

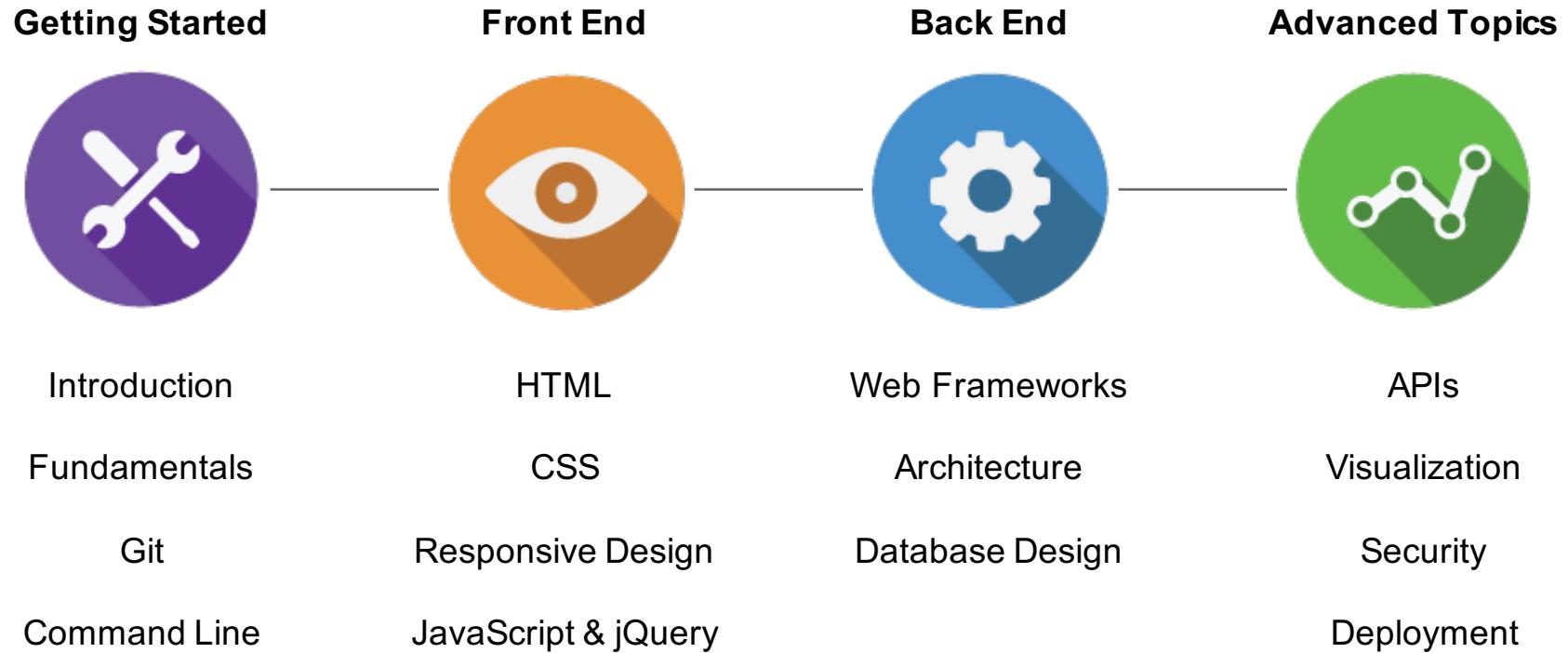
IO Lab

Spring 2016

Module 2 - Lec 1 - HTML

Jan 27, 2016

Course Map



Course Map

Getting Started



Fundamentals

Git

Command Line

Front End



HTML

CSS

Responsive Design

JavaScript & jQuery

Back End



Web Frameworks

Architecture

Database Design

Advanced Topics



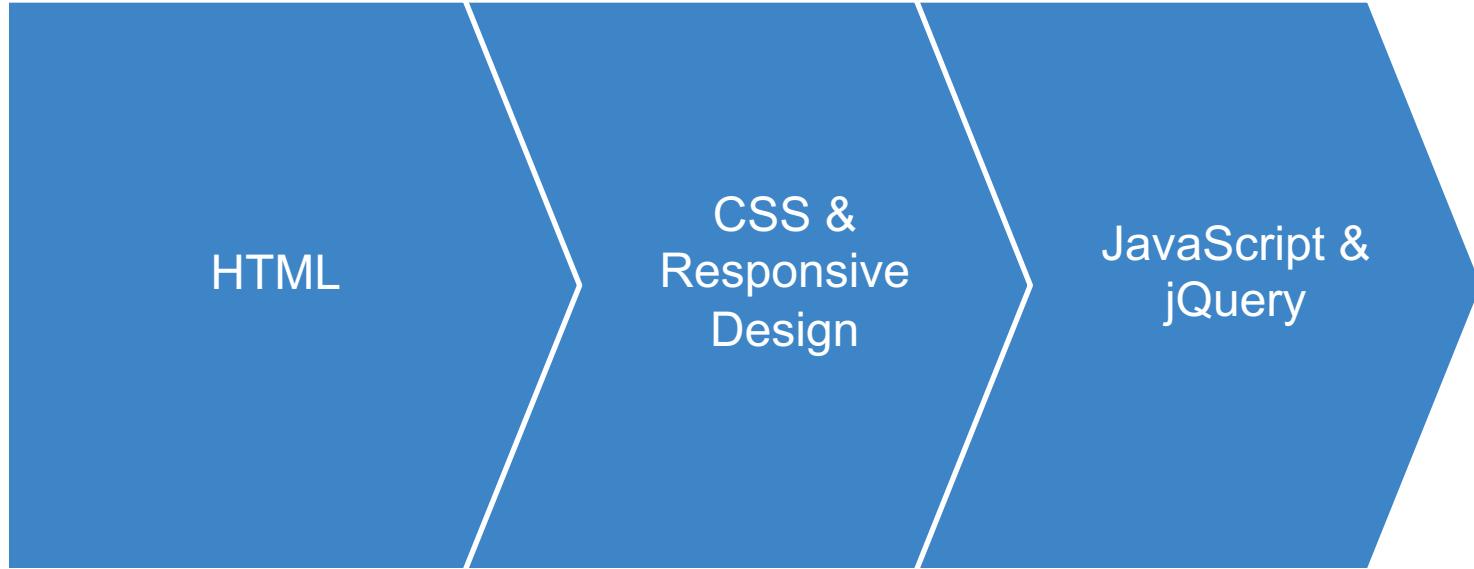
APIs

Visualization

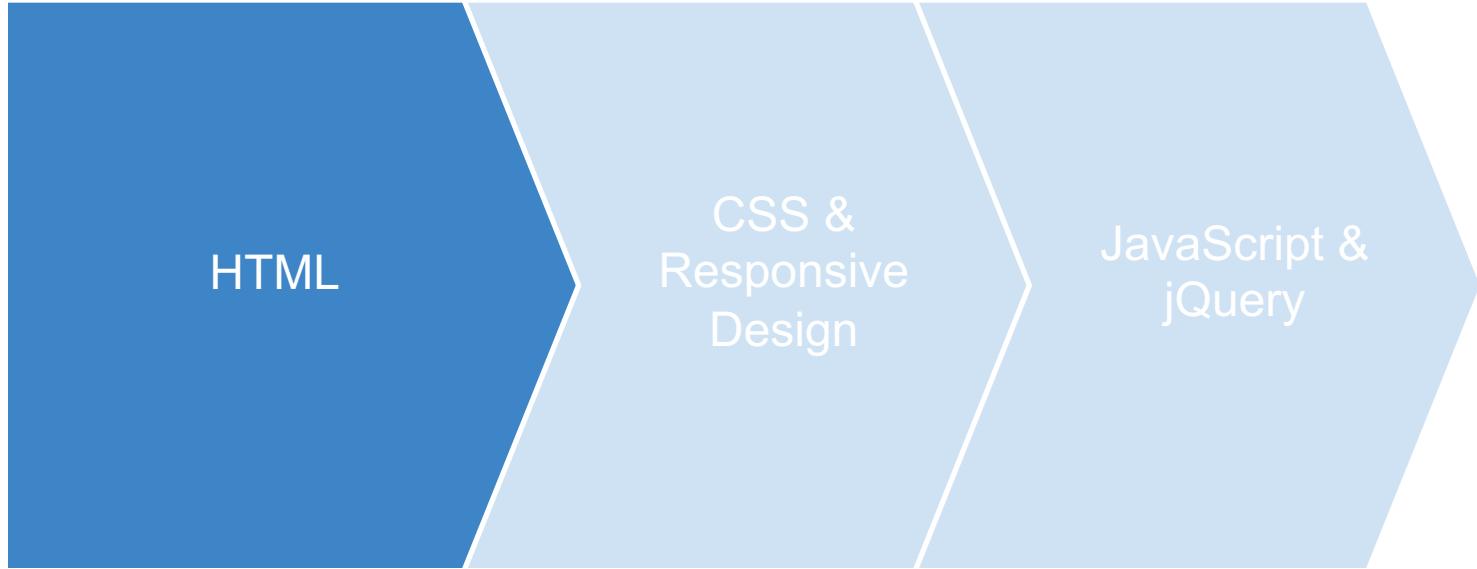
Security

Deployment

Front End Module



Front End Module - HTML



Goals Today

- Foundational understanding of HTML
- Brush up & fill in the gaps
- Think & design before coding

HTML

- Hypertext
- Markup
