

Objectives

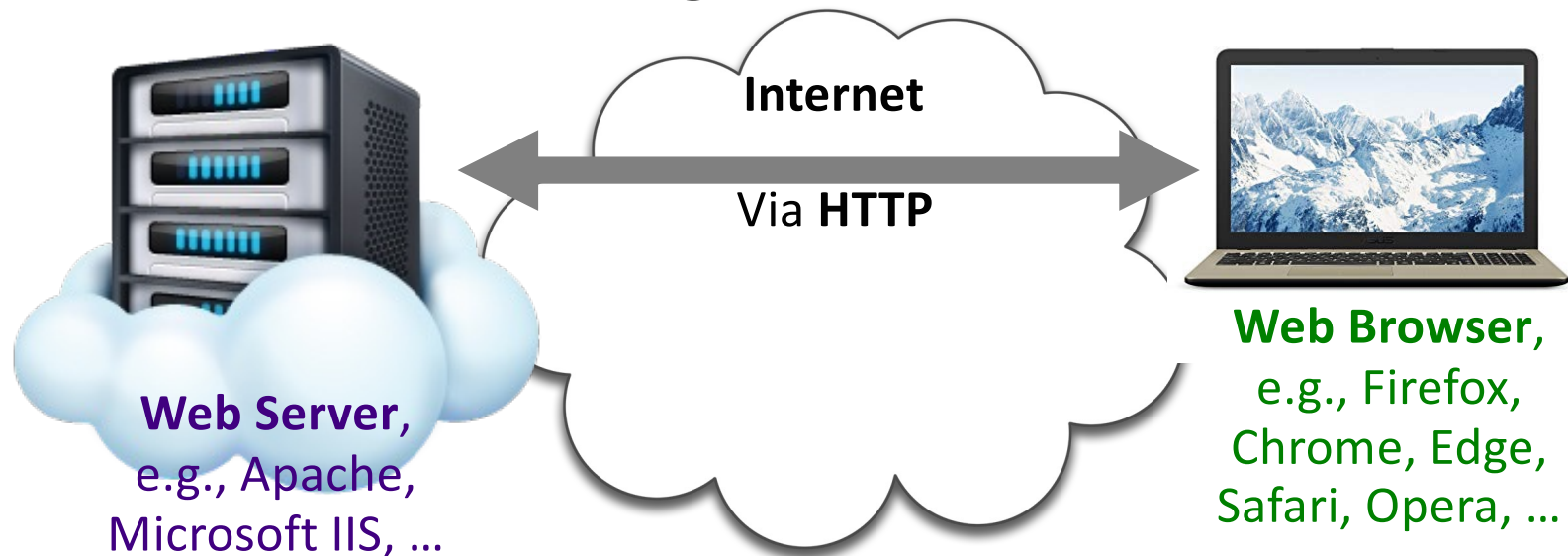
- HTML
- Favicons
- HTML Tool

Review

- What made the WWW possible?
- What are the main components that enable the Web?
 - What protocol do they use to communicate?
- How does the process of retrieving a web page work?
 - Talk through the steps, using the appropriate terminology
- What is the name of the most popular web server?

World Wide Web

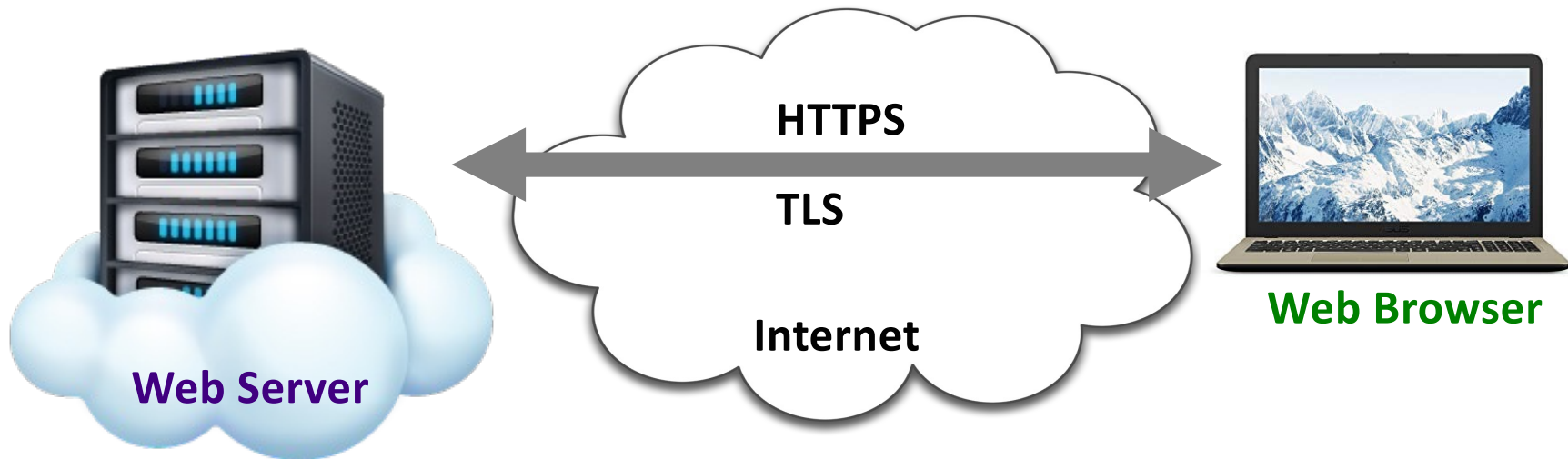
- Built on top of the Internet
- Web browsers and Web servers
- Communicate using HTTP (Over IP/TCP)



