitted in a few common cases.

```
<h1>Introduction</h1>
```

```
<p>
```

A paragraph...

```
<ul>
```

```
  <li>First item
```

```
  <li>Second item
```

```
</ul>
```

```
<p>
```

Second paragraph...

<http://www.w3.org/TR/html5/syntax.html#optional-tags>

Attributes

Attributes for an element can be specified in the opening tag:

```
<p id=intro class=simple>
```

id, class, title, and others are *global attributes*—they can be specified for any element.

Some attributes are element-specific.

```
<img src='pic1.png' title="Fluffy's Birthday">
```

Quotes can be ' or ". They are optional, with rules.

<http://www.w3.org/TR/html-markup/syntax.html#syntax-attributes>

<http://www.w3.org/TR/html5/dom.html#global-attributes>

Void elements

Void elements never have content.

Four examples:

```
<br>
```

```
<hr>
```

```

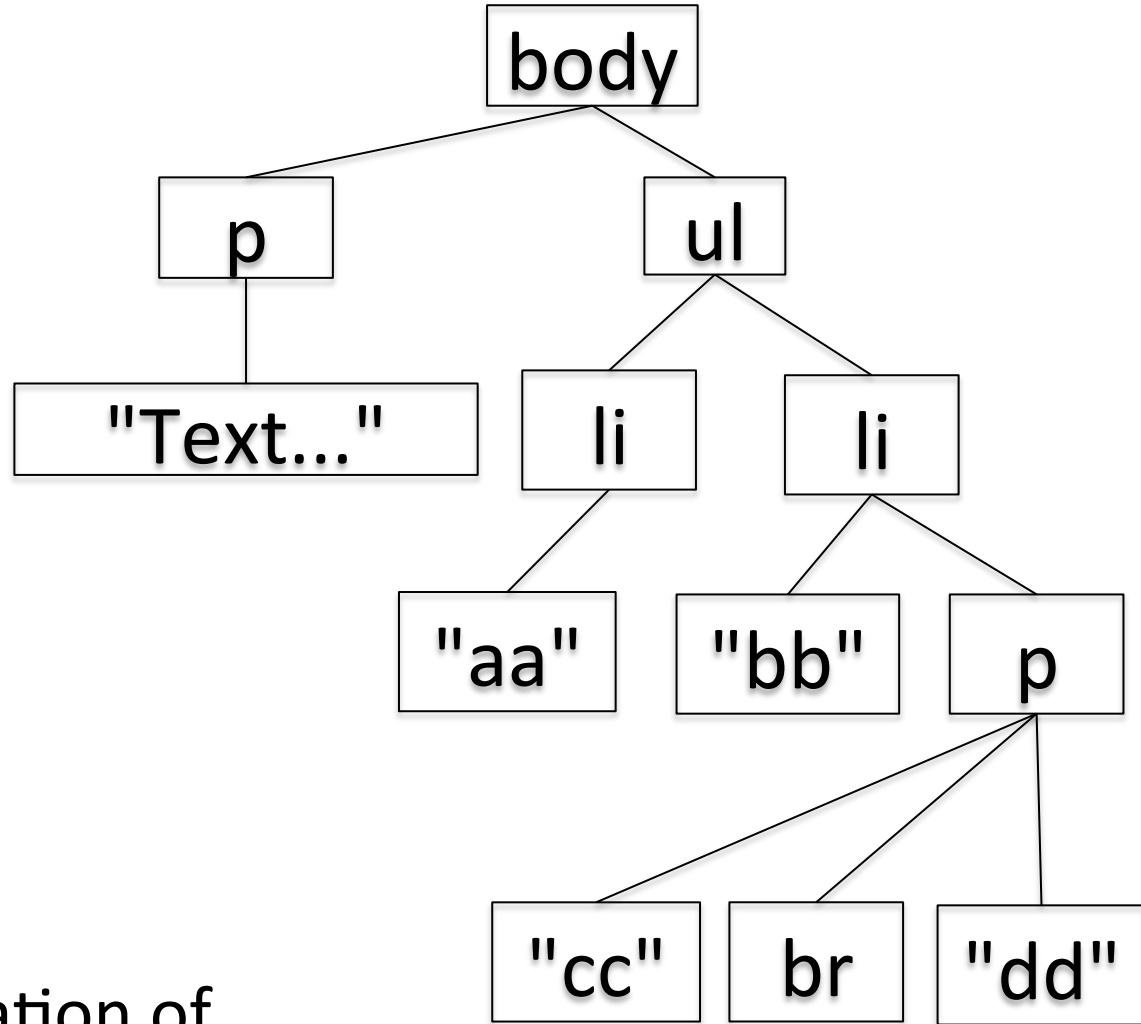
```

```
<input type="button" value="Test">
```

Void elements do not have closing tags.