- Language

HTML



HTML - What is it?

- **Hypertext:** text that links to other documents and resources
- **Markup Language:** association of context for definition, display, and processing
- A predefined vocabulary and set of rules for creating Web pages
- A set of standards recommended by the W3C (Worldwide Web Consortium)

Dig deeper with the W3C Spec

Wait, what is the W3C?



“The World Wide Web Consortium (**W3C**) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards.”

- W3C website

Mission: “To lead the Web to its full potential.”

HTML - Content and Structure

- Contains a web page's content to be rendered, whether it's hard-coded or dynamically generated
- Provides structure to a web page
 - It's like the infrastructure of a building
- What does CSS and Javascript do then?



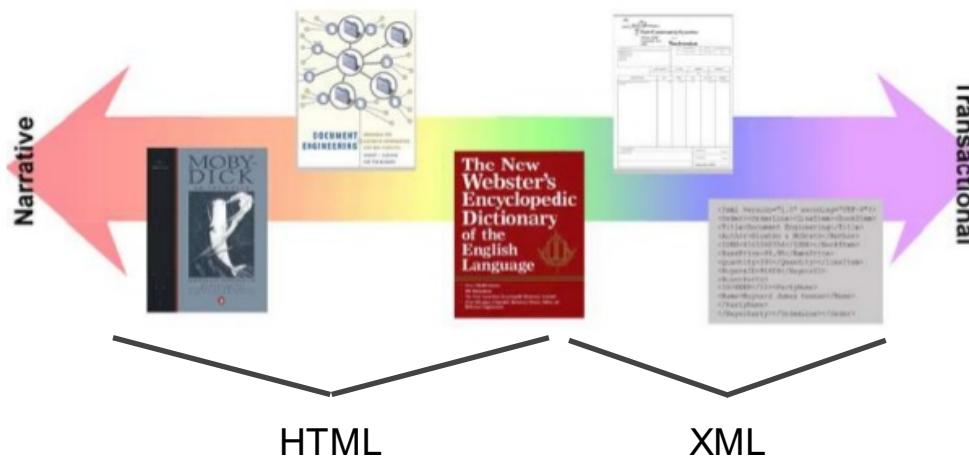
HTML vs XML

HTML	XML
<ul style="list-style-type: none">• Used to display data• Pre-defined vocabulary• Forgiving syntactical rules• Focus on human readability• Focus on Web development	<ul style="list-style-type: none">• Used to describe/store data• User-defined vocabulary• Strict syntactical rules• Focus on machine readability• Focus on transactional documents

[For more on XML, check out the W3C spec](#)