https

More when we talk about security

- http + secure
- Encrypted using Transport Layer Security (TLS)
 - Formerly Secure Socket Layer (SSL)



Review: Git

- What are the benefits of version control?
- What is the git workflow?
 - i.e., what is the typical process for working in git?

Discussion

- How did Lab 0 go?
- Questions about anything?

HTML: HYPERTEXT MARKUP LANGUAGE

HTML: HyperText Markup Language

- Describes the *content* and *structure* of information on a web page
 - Not the same as the *presentation* (appearance on screen)
- **Markup** document with *elements*
- Written in plain text

HTML: An Evolving Standard

- Previous versions of HTML
 - “Loose” on standards for how to write/render HTML
 - Benefit: allows for authoring “imperfect” HTML that still renders okay
 - Limitations:
 - Page can look different in different browsers (e.g., mobile devices)
 - Lack of Multimedia, graphical content
 - Lack of semantic information



HTML5

- Current standard
- Goals
 - Support multimedia (Web 2.0 applications)
 - Better definitions for markup
 - Consistent support by browsers
 - Including mobile devices

What does this all mean?

- We're going to write in HTML5
 - Current stable version
 - Tendency towards the XHTML syntax
 - Relatively easy, especially with right tools
 - Less ambiguous documents
 - Can be validated

Extensible HyperText Markup Language (XHTML)

- XML-based version of HTML
 - XML: Extensible Markup Language
 - Used to represent data
- Stricter version of HTML
 - All tags must be closed

Element Syntax

- An **element** is made up of **tags** and **content**
 - Syntax: `<tag>content</tag>`
 - Example: `<p>This is a paragraph</p>`
- If an element contains no content, can open and close a tag together (although not required):
 - Syntax: `<tag/>`
 - Example: `<hr/>`
- Most whitespace is insignificant in HTML
 - It gets ignored or *collapsed* into a single space

HTML Tags

- Some tags can contain additional information called *attributes*
 - Syntax: `<tag attribute="value" attribute="value"> content </tag>`
 - Example: `Next page`
- Tags without content:
 - Syntax: `<tag attribute="value" attribute="value"/>`
 - Example: ``

An HTML Page

- An HTML page is saved in a file with the extension `.html`
- Structure
 - The **header** describes the page
 - The **body** contains the page's contents

```
<html>  
  
<head>  
information about the page  
</head>  
  
<body>  
page contents  
</body>  
  
</html>
```

(E)Mail Metaphor

- You want to send an (e)mail message
- The message is the body
- The head would contain information *about* the message, e.g., recipient, return address, postage/rate

```
<html>  
  
<head>  
information about the page  
</head>  
  
<body>  
page contents  
</body>  
  
</html>
```

Structure of an HTML5 Page

```
<!DOCTYPE html> ← Says we're using HTML5
<html>
<head>
information about the page
</head>
<body>
page contents
</body>
</html>
```

Adding a Title to a Page

- **title** elements go inside of the **head** element

```
<title>CSCI335: Web Applications</title>
```

This text is the title

- Displayed in the web browser's title bar and the bookmark text
- Displayed in search engine results pages

```
<!DOCTYPE html>  
<html>  
  <head>  
    information about the page  
  </head>  
  <body>  
    page contents  
  </body>  
</html>
```

STRUCTURE OF AN HTML5 PAGE: body

Block-level vs. Inline Elements

Two different types of elements

Block-level Elements

- Browser displays a margin of vertical whitespace between block-level elements for separation
 - Starts on a “new line”
- Take up entire width available
- Examples: paragraphs, lists, table cells

Inline Elements

- Must be *nested* inside of a block-level element
- Only takes up as much space as necessary
- Examples: links, code fragments, images
- Browser allows many inline elements to appear on the same line or within the same block-level element

<p> : Paragraph Tag (block-level)

```
<p>I have a dream that one day this
nation will rise up
and live out the true meaning of its
creed:
    "We hold these truths to be self-
evident: that all men are created
equal."
```

- Browser *collapses* whitespace:

```
I have a dream that one day this nation will rise up and live out the
true meaning of its creed: "We hold these truths to be self-evident:
that all men are created equal."
```

Headings: <h1>, <h2>, ... <h6>

- Headings label major areas of the page
- h1 is the most important heading; h6 the least
- *Block-level*

```
<h1>Famous American Speeches</h1>  
<h2>Civil Rights Leaders</h2>  
<h3>Martin Luther King, Jr.</h3>
```

Famous American Speeches

Civil Rights Leaders

Martin Luther King, Jr.