Nesting and tree structure (corrected!)

```
<body>
<p>
Text...
<ul>
<li>aa
<li>bb
<p>
cc<br>dd
</ul>
</body>
```



Note the combination of elements and *text nodes*.

Nesting and tree structure

Tree structure cannot be forced.

An element's *content model* specifies what the element can have as children.

Based on the previous diagram, what can we say about the content model for the p, ul, and li elements?

Documentation on the content model for ul:

<http://www.w3.org/TR/html5/grouping-content.html#the-ul-element>

Lexical details

Tag names and attributes are case-insensitive.

Comments are `<!-- . . . -->` and do not nest.

Text nodes and attribute values must be Unicode characters \geq U+0020 (space) except for tab, newline, formfeed, and carriage-return.

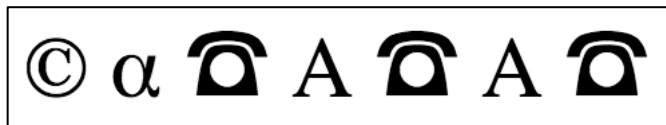
Character references

Character references can be used to escape characters. There are three forms:

&name; © α ☎

&#decimal-digits; A ☎

&#xHex-digits; A ☎



<http://www.w3.org/TR/html5/syntax.html#named-character-references>

Character references

The characters & and < can appear in some places but not others.

```
<p title="<">
```

OK!

```
c<d
```

Trouble...

```
a&b
```

Trouble!

Solution: Use & and <

```
c&lt;d a&amp;b
```

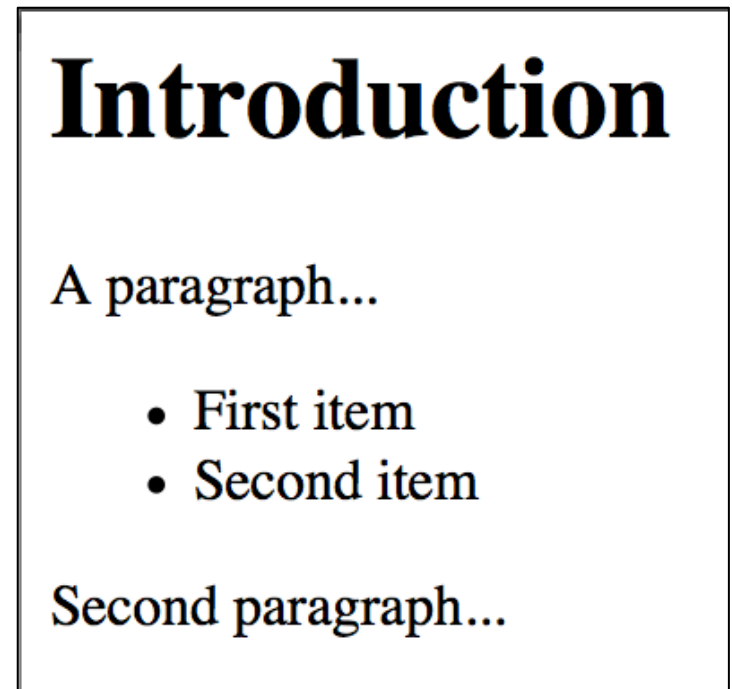
What's this?

```
title="&quot;&apos;1&amp;2&apos;&quot;"
```


HTML rendering

Browsers produce a rendering of HTML.

```
<h1>Introduction</h1>
<p>
A paragraph...
<ul>
  <li>First item
  <li>Second item
</ul>
<p>
Second paragraph...
```



Layout engines

A browser's *layout engine* is the component that displays HTML content, taking all markup and styling into account.

