

Department of Computer Science
University of California, Los Angeles

Computer Science 144: Web Applications

Spring 2025
Prof. Ryan Rosario

Lecture 2: April 2, 2025

Outline

- 1 Brief History of Browsers/User Agents
- 2 HTML and HTML5
- 3 Introduction to HTML5 Page Structure

1 Brief History of Browsers/User Agents

2 HTML and HTML5

3 Introduction to HTML5 Page Structure

Brief History of Browsers: 1990/1991

Last time we talked about HTTP, but we don't actually create HTTP requests or pass HTTP responses ourselves.

A user agent does it for us, and that user agent is typically a **browser**.

Brief History of Browsers: 1990/1991 (contd.)

Tim Berners-Lee creates WorldWideWeb. For the first time, structured text documents can link to other documents with just a click.

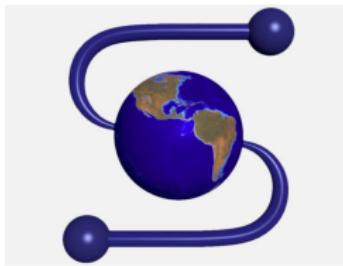


Nicola Pellow, a math student at CERN, created the Line Mode Browser for text terminals.

She is included in the site [No Web Without Women](#).

Brief History of Browsers: 1993

Marc Andreessen and Eric Bina create NCSA Mosaic at UIUC.
Windows only. Also has image galleries and chat rooms.



It was a predecessor to Mozilla Firefox.

Brief History of Browsers: 1995

Andreessen creates *Netscape Navigator* for use on multiple OS.

Microsoft **licenses the code for Mosaic** and develops *Internet Explorer* from it. **The browser wars were on.**



Licensed
→



Netscape develops client-side JavaScript, Microsoft creates CSS.

Brief History of Browsers: 1996–1999

1996



90% marketshare



1999



99% marketshare



Brief History of Browsers: 2000 Onward

2003



2004-2006



2008



1 Brief History of Browsers/User Agents

2 HTML and HTML5

- Introduction to HTML Tags for Structure

3 Introduction to HTML5 Page Structure

HTML and HTML5

This year we will not do a thorough review of HTML.

Instead, we will focus on aspects of HTML that are frequently missed by developers and in classes that are not focused on the web.

We will also discuss HTML5 and standards.

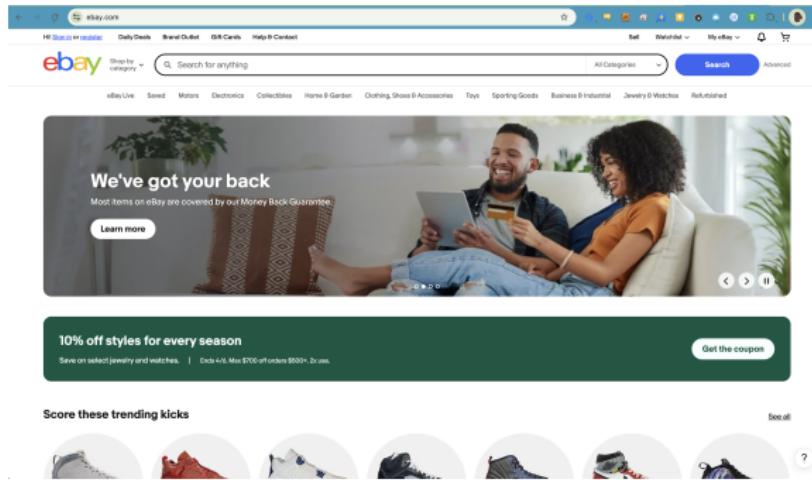
HTML and HTML5 (contd.)

If you need a refresher, or want to read more history, please see these references:

- ① [MDN Web Docs](#)
- ② [Introduction to HTML](#)
- ③ [WHATWG HTML Living Standard \(canonical spec\)](#)
- ④ [W3C HTML5 Specification](#)
- ⑤ [History of HTML](#)

Introduction to HTML

When we use a browser we do not see HTTP requests and responses.



How is this structure represented? HyperText Markup Language (HTML).

Brief History: 1.0 to 4.01

- ➊ HTML 1.0 (1993) – Tim Berners-Lee
 - 18 tags, capitalized, text only
 - no images (1.2), tables, frames, forms, pixels
- ➋ HTML 2.0 (1995) – IETF
 - <html>, <head>, <body> sections
 - added support for images, forms, tables, meta tags
 - more text formatting
 - beginning of HTML validation (DTD)
- ➌ HTML 3.2 (1997) – W3C
 - CSS, frames, basic JavaScript
 - client-side image maps
- ➍ HTML 4.0 (1997) – W3C
 - more CSS and JS
 - a lot was deprecated
 - international character sets, <div>,
 - SEO, accessibility, **huge leap forward**

Image Map Example from HTML 3.2

Image Maps were quite a thing back then. But the image could take a while to load, and the code was tedious.