Headings: <h1>, <h2>, ... <h6>

- Headings label major areas of the page
- *Block-level*
- Presentation:
 - 1, 2, and 3 use font sizes that are larger than the default font size
 - 4 uses the default size
 - 5 and 6 use smaller font sizes

Hypertext Links/Anchors: <a>

- Basis of Web: Links to other pages or parts of a page
- Are *inline* elements
- **href attribute** specifies the destination URL
 - Must be placed inside a block-level element, such as <p> or <h1>

```
<p>Martin Luther King gave his most famous  
<a  
href="https://www.americanrhetoric.com/speeches/mlkihaveadream.htm"  
>speech</a> in 1963.
```

Martin Luther King gave his most famous [speech](https://www.americanrhetoric.com/speeches/mlkihaveadream.htm) in 1963.

More on Anchors: <a>

- href URLs can be
 - *Relative*: another page within this Web site
 - Just the resource path
 - Easier to maintain than absolute paths
 - *Absolute*: full URL

```
<p>Today 's <a href="labs/html_lab.php">lab</a> .  
<p><a href="https://www.w3schools.com/tags/">  
HTML Help</a>
```

Today's lab.
HTML Help

Making links

- “Click here” is intrusive—the control interferes with the content
- Press the blue text is even worse
- A link traversal should be ancillary
- Good approaches:
 - Highlight glossary terms
 - Embed the links in footnotes
 - (Who uses this approach a lot?)
 - Allow the anchor to fit into the rest of the document naturally

Optional Anchor Attributes

Attribute	Meaning	Value
target	Where to open the target URL	_self = default, this window/frame _blank → new window _top → full window _parent → parent frame
title	Text to display in a tool tip	The tool tip text
id (preferred) or name	Names an anchor. Use to create a bookmark in a document.	The name From Lab 0 page: goals-for-lab-zero

Nesting Tags

- Tags must be correctly nested
 - A closing tag must *match* the *most recently opened* tag
- Browser may render incorrectly nested tags correctly anyway, but maybe not
- Bad HTML:

```
<p><a href="01-html.pdf">Slides  
<p>  
  This text also links to Lecture</a>
```


Line Break: `
`

- Forces a line break in the middle of a block-level element (*inline*)

```
<p>Today's lab  
The lab is due at 11:59 p.m. <p>The next lab  
is coming soon...
```

Larger gap
between paragraphs



Today's [lab](#).
The lab is due at 11:59 p.m.
The next lab is coming soon...

Alternative: Line Break: `
`

- Forces a line break in the middle of a block-level element (*inline*)

Close with `/>`
immediately

```
<p>Today's lab  
The lab is due at 11:59 p.m. 

The next lab  
is coming soon...


```

Larger gap
between paragraphs

Today's [lab](#).
The lab is due at 11:59 p.m.
Tuesday's lab coming soon...

Images: ``

- Inline element
- Inserts a graphical image into the page
 - `src` attribute specifies image URL
 - For accessibility, use an `alt` attribute that describes the image

```
<p></p>
```

`alt`: Used by search engines, screen readers, browser if image can't be displayed



Images as Links

- Nest inside an `<a>` tag
- Recall: **title** attribute for tool tip

```
<p><a href="https://icanhas.cheezburger.com">  
</a></p>
```



Terminology Check

- What is a(n) ...
 - Element, Tag, Attribute, Inline element, Block element
- Give an example for each
- If I want to make a(n) X, I use the Y element, which is an [inline or block] element

What I want to do (X)	Element (Y)	Block or Inline element?
Heading		
Image		
Link		

Fill in the Table

- If I want to make a(n) X, I use the Y element, which is an [inline or block] element

What I want to do (X)	Element (Y)	Block or Inline element?
Heading	<code><h1>, <h2>, ...</code>	Block
Image	<code></code>	Inline
Link	<code>link</code>	Inline

<https://jsfiddle.net/84atgm5b/6/>

Practice

- What should be in the <body> element of your page if you want your page to look like:

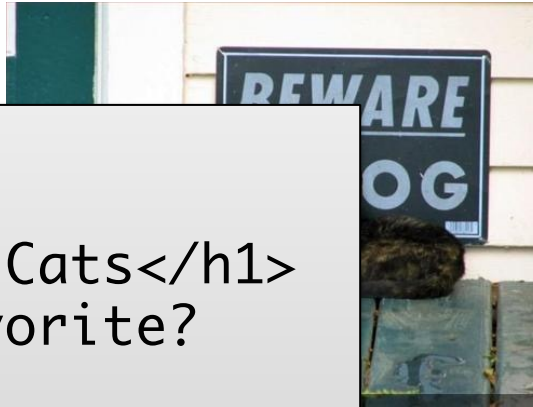


Filename: [beware_of_dog.jpg](#) Filename: [the_ring.jpg](#)

Practice

My Favorite LOL Cats

Which is your favorite?



```
...  
<body>  
<h1>My Favorite LOL Cats</h1>  
<p>Which is your favorite?  
<p>  
  
  
</body>  
</html>
```

lol_cats.html

The Web is World-Readable

- Everything you put on the web is world-readable
- That's good for us!
 - If there is a web page we like, we can see the source used to create it!
 - In your web browser: View the source
- That's bad for us!
 - Others can view (and “steal”) what we do
 - We can obfuscate it, but others can read it

Comments `<!-- -->`

- Document your HTML file
- Comment out text

```
<!-- Lab Web Page, By Sara Sprenkle -->  
<p> Lab is due <!-- tomorrow --> today. </p>
```

Lab is due today.

Three Types of Lists

- Unordered Lists ``
- Ordered Lists ``
- Definition Lists `<dl>`

Unordered List: ``, ``

- `ul` represents a bulleted list (block-level)
- `li` represents a single item within the list (block-level)

```
<ul>
  <li>6-cylinder engine
  <li>Cruise control
  <li>Automatic locks
</ul>
```

- 6-cylinder engine
- Cruise control
- Automatic locks

Nested Unordered Lists

- A list can contain other lists

```
<ul>
  <li>Standard Features:
    <ul>
      <li>4-cylinder engine
      <li>CD player
    </ul>
  <li>Optional Features:
    <ul>
      <li>Cruise control
      <li>Automatic locks
    </ul>
  </ul>
```

- Standard Features:
 - 4-cylinder engine
 - CD player
- Optional Features:
 - Cruise control
 - Automatic locks

Ordered List

- **ol** represents an ordered (by default, numbered) list of items (block-level)

```
<p>Underpants Gnomes' Three-Phase Business Plan</p>
<ol>
  <li>Collect underpants</li>
  <li>?</li>
  <li>Profit</li>
</ol>
```

```
Underpants Gnomes' Three-Phase Business Plan
1. Collect underpants
2. ?
3. Profit
```

Common Error: Not Closing a List

```
<p>Underpants Gnomes' Three-Phase Business  
Plan  
<ol>  
  <li>Collect underpants  
  <li>?  
  <li>Profit  
<p>More information
```

Underpants Gnomes' Three-Phase Business Plan

1. Collect underpants
2. ?
3. Profit

More information

Everything after list is
also indented.



HTML Character Entities

- A way of representing any Unicode character within a web page

Character(s)	Entity
< >	< >
“ ” &	"e; &
™ ©	™ ©
π δ Δ	π δ Δ
é è ñ	é è ñ

➔ How would you display & on a web page?

Phrase Elements:

``, ``, `<code>`

- **em**: emphasized text, usually rendered in italic
- **strong**: strongly emphasized text, usually rendered in bold
- **code**: a short section of computer code, usually rendered in a fixed-width font

```
<p>The ul and ol tags make lists.</p>  
<p>HTML is emreally</em>, strong  
REALLY</strong> fun!</p>
```

```
The ul and ol tags make lists.  
HTML is really, REALLY fun!
```

DOM: Document Object Model

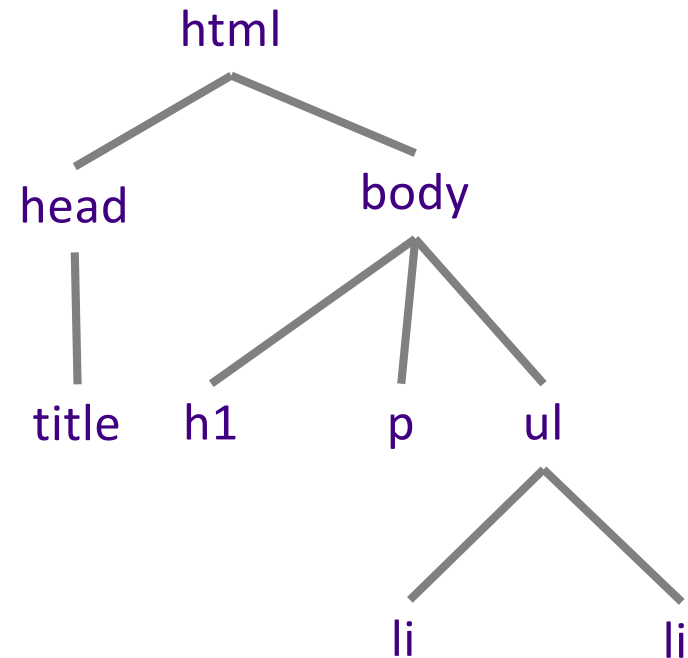
- HTML document's elements can be viewed as a tree

```
<html>
  <head><title>My Web Page</title></head>
  <body>
    <h1>My Web Page</h1>
    <p>My Favorite Movies: </p>
    <ul>
      <li>Tombstone</li>
      <li>The Muppet Movie</li>
    </ul>
  </body>
</html>
```