Browsers have various engines:

Firefox – Gecko

IE – Trident

Konqueror – KHTML (from KDE)

Safari – WebKit (based on KHTML)

Chrome and Opera – Blink, fork of WebKit

[Usage share](#) of web browsers

Demo: Chrome Developer view

Chrome: View>Developer>Developer Tools

Learn the keyboard shortcut, which toggles it

Explore element structure—expand/collapse with mouse or arrow keys

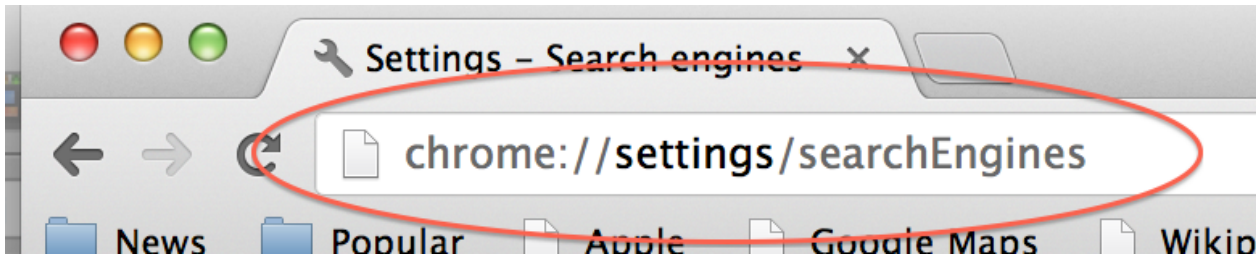
Right-click, Inspect Element is another way to bring up the Developer Tools view

Breadcrumbs at bottom show path to element.

We'll see later that the blue/orange/green areas are the content/margin/padding in the box model.

hox1.html

Quick access to docs



Other search engines

Chrome Search Engines	se	chrome://settings/searchEngines
MDN HTML Element	mdnh	https://developer.mozilla.org/en-US/docs/...
W3C HTML5	html5	http://www.w3.org/TR/html5/

Add a new search en

Keyword

URL with %s in place

developer.mozilla.org/en-US/docs/Web/HTML/Element/%s

To go directly to the MDN doc for the "img" tag, type Ctrl-L, then "mdnh img"

Boilerplate

Boilerplate

Browsers render our examples thus far but common practice is to include more structure:

```
<!doctype html>  
<html>  
  <head>...</head>  
  <body>...</body>  
</html>
```

<!doctype ...>

"<!doctype ...>" indicates which set of specifications the browser is to use for this document.

HTML 4.01 Transitional:

```
<!doctype html public "-//w3c//DTD html 4.01  
transitional//EN"> (minimal caps)
```

HTML5:

```
<!doctype html>
```

Space characters and comments can precede
<!doctype ...>

<head>

The <head> element contains *metadata* elements.

Most common are <title>, <style>, and <script>

<title> is the only required element in <head>

<http://www.w3.org/TR/html5/dom.html#metadata-content>

<meta>

The meta element is a catch-all for metadata that doesn't have its own element.

A common use is to specify the *character encoding*:

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

With the charset attribute, meta can only appear in the head element.

<http://www.joelonsoftware.com/articles/Unicode.html>

<body>

"The <body> element represents the content of the document." (W3C)

One way to think about it: The <body> content is what you see.

Only one body element is permitted.

All together

```
<!doctype html>  
<html>  
  <head>  
    <title>Greetings</title>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    Hello, world!  
  </body>  
</html>
```

Error Handling and Validation

Hands-on: Error handling

"Authors must not use HTML elements anywhere except where they are explicitly allowed, as defined for each element, or as explicitly required by other specifications."

--W3C

Let's examine some errors...

Error handling

What other software is equally permissive?

Why are browsers so permissive in what they accept?

What risks are posed by permissiveness?

Speculate: Are browsers equally flexible with CSS? JavaScript?

Hands-on: Markup validation

validator.w3.org provides validation of markup

Let's try it!

What's the minimal valid HTML5 document?

<http://www.w3.org/TR/html5/syntax.html#optional-tags>

More elements

strong, em, mark

`` "gives text strong importance"

`` "marks text that has stress emphasis"

`<mark>` represents "a run of text marked for reference purpose, due to its relevance in a particular context"

strong, em, mark

I strongly
emphasize that
we should
<mark>mark</mark> this!