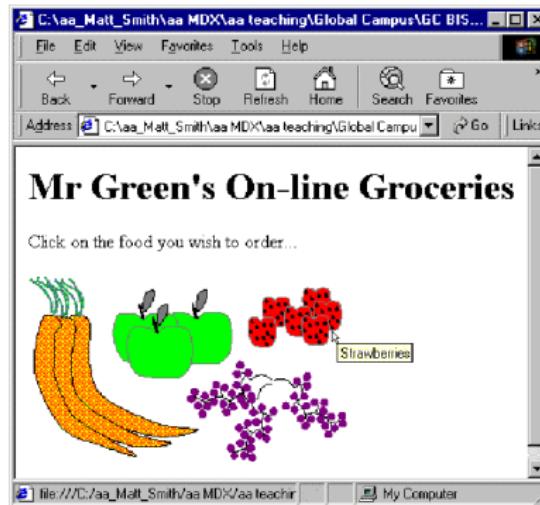


Image Map Example from HTML 3.2 (contd.)

The code would look something like this:

```
<map name="geometry">
<area shape="rect" cords="10,10,50,50"
      href="square.html">
<area shape="circle" cords="100,100,40"
      href="circle.html">
<area shape="poly" cords="100,200,200,200,200,300,100,200"
      href="triangle.html">
</map>
```

XHTML (2000)

In 2000, things got weird for a bit...

- HTML was a mess — tag soup, few standards
- Every browser handled syntax errors differently
- Developers had to rely on the browser's implementation

XHTML tried to combine the ease of HTML with the strictness of XML. HTML expressed as XML rules.

XHTML (2000) (contd.)

Example of problematic HTML:

```
<p>Some text
    
    <p>Powell Cat was a special animal</p>
</p>
```

Each browser would deal with this situation differently.

XHTML (2000) (contd.)