DOM: Document Object Model

- HTML document's elements can be viewed as a tree

```
<html>
  <head><title>My Web Page
  </title></head>
  <body>
    <h1>My Web Page</h1>
    <p>My Favorite Movies: </p>
    <ul>
      <li>Tombstone</li>
      <li>The Muppet Movie</li>
    </ul>
  </body>
</html>
```



Bad HTML

```
<p>This is <strong>the <em>best</strong>  
time </em> to invest!
```

- What is wrong?
- How did the author want this to display?
- How can we write it correctly?

Bad HTML

```
<p>This is <strong>the <em>best</strong>  
time </em> to invest!
```

- What is wrong?
 - Open/close tags aren't matching
- How did the user want this to display?

This is **the *best time*** to invest!

- How can we write it correctly?

```
<p>This is <strong>the <em>best</em>  
</strong> <em>time</em> to invest!
```

<https://jsfiddle.net/y60L1crz/1/>

Practice Problem

- What should the body of our web page look like if we want it to look like:

The Simpsons Fun Facts



- Longest-running scripted series
- Started on [The Tracy Ullman Show](#)

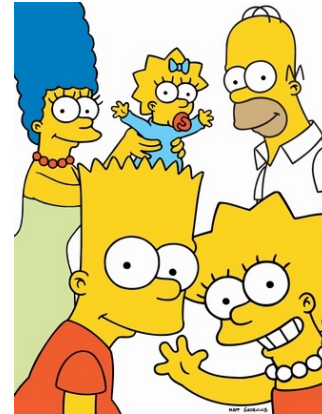
Check back for more info!

Filename:
simpsons_family.jpg

Practice Problem

```
<body>
<h1>The Simpsons Fun Facts</h1>
<p>
<ul>
<li>Longest-running script
series
<li>Started on <a
href="http://en.wikipedia.org/w
iki/The_Tracey_Ullman_Show">The
Tracey Ullman Show</a>
</ul>
</body>
</html>
```

The Simpsons Fun Facts



- Longest-running scripted series
- Started on [The Tracy Ullman Show](#)

simpsons.html

Preformatted Text: `<pre>`

- A large block of code text in a fixed-width font (block-level)
- **pre** blocks are displayed with exactly the whitespace and line breaks given between the tags

```
<pre>
    public static void main(String[] argv) {
        System.out.println("Hello, world");
    }
</pre>
```

```
public static void main(String[] argv) {
    System.out.println("Hello, world");
}
```


Preformatted Text: `<pre>`

- A large block of code text in a fixed-width font (block-level)
- **pre** blocks are displayed with exactly the whitespace and line breaks given between the tags

```
<pre>
    public static void main(String[] argv) {
        System.out.println("Hello, world");
    }
</pre>
```

How would it look if we had instead enclosed it in **code** tags?

Preformatted Text: `<pre>`

- A large block of code text in a fixed-width font (block-level)
- **pre** blocks are displayed with exactly the whitespace and line breaks given between the tags

```
<pre>
    public static void main(String[] argv) {
        System.out.println("Hello, world");
    }
</pre>
```

```
    public static void main(String[] argv) {
        System.out.println("Hello, world");
    }
```

Using code: `public static void main(String[] argv) { System.out.println("Hello, world"); }`

Tags for Quotations

- `<q>`: a short, *inline* quotation

- ▶ Typically displayed with quotation marks, italics

```
<p>Quothe the Raven <q>Nevermore.</q></p>
```

```
Quothe the Raven, "Nevermore."
```

- `<blockquote>`: a lengthy, *block-level* quote

```
<p>Lincoln, in his Gettysburg Address:</p>  
<blockquote><p>Four score and seven years  
ago, our fathers ...</p></blockquote>
```

```
Lincoln, in his Gettysburg Address:  
Four score and seven years ago, our fathers ...
```