<mark>
I strongly emphasize that
we should mark this!
</mark>

I strongly
emphasize that
we should
mark this!

I strongly
emphasize
that we should
mark this!

Nesting

Improperly nested tags won't validate, but typically work.

This is `strong and emphasized` text.

This is *strong and emphasized* text.

q for quotations

The q element is used for short quotations. The content is displayed with quotation marks.

```
<q>Life is hard. It's harder if you're stupid.  
</q><br>&mdash;John Wayne
```

"Life is hard. It's harder if you're stupid."
—John Wayne

The cite attribute can be used to cite the source.

blockquote

Use blockquote for longer quotations.

Theodore Roosevelt said,

`<blockquote>`

Far better is it to dare mighty things, to win glorious triumphs, even though checkered by ...

`</blockquote>`

Theodore Roosevelt said,

Far better is it to dare mighty things, to win glorious triumphs, even though checkered by failure, than to take rank with those poor spirits who neither enjoy much nor suffer much because they live in the gray twilight that knows neither victory nor defeat.

Sidebar: Structure vs. appearance

Elements like h1, p, ul, and li are clear examples of markup that describe structure/meaning.

Are q and blockquote describing structure/meaning or appearance? How about em and strong?

Would elements that specify bold, italics, and underlining be describing structure or appearance?

b, i, and u elements

"The b element represents a span of text to which attention is being drawn for utilitarian purposes without conveying any extra importance and with no implication of an alternate voice or mood, such as key words in a document abstract, product names in a review, actionable words in interactive text-driven software, or an article lede."

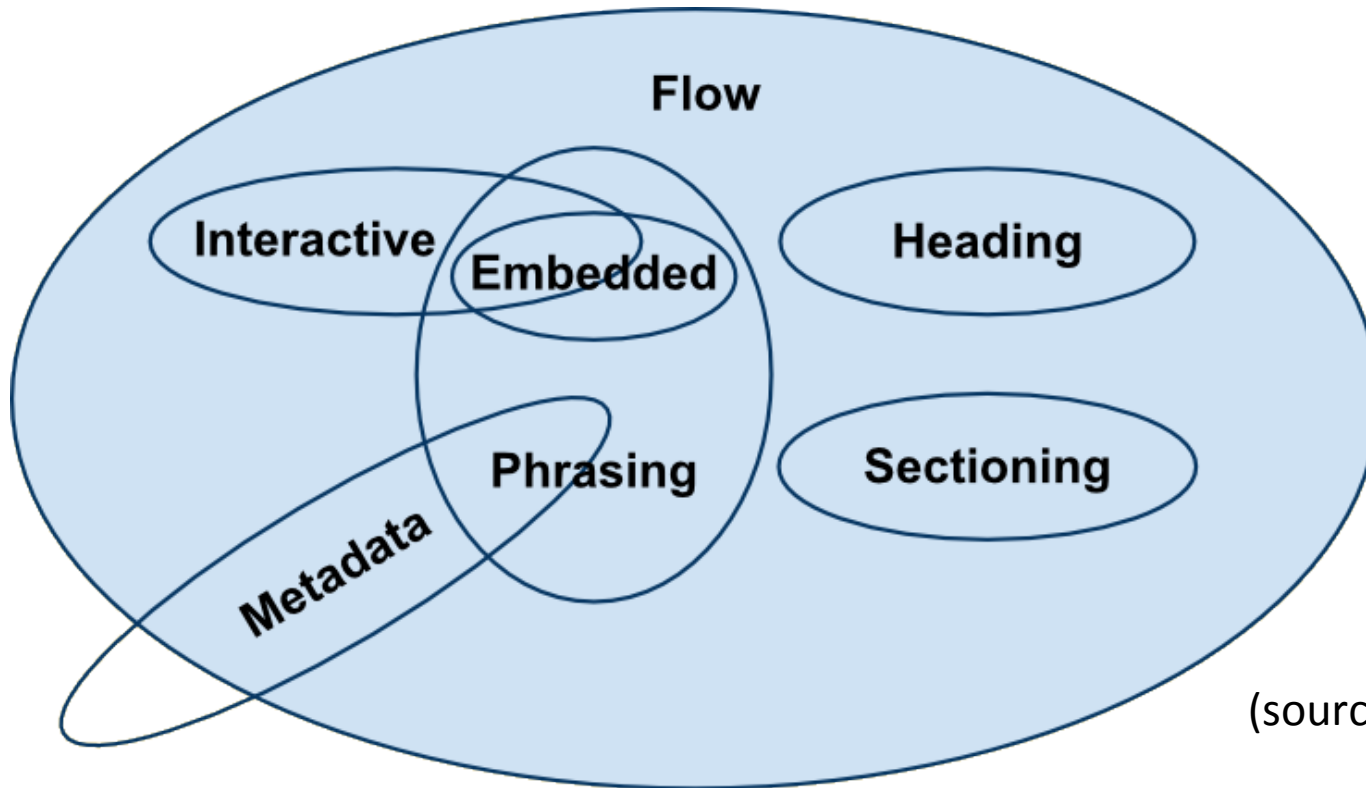
"The i element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose in a manner indicating a different quality of text, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, transliteration, a thought, or a ship name in Western texts."