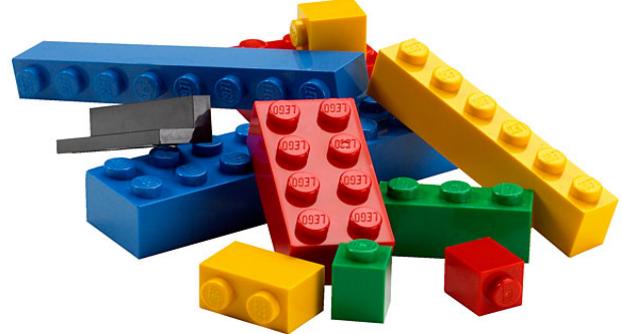
HTML, XML, & The Document-Type Spectrum

The Document Type Spectrum – A
Continuum Between Documents and Data



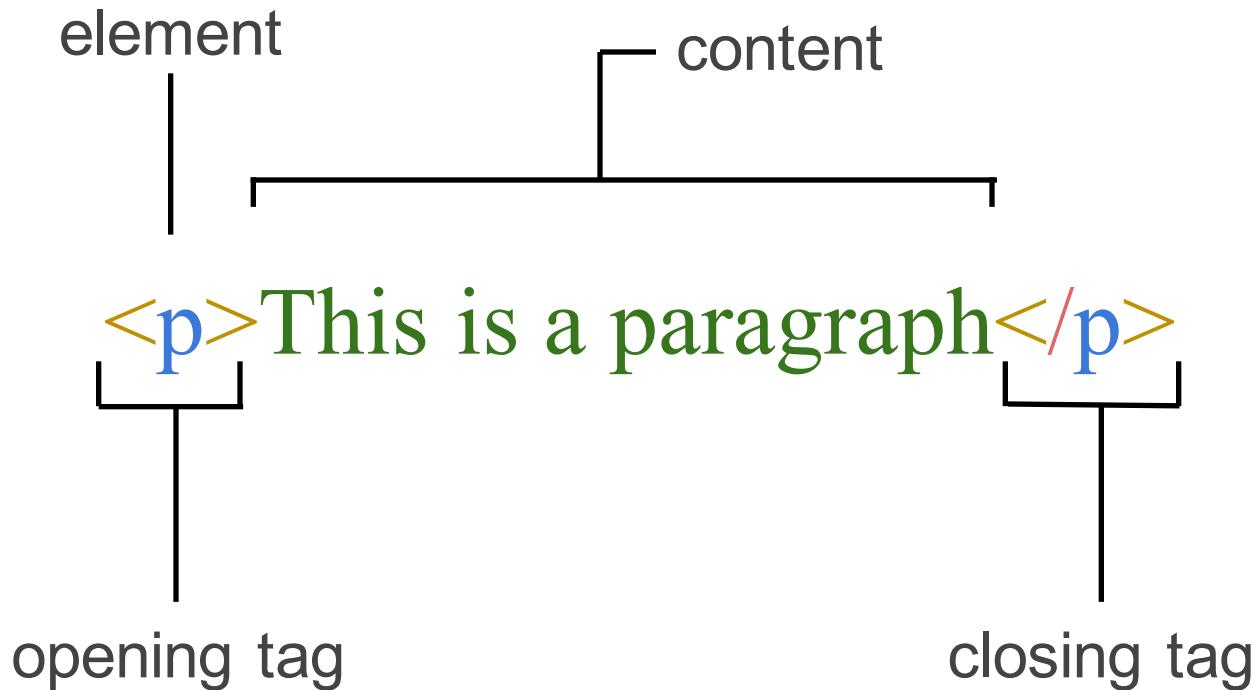
- *We'll primarily be focusing on HTML in this course*

HTML - Elements & Tags



- Elements are the building blocks of HTML
- Tags are used to mark beginning & end of an element
 - Opening tags and closing tags: <> </>
 - ex: <p>This is a paragraph</p>
- The content between the opening and closing tags is rendered
 - In the above paragraph element, “This is a paragraph” is the content and will be rendered

HTML - Elements & Tags



HTML - Common Element Types

- div (division) - used for sectioning & styling; no semantic value!
- span - used for small bits of text & styling; no semantic value!
- h1 - h6 - headers, ranging in hierarchical importance
- p (paragraph) - used for larger blocks of text
- a (anchor) - used to link to other web pages, internal or external
- ol & ul - ordered and unordered lists (i.e., numbered & bulleted lists)
- img (image)
- button