Tables `<table>`

- Used to layout tables
 - Sometimes used for layout but better techniques (like CSS), which we'll cover later
- Matrix of rows and columns
- **Cell** is an intersection of row and column
 - Contains almost any document element, e.g., text, headings, images, tables, ...
- Table caption: `<caption>`
 - Inside of a `<table>` element
 - Defines the title that proceeds a table

Table Row: `<tr>`, Label: `<th>`,
Data: `<td>`

- `tr`: defines a row in the table
- `th`: defines a row or column's label (heading)
- `td`: defines a table's data cell

```
<table border="2">
  <caption>Baseball Statistics</caption>
  <tr>
    <th>Name</th>
    <th>Batting Average</th>
  </tr>
  <tr>
    <td>Cal Ripken, Jr.</td>
    <td>.276</td>
  </tr>
</table>
```

Table Row: `<tr>`, Label: `<th>`,
Data: `<td>`

```
<table border="2">
  <caption>Baseball Statistics</caption>
  <tr>
    <th>Name</th>
    <th>Batting Average</th>
  </tr>
  <tr>
    <td>Cal Ripken, Jr.</td>
    <td>.276</td>
  </tr>
</table>
```

Name	Batting Average
Cal Ripken, Jr.	.276

Notice formatting

Tables

- There is a lot more to tables that we won't cover in class
 - Headings, footers
 - Spanning rows and columns

Further HTML Structure

- Introduced in HTML5
- Allow us to semantically structure the content to promote organization and readability
- Examples:
 - `<header>`
 - `<main>`
 - `<nav>`

HTML5 Semantic Elements

Element	Represents	Example Use
<code><header></code>	Introductory content at the beginning of a page, section, or article	Logo, heading, search bar
<code><main></code>	dominant content of the body of a document	Focus of the page
<code><footer></code>	Content that typically signifies the end of a page, section, or article	Bottom of the page – copyright, authorship
<code><aside></code>	Content that is indirectly related to the document's main content	Sidebar, call-out box

Comparing Grouping Elements

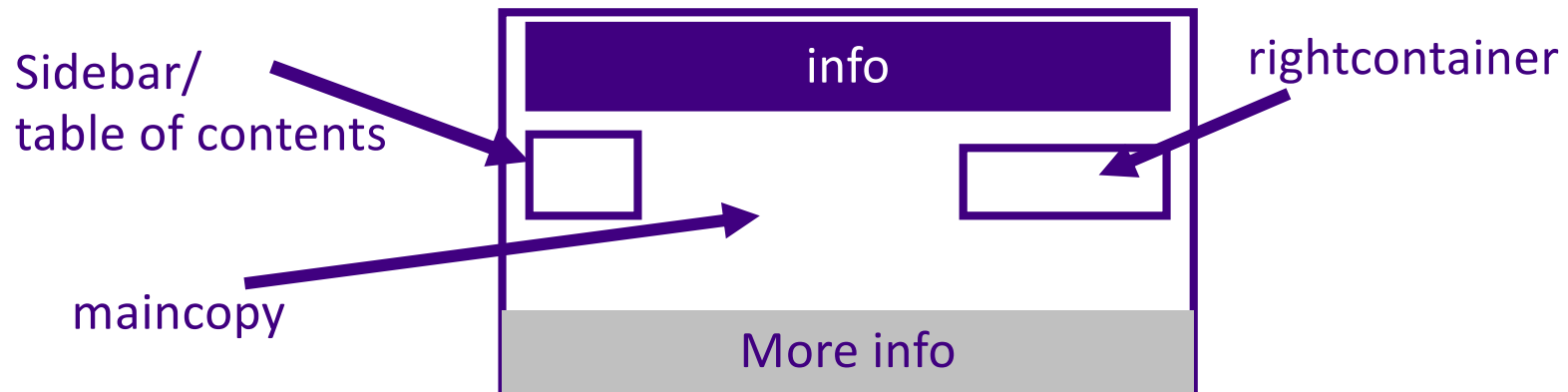
Element	Represents	Example Use
<code><article></code>	Self-contained content, could be independent of the page	a forum post, a blog entry, an article
<code><section></code>	A standalone section within a larger body of content. If separated from that other content, would lose meaning.	search results with a map and its controls
<code><div></code>	A generic container, doesn't add any semantic information	When an <code>article</code> or <code>section</code> doesn't apply

<nav>

- Represents a section of code to provide navigation
 - Links to other documents
 - Links to places within the current document
- Examples: menu, table of contents
- Convention: within ...
 - <aside> element: vertical sidebar
 - <header> element: horizontal navigation

Apply Semantic Elements

- What would be the appropriate semantic element for each of these boxes?
 - Semantic elements: header, main, footer, aside, article, section, nav



Hypothetical Structure of Previous Page

```
...  
<body>  
  <header>info ... </header>  
  <section>sidebar</section>  
  <aside>right-container</aside>  
  <main>main-copy ...</main>  
  <footer>more info...</footer>  
</body>  
</html>
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    information about the page  
  </head>  
  <body>  
    page contents  
  </body>  
</html>
```