"The u element represents a span of text with an unarticulated, though explicitly rendered, non-textual annotation..."

Content Categories

Content categories

Elements are in zero or more *content categories*.



(source: MDN)

Animated diagram:

<http://www.w3.org/TR/html5/dom.html#kinds-of-content>

Content categories

An element's specification includes *categories*:

4.5.1 The p element

Categories:

Flow content.

Palpable content.

Content model:

Phrasing content.

4.6.20 The mark element

Categories:

Flow content.

Phrasing content.

Palpable content.

Content model:

Phrasing content.

Can a <p> contain a <mark>?

Can a <mark> contain a <p>?

Can a <p> contain a <p>?

Flow content

"Most elements that are used in the body of documents and applications are categorized as *flow content*." (W3C)

Examples:

article, audio, br, code, em, embed, form, hN, header, iframe, img, input, math, ol, p, section, svg, video

Also, text is considered to be flow content.

Phrasing content

"Phrasing content is the text of the document, as well as elements that mark up that text at the intra-paragraph level. Runs of phrasing content form paragraphs."--W3C

What elements have we seen that are phrasing content?

<http://www.w3.org/TR/html5/dom.html#phrasing-content>

Embedded content

"Embedded content is content that imports another resource into the document, or content from another vocabulary that is inserted into the document." (W3C)

Examples:

audio, embed, iframe, img, math, svg, video

math and svg are examples are examples of content from another vocabulary.

Element display

Element display

Two common types of element display behavior are *block* and *inline*.

Block display uses all available width and causes line breaks. `<p>` uses block display.

Inline display uses only as much width as needed and does not cause line breaks. All (most?) elements that are phrasing content use inline display, including `img`.

Hands-on with element display

Adjust width and observe how layout adjusts.
What's the algorithm?

Experiment and classify:
ul, li, h1, strong

We'll use some CSS rules to explore and understand behavior but we won't get caught up in CSS yet!

blockinline.html

Odds and Ends

Details on hN elements

There are six levels, h1 through h6, from most to least important.

Default rendering produces a range of sizes but do NOT use levels for sizes. (Use styles instead!)

Software agents might use headings to construct a table of contents for a page. The W3C validator has an option to produce an outline.

Details on lists

There are three list elements:

ol Ordered list

ul Unordered list

dl Description list (look it up!)

Ordered lists

The ordered list (ol) element is used for lists where the sequence is important.

```
<ol>  
  <li>Place order at counter  
  <li>Pay for order  
  <li>Wait for order  
  <li>Eat it!  
</ol>
```

1. Place order at counter
2. Pay for order
3. Wait for order
4. Eat it!

Only li elements can be children of ol.
Note that is not needed

Unordered lists

The unordered list (ul) element is used for lists where the sequence is not important.

```
<ul>  
  <li>Sunscreen  
  <li>Sunglasses  
  <li>Beach chair  
  <li>Book  
</ul>
```

- Sunscreen
- Sunglasses
- Beach chair
- Book

Only li elements can be children.

Problem: ordered bullets

Should we use a ul element to show a sequence of steps if we don't want the steps numbered?

- Place order at counter
- Pay for order
- Wait for order
- Eat it!