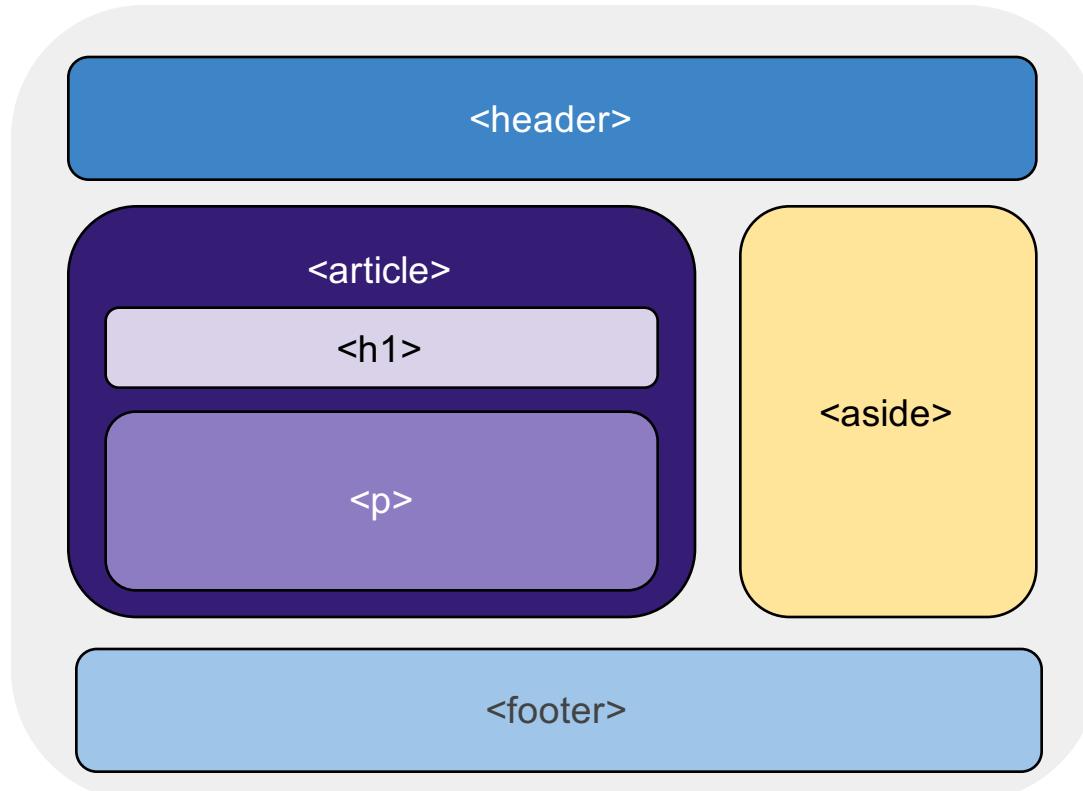
HTML - Common Element Types Cont'd.

Structural Elements

- nav - used for navigation-specific content
- header - not the <head> element
- article - used for the main text of a web page; great for syndication
- section - used for grouping related content
- aside - used for tangential content, like a block quote
- footer - used at bottom of page

[for even more tags, check out HTML Dog's tag reference](#)

HTML - Structure & Meaning Using Types



HTML - Structure & Meaning - You Try It!

1. Pair up with someone next to you
2. Go to the Labs folder in either Github or Bcourses and find the png file of the Large Pokedex
3. Take 5 minutes to draw out on paper how you would organize & nest the content using different elements
4. Minimize your use divs or spans

HTML - Attributes

- Properties that give additional information about an element
 - Element-level metadata
 - Used to style elements, locate resources, and provide a ‘handle’ for Javascript & jQuery
- Located in opening tag
 - Ex: <div id = “example-div” class = “left wide” > Example Div </div>
- Consist of a name and a value
 - Ex: href=”<http://www.google.com>”
- Some elements require attributes (e.g., anchors), while others don’t (e.g., divisions)

HTML - Common Attribute Types

- class - identifies groups of elements to be styled or interacted with similarly
- id - uniquely identifies an element, often for styling, Javascript, or as a fragment identifier
- href (hyperlink reference) - links to other pages (internal or external)
- src - links to source media, often images
- alt - alternate content to be displayed, often used by screen readers; try to have alt attributes whenever possible