STRUCTURE OF AN HTML5 PAGE: head

Web Page Metadata: <meta>

- Represents information about your page
 - For Web browser, search engine, etc.
- Placed into a page's **head** element
- Not displayed in the page itself
- **meta** tags have
 - **content** attribute and
 - **name** or **http-equiv** (for HTTP headers) attribute

```
<meta name="description" content="HTML help page" />  
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />
```

Web Page Metadata: <meta>

- Describe the page

```
<head>
<meta name="author" content="author's name"/>
<meta name="revised" content="web page version and/or
last-modified date" />
<meta name="generator" content="software used to
create the page"/>
</head>
```


Web Page Metadata: <meta>

- Suggestions to search engines about how to index your page
- Search engine may choose to ignore them

```
<head>
<meta name="description"
  content="All about LOL Cats" />
<meta name="keywords" content="LOL cats, cute
kittens, cat images" />
</head>
```

Web Page Metadata: <meta>

- Aid browser or Web server

```
<head>
<meta http-equiv="refresh" content="how often to
refresh the page in seconds" />
<meta http-equiv="Content-Type" content="what
type of document this is (may include
internationalization info such as character
encoding)" />
</head>
```

- Can also use refresh tag to redirect, e.g.,
`content="5;url=https://cs.wlu.edu"`

When would refresh tags be useful?

Favorites Icon (favicon)

- **Link** tag - placed in **head** element

- Can specify an icon for a web page
- Icon is placed in browser title bar and bookmark/favorite

- **Syntax:**

```
<link rel="shortcut icon" type="MIME type" href="filename" />
```

- Default type is **ico**

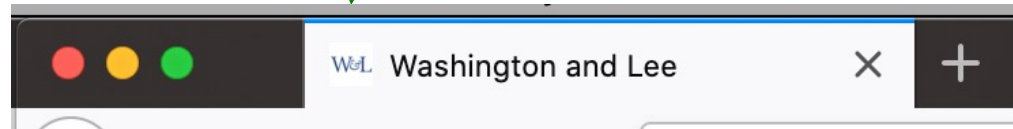
- More you can do here with App icons (for Apple, Android, ...)

Favicon Example: W&L Web Site

```
<link rel="shortcut icon"  
      href="https://www.wlu.edu/favicons/favicon.ico"/>
```

48x48 pixel image

W&L



Internet Media (“MIME”) Types

- **Multipurpose Internet Mail Extensions**
 - Classifications of data that travel over the internet
 - Originally developed for email
- Used to tell the browser the form (**type**) of a file returned by the server
- Type specifications
 - Form: **type/subtype**

Internet Media (“MIME”) Types

- **Multipurpose Internet Mail Extensions**
 - Sometimes used to specify something’s type when including resources on a page
- Form: **type/subtype**

MIME Type	File Extension
text/html	.html
text/plain	.txt
image/gif	.gif
image/jpeg	.jpg
video/quicktime	.mov

Presentation of Web Pages

- Talked mostly about structure and content of HTML pages
- Want presentation to be separate
 - Don't encode style into the HTML page itself
 - Easier to apply different styles to a set of web pages or a whole web site

Cascading Style Sheets (CSS)

- Describe the appearance, layout, and presentation of information on a web page
 - As opposed to HTML, which describes the content, structure of the page
- Describe **how** information is to be displayed, not what is being displayed
- Can be embedded in HTML document or placed into separate **.CSS** file
 - Advantage of separate **.CSS** file: one style sheet can be shared across many HTML documents

Attaching a CSS File: <link>

- **link** is within **head** element
- Can link to multiple style sheet files
- Syntax:

```
<link rel="stylesheet" type="text/css"  
href="filename"/>
```

- Example from W&L site:

```
<link rel="stylesheet" type="text/css"  
href="https://www.wlu.edu/css/site.css?161  
7205978" />
```

Rules vs Conventions

Rules

- Absolutes that cannot be violated
- The browser will enforce rules
 - If you forget to add an ending tag for `strong `, then all of the text following the opening `` tag will appear bold.
- Modern browsers often attempt to fix sloppy HTML by predicting and fixing *some* violations of rules
 - This is NOT an excuse to write poor, incorrect HTML
 - Some browsers or future versions of them may not be so forgiving

Conventions

- Suggested ways of writing code by consensus
- The browser doesn't care about conventions
- The end user viewing your site doesn't care or realize there were conventions followed
- Purely for others who may revise or build upon your code
 - Clear organization
 - Consistent style
 - Highly readable