<pre>

The <pre> element is used for preformatted text. Whitespace is preserved, and the text is shown in a fixed-width font.

```
<pre>
  | \
  / , ` . - ' ` ' _ ' ' ' ' - - - ' ' ' _
  | , 4 - ) ) - , _ . . ; \ ( ` ' - '
' - - - ' ' ( _ / - - ' ` - ' \ _ )
                                Felix Lee
```

```
</pre>
```



Hyperlinks

The anchor (a) element

The "hyper" in HyperText refers to the ability to jump from one point in text to another.

Hyperlinks are defined with anchor elements:

Here's the `fine print`.

Here's the [fine print](#).

<a>

At hand:

Here's the `fine print`.

Clicking the link causes the browser to open the HTML document in the file `fp.html` that's in the same directory as the current document.

Relative paths to files can be used:

`First step`

`Glossary`

<a>

By default, clicking a link opens the new document in the current view.

The target attribute can direct that a new view be created:

```
<a href="fp.html" target="_blank">
```

The default target is "_self". There are other targets, too.

hrefs with fragments

An href can reference a *fragment*—a name preceded by a hash mark:

...the `grading rubric ...`

The hyperlink will jump to the element on this page with a matching id attribute:

```
<h2 id="rubric">
```

```
<p>
```

```
The grading rubric....
```

Problem: glossary page

[a](#) [b](#) [c](#) [d](#) [e](#) [f](#) [g](#) [h](#) [i](#) [j](#) [k](#) [l](#) [m](#) [n](#) [o](#) [p](#) [q](#) [r](#) [s](#) [t](#) [u](#) [v](#) [w](#) [x](#) [y](#) [z](#)

[a](#)

- accelerate
- adolescent
- adulthood

[b](#)

- bamboo
- blend
- blubber

hagloss1.html

URL: Uniform Resource Locator

A URL specifies where something can be found and how it should be accessed.

Here's a URL for the glossary page from the previous slide:
`file:///w/337/s/hagloss1.html`

Its *scheme* is "file" (some call it the *protocol*)

Its *path* is `/w/337/s/hagloss1.html`

The URL above has no *host*, but this is equivalent:

<file://localhost/w/337/s/hagloss1.html>

On Windows I might have used this, assuming a subst for w:
`file:///w|/337/s/hagloss1.html`

An anchor's href specifies a URL

The anchor element's href attribute must specify a URL, but in anchor1.html we used this:

```
href="fp.html"
```

That's considered to be a relative URL.

The absolute URL for anchor1.html is
`file:///w/337/s/anchor1.html`

The corresponding base URL is
`file:///w/337/s/`

Thus `href="fp.html"` is equivalent to
`href="file:///w/337/s/fp.html"`

Is it better to use absolute URLs or relative URLs?

The http URL scheme

A more common scheme is "http". Here are some absolute URLs with the http scheme:

http://google.com

http://localhost:8080/x.html

<http://support.microsoft.com/?kbid=315265>

URLs using the http scheme can have several parts:

scheme *://* *host* *:port* */ path* *? query-string* *# fragment*

:port and */ path* ... are optional.

The http scheme, continued

When the "file" scheme is used, the browser opens the file directly.

When the "http" scheme is used, the browser uses the Hypertext Transfer Protocol (HTTP) to fetch the data.

The "https" scheme is very similar but uses Hypertext Transfer Protocol Secure (HTTPS).

We'll talk about HTTP and HTTPS later.

The http scheme, continued

Here's a URL that accesses another copy of anchor1.html:

```
http://www.cs.arizona.edu/classes/cs337/fall13/  
slides/anchor1.html
```

What's the base URL in this case?

Here's a URL with a fragment. What does it reference?

```
http://www.cs.arizona.edu/classes/cs337/fall13/  
slides/hagloss1.html#p
```

The mailto scheme

The "mailto" scheme produces a link that when clicked opens the user's mailer and creates a message.

Or ``
mail me`` for more info!

Due to spammers, public facing "mailto" URLs are seldom used.

anchormail1.html

Images

Images

The `img` element displays an image.

The `src` attribute specifies the source of the image, which can be a local file or a URL (more on URLs soon)

```

```

```

```

Images have inline display.

`img` is a void element.