HTML - Comments



Always comment your code - for yourself
and for others

Comments enclosed by comment tags:

<!-- -->

Ex: <!-- This is a comment -->

HTML - Good, Clean Code

- Just like comments are important, writing clean, readable code is crucial
 - Be semantic
 - Use comments
 - Use appropriate spacing
 - Use indentation
 - Use appropriate naming conventions
- [Here is a roundup of best practices](#) for writing clean HTML and CSS

HTML - Basic Document Structure

- <!DOCTYPE html> - specifies document type; in this case, HTML
- <html> - the root element; contains all other elements in the HTML doc
- <head> - a container for the web page's metadata; this info is not displayed
 - <meta> - a specific piece of metadata about web page, such as the text encoding (UTF-8 is common)
 - <title> - title of web page
 - <link> - link to additional resources, such as CSS or JS files; can be internal or external
 - CSS files in the head
- <body> - contains the page's visible content
 - <link> to JS files at the end of body

HTML - Basic Structure Example

```
<!DOCTYPE html>

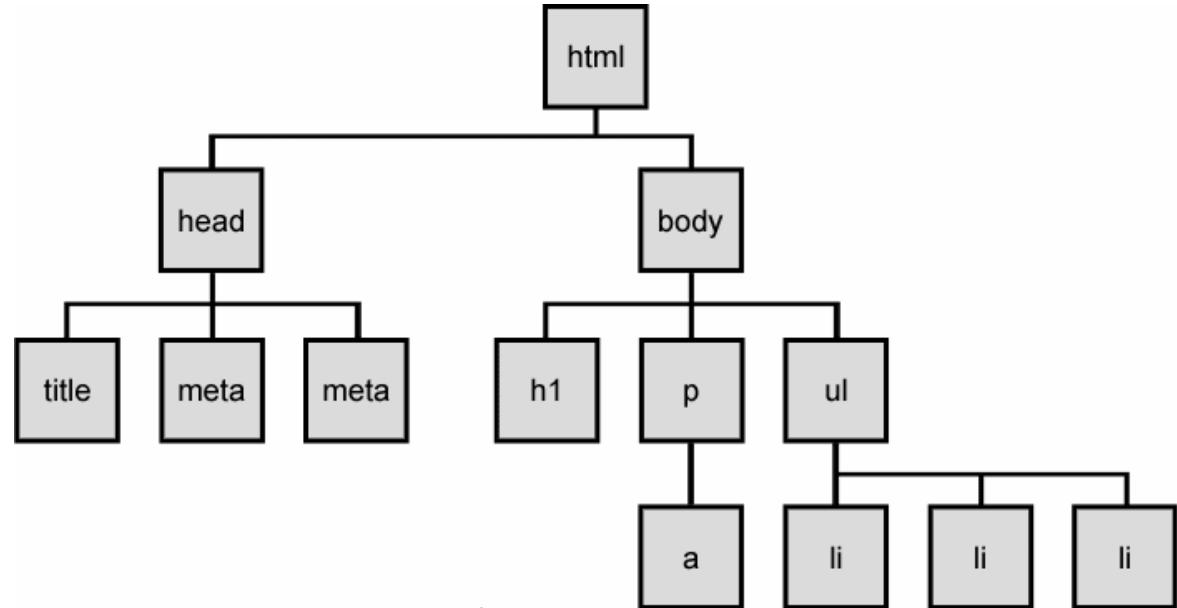
<html>
    <head>
        <meta charset="utf-8">
        <title>A Very Simple Page</title>
    </head>
    <body>
        <h1>This is a header</h1>
        <p>Text goes here.</p>
    </body>
</html>
```

HTML - The DOM Tree

Document

Object

Model



Your browser converts HTML code into a virtual model (the DOM) for rendering and manipulation

HTML - Semantic Code

- Assign meaning based on content, not style/presentation
- Machine-understandable
 - Search engines, screen readers, RSS feeds, etc.
- Human-understandable
 - Describe type of content (divs & spans don't inherently do this)
 - Always assume other people will read your code!

HTML - Semantic Code - Unrealistic?