An XHTML web page starts out as follows:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5  <head>
6      <title>XHTML 1.0 Example</title>
7      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8  </head>
9  <body>
10     <h1>Welcome to My XHTML Page</h1>
11     <p>This is a simple XHTML document. All tags must be lowercase.</p>
12     <p>Elements must be properly closed, and document must be well-formed.</p>
13 </body>
14 </html>
```

XHTML (2000) (contd.)

You will still see web pages that use XHTML, but HTML5:

- Does not use the XML line
- Does not use the namespace declaration
- **Does** use the DOCTYPE line

HTML5 (2014 to today)

HTML5 was revolutionary because it standardized and integrated several features commonly used on the web in one technology.

Prior the developer had to rely on:

- ① hacky JavaScript
- ② external tools
- ③ plugins
- ④ server-side fallbacks

HTML5 (2014 to today) (contd.)

New features can be categorized as —

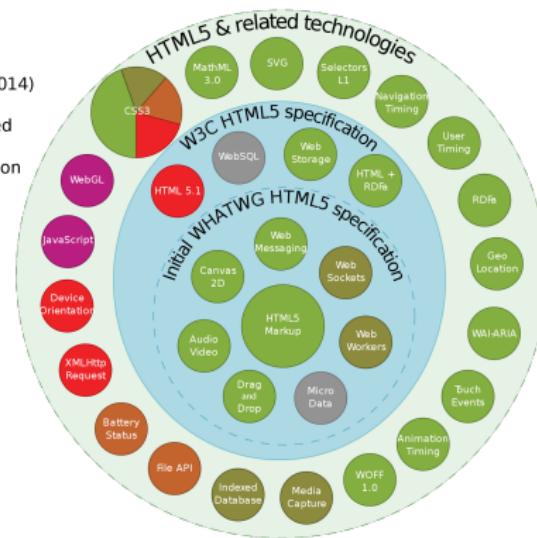
- ① **Semantic elements** — higher level than `<div>` or ``
- ② **Enhanced multimedia support**
- ③ **Graphic and visualizations** — `<canvas>` and SVG for graphics, animations, games.
- ④ **New form controls and input types**
- ⑤ **Improved cache control** — background and offline data synchronization
- ⑥ **APIs** — geolocation, drag-and-drop, web storage

HTML5 (2014 to today) (contd.)

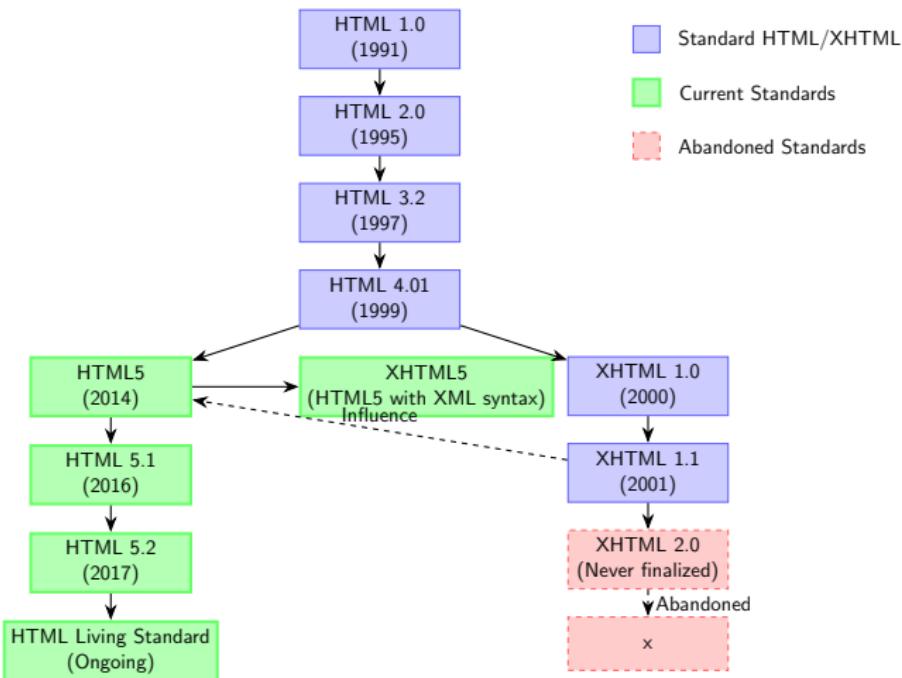
HTML5

taxonomy & Status (October 2014)

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

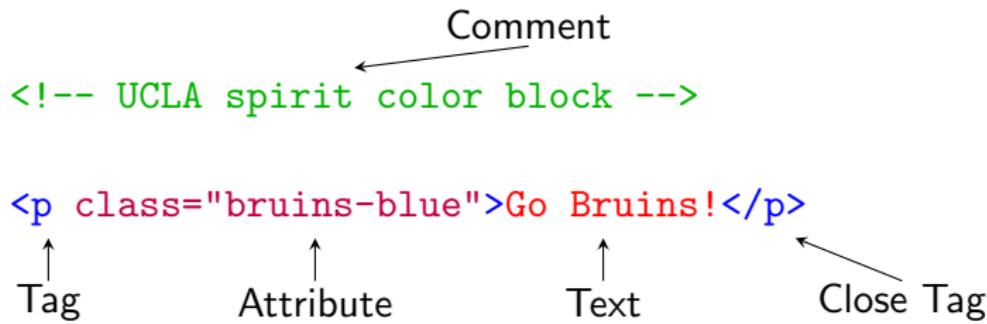


Brief History Summary



HyperText Markup Language (HTML) Constructs

HTML is the standard way of representing **structure** but not style these days.



Void tags do not have closing tags: `
`, `<hr/>`, `<img... />`, `<input... />`.

Types of HTML Tags

HTML tags fit into one of two classes:

- ① **Block-level**: always starts on a new line, and the browsers add space (a margin) around the block before and after. *They always use the full width available.*
- ② **Inline**: does not start on a new line. *They use only as much width as necessary.*

HTML Tag Attributes

Tags can have attributes:

- For example
 - src (source) is the attribute, and for img, is a very important one!
- Enclosed in "" (double) or '' (single), should be lowercase
- id attribute is special. It serves as a "unique key" for an element. Cannot have more than one element with a particular ID.
- class attribute is special. It declares that tag as needing to be styled a certain way. It's a way of grouping related elements together.

HTML Tags Are For Structure, Not Presentation

One nuance commonly overlooked: some commonly used tags are so specific that they are actually presentation than structure, **and have been replaced.**

`` and `<i>` tags are still commonly used on the web.

Semantically, **bold** is for strong emphasis while *italics* is for *stress emphasis*.

So use `` and `<emph>` instead, respectively.

Then let the user control how the emphasis is displayed.

HTML Tags Are For Structure, Not Presentation (contd.)

Some others:

- →
- <i> →
- <tt> → <code>
- <s> → (strikout to deletion)
- <u> → <ins> (underline to insertion)

Important Structure Elements: div and span

The special tags `<div>` and `` are used to divide up a webpage into units

- The `<div>...</div>` tag places the enclosing HTML code into organized units or sections. divs can be nested within each other. It is block level, and starts on a new line, uses full width of its container, and imparts a new line after it.
- The `...` on other hand is used inline. Does not start a new line and only occupies the space the element needs.

If you ever think you may need to style something, put it into a div or a span. Both can have an id or class attribute for CSS styling.

Special Characters

Since < and > are used to denote tags, and "" are used to denote attributes, we must represent them differently, using *HTML entities* (similar to escaping)

Entity	Display
<	<
>	>
"	"
 	Prevents the browser from (1) reducing several spaces into one and (2) inserting a line break.
&	&

There are many others. See [here](#).

Line breaks may be ignored. Multiple whitespaces are trimmed to one. The only exception is when using the <pre> and <textarea> tags

1 Brief History of Browsers/User Agents

2 HTML and HTML5

3 Introduction to HTML5 Page Structure

Over the Years...

Over the years, the structure of an HTML page has changed slightly.

We will introduce the HTML5 skeleton of a web page right now as it is the standard we will use.

Later in this lecture we will discuss additional features of HTML5 in the context of the history of HTML.

Anatomy of an HTML(5) Page

The bare minimum...