Image sizing

By default, an image is shown at its native size in pixels.

Use width and height attributes to display at other sizes.

If only one is specified, the other is calculated to preserve the *aspect ratio* (width:height)

If both are specified, the image is stretched.

Example: `img1.html`

Image sizing, continued

Image resizing for width/height attributes is done in the browser.

Shrinking images in the browser wastes network bandwidth.

HFHC recommends always specifying width and height, to save recalculation of layout.

With PHP we'll use ImageMagick for resizing

img's alt attribute

The alt attribute "defines the alternative text describing the image. Users will see this displayed if the image URL is wrong, the image is not in one of the supported formats, or until the image is downloaded." – MDN

There's a LOT written about alt in the CR.

Example: [imgalt.html](#) (try with Chrome, Firefox, and Opera)

Image formats

There are three widely supported image formats:

JPEG Joint Photographic Experts Group

GIF Graphics Interchange Format

PNG Portable Network Graphics ("ping")

JPEG

Created for compressing photographs

Millions of colors

No transparency

"Lossy" compression; artifact on drawn lines

Compare blockA-q20.jpg and blockA.png

Can trade off compression against image quality

Examples on <http://en.wikipedia.org/wiki/Jpeg>

GIF

Maximum of 256 unique colors per frame, from palette of 16 million.

Can contain multiple frames, for animation

Lossless compression

One transparent color per frame—provides transparency but not of varying degree

"Choosy developers choose GIF"

Example: [imggif1.html](#)

PNG

Lossless compression; might be smaller or larger than corresponding GIF.

No animation

Supports variable transparency (see [imgxp1.html](#))

PNG-8 (256 colors), PNG-24 (16M colors),
PNG-32 (16M colors + 8-bit transparency)

Creation motivated by IP (intellectual property) issues with GIF compression algorithm

Examples: [blockA.png](#), [awvip.png](#), and others

PNG-8 or PNG-24?

Two ways I know to find out which you've got:

```
% file *.png # *NIX file(1) (do "man file" for documentation)
```

```
blockA.png: PNG ..., 3264 x 3219, 8-bit/color RGBA, ... 24
```

```
googlelogo.png: PNG ..., 550 x 190, 8-bit colormap, ... 8
```

```
lavender.png: PNG ..., 2592 x 1944, 8-bit/color RGB, ... 24
```

Open with Photoshop and look at top bar:

"x.png (Index)" (PNG-8)

"y.png (RGB/8)" (PNG-24)

GIF, JPEG, or PNG?

Photograph?
JPEG!

Animated?
GIF!

Drawing?
GIF or PNG

Transparency needed?
GIF or PNG

1.9M lavender.gif
2.6M lavender.jpg
6.5M lavender.png
(img3.html)

Good stuff on image formats: HFHC pp. 167-169

Adding transparency with GIMP

I don't know much about GIMP, but having said that...

File>Open...

Layer>Transparency>Add Alpha Channel

Select "Fuzzy Select Tool" (a.k.a magic wand)

Click on an area you want make transparent

Maybe: undo, adjust threshold, and try again

(Windows>Dockable Dialogs>Tool Options)

Right click>Edit>Clear

Gray checkerboard should appear

File>Export, select GIF or PNG for File Type

See HFHC p.204-208 for a technique involving backgrounds.

[imgxp.html](#)

Simple drawing with GIMP

Multi-color filled rectangle:

File > New..., pick dimensions

Pick color using swatches at bottom of toolbox

Select Bucket Fill tool

Click in rectangle to fill it

Select Rectangle Select Tool

Drag with mouse to sweep out a rectangle

Note tool options for Fill Type and Affected Area

Make marks with Pencil, Brush, or Airbrush

A key to avoiding madness: If there's a selection active, you can only draw in it! (Select>None to clear it.)

Viewing GIF frames with GIMP

Open file

Right click>Filters>Animation>Playback

Opens a window (with terminal-like icon) named "Animation Playback"

Seems like no way to go backwards or view all frames at once.

Photoshop: just open it and look for "Animation (Frames)" tab.

Supported Formats

The HTML5 CR does not require that any specific image format be supported.

img's src must reference "a non-interactive, optionally animated, image resource that is neither paged nor scripted."

Support for GIFs, JPEGs, and PNG is common.