HTML and Websites

Reese Hatfield

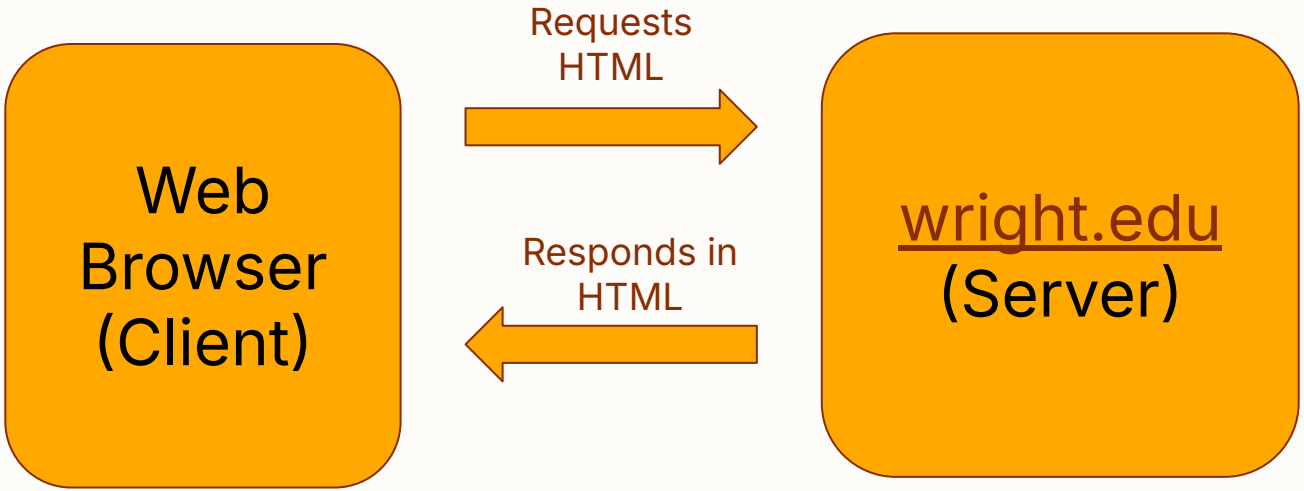
HTML

- The internet runs on website
- But how do we make one?

- HTML

- HyperText Markup Language
- “Programming language**”

HTML


Requests
HTML
Responds in
HTML
Web
Browser
(Client)
wright.edu
(Server)

HTML

- Web browsers are just HTML viewers
- They know how to look at HTML and turn it into a fancy looking website
  - HTML as a standard
- All just implement the same document

HTML