It is best practice to be semantic when possible

Strive to be semantic, but don't let it hold you back

Many modern web pages != traditional written publications

Sometimes a div is less confusing than awkwardly using a 'semantic' tag

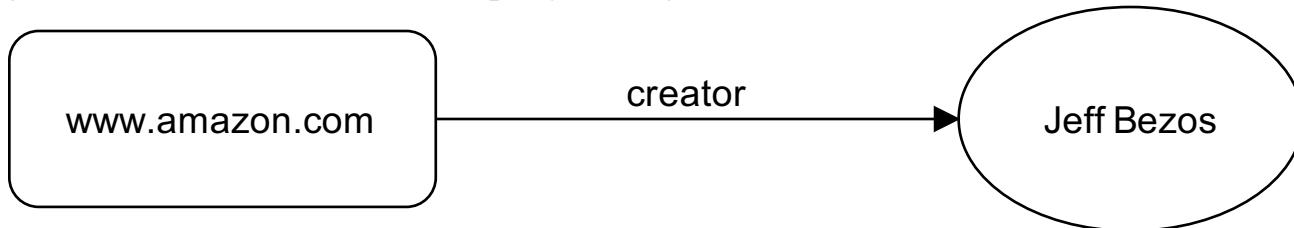
HTML - Unsemantic & Semantic Example

Unsemantic Code	Semantic Code
<div class='footer'>Content</footer>	<footer>Content</footer>
Content	<h1>Content</h1>
<button class="bright-green"></div>	<button class="priority-2"></button>

HTML - Semantics via RDF

- RDF (Resource Description Framework) is a W3C recommendation used to define relationships between entities and make content more machine-understandable
- Directional relationships in the form of triples
 - Subject (Resource) -- Predicate(Property) -- Object(Literal)

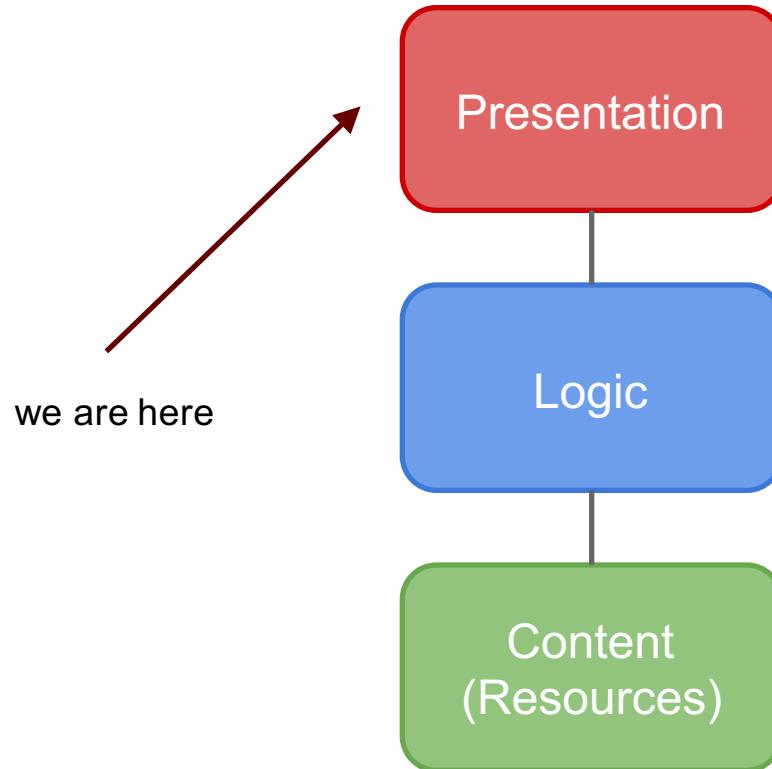
Ex:



[for more, check out the W3C spec](#)

Architectural Thinking

- This is a concept we'll revisit many times this semester
- Separation of concerns
 - Modular
 - Reusable
 - Maintainable



Architectural Thinking cont'd.

- Assign ids & classes strategically
 - Give elements with common styling a single class to avoid repetition and make your life easier when changing styling
 - Name classes and ids based on content/purpose, not styling
- Avoid using in-line styling
 - This is different from using `` and `` semantically
 - Leave the styling to CSS, whenever possible