```
<!DOCTYPE html>
<html lang="en">
<head><title>...</title></head>
<body>...</body>
</html>
```

Note that the DOCTYPE line is very distinct in each version of HTML. CSS style can also be embedded in the head section by using the `<style>...</style>` tags.

The Body

The body contains the main content of the page, and makes use of all the tags we learned about earlier.

Note that Javascript can be embedded in the body by using the `<script>...</script>` tag though it can also be in the head.

CSS style can also be embedded in the body by using the `style` attribute of tags, or by using `div` and `span`.

The Body (contd.)

All of your HTML for this class must be validated.

One validator is [here](#).

Some HTML Style to Keep in Mind

As we move forward learning and using HTML, there are some style issues that we want to enforce:

- ① Use 2 spaces, not tabs, for indentation.
- ② Tags and attributes are lowercase.
- ③ Tags must be closed, unless they are void.
- ④ When using attributes, there is no space around the = sign and use quotes ``.
- ⑤ When using images, always specify the height, width and alt text.
- ⑥ Set the viewport.
- ⑦ Set the DOCTYPE and document language.

Note that we will cover a lot more, but these are some good ones to start with.

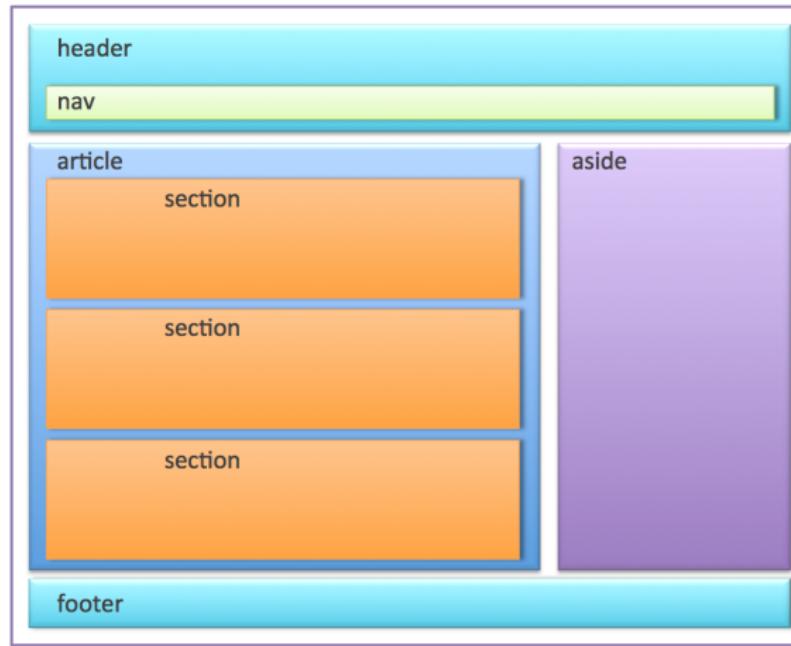
HTML5 Features

We will go through a demo of several of HTML5's features. The rest of these slides just summarize syntax and you can read more on your own.

HTML5: Semantic Tags

For each part of an HTML document, we used to have to use `<div>` and `` to style common page layout components like headers, footers and sidebars.

HTML5: Semantic Tags (contd.)



This image is part of the Bioinformatics Web Development tutorial at http://www.cellbiol.com/bioinformatics_web_development/
© cellbiol.com, all rights reserved

HTML5: Multimedia

```
<video id="demo-video" width="600"  
    controls autoplay muted loop  
    poster="thumbnail.jpg">  
    <source src="example.mp4" type="video/mp4" />  
    <source src="example.webm" type="video/webm" />  
    Your browser does not support the video tag.  
</video>
```

HTML5: Other APIs

We will do a demo of the following features:

- The headspace
- Video
- Geolocation API
- Drag and Drop API

References

- [JSFiddle: Interactive practice with HTML, CSS, JS](#)
- [HTML Validator](#)
- [Mozilla HTML Documentation](#)
- [HTML Standard](#)
- [Google HTML/CSS Style Guide](#)