My First Web Page

- Circa 1995

<https://cs.wlu.edu/~sprenkle/personal/China.html>

HTML TOOLS

W3C HTML Validator

<https://validator.w3.org>

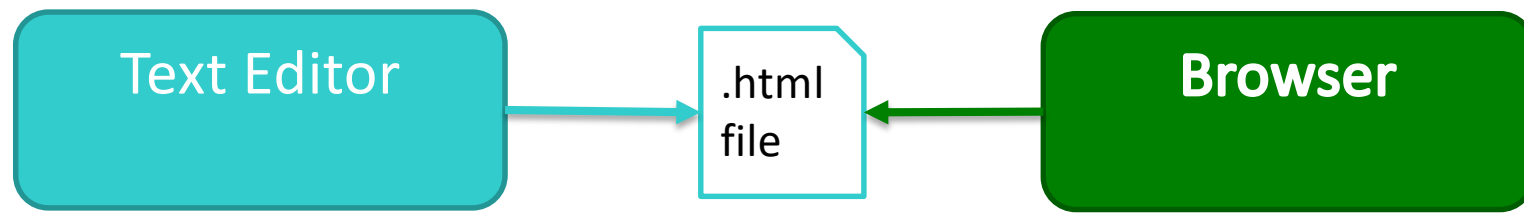
- Checks your HTML code to make sure it meets the HTML specifications
- More picky than the Web browser, which may render malformed HTML in the way the user wanted

Link checker – used in lab

WORKFLOW FOR DEVELOPING WEB PAGES

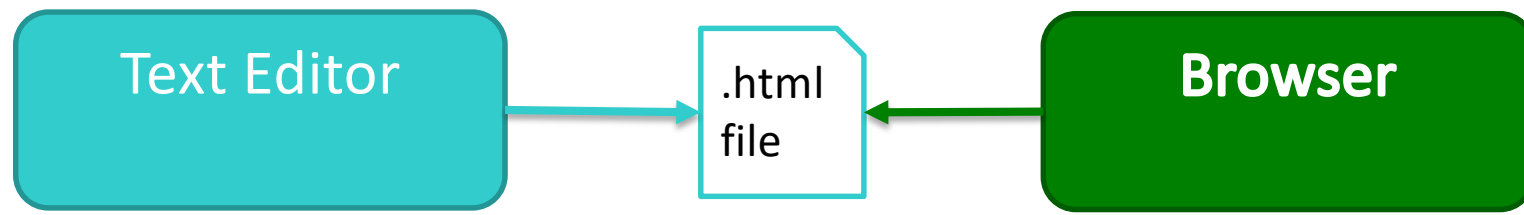
Workflow for Developing Web Pages

- Text editor for writing HTML pages
- Browser for viewing the HTML pages



Workflow for Developing Web Pages

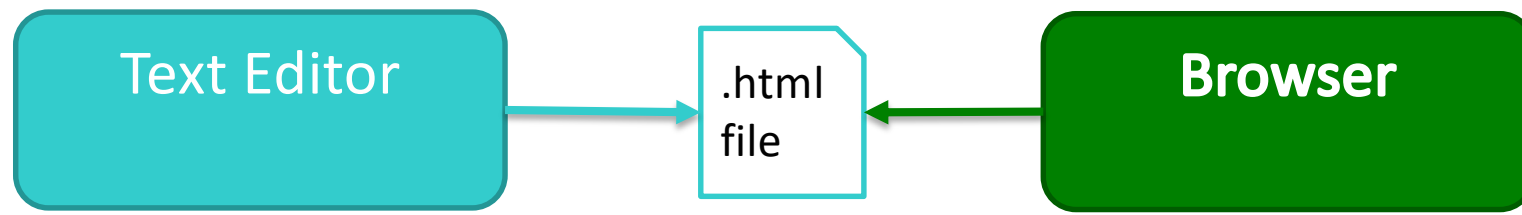
- Text editor for writing HTML pages
- Browser for viewing the HTML pages



In the end, we need this file to be on the (web) server, in your `public_html` directory.
But, while we're developing, it doesn't need to be.

Workflow for Developing Web Pages

- Text editor for writing HTML pages
- Browser for viewing the HTML pages



Scenario	Text Editor	Browser
Physically with CS lab machines	CS web server	CS lab machine
On Campus or Remote with great internet	ssh to CS lab machine	On your personal laptop
Remote	Personal machine. Later, put on server	Personal machine: open the file

Lab 1: HTML, Web Tools, Favicon

- I changed it – no longer using git
 - I am hopeful that I caught all the changes
- Issue: getting your files/images to the web server
 - Using scp, sftp
- Alternatively, could set up a git repository and add files to it on your personal machine and set up git on the web server to pull changes to the repo

TODO

- Lab 1
- Read about how Google search works
 - Write summary in Canvas discussion by **Wednesday 11:59 p.m.**

- Celebrating 35 years of the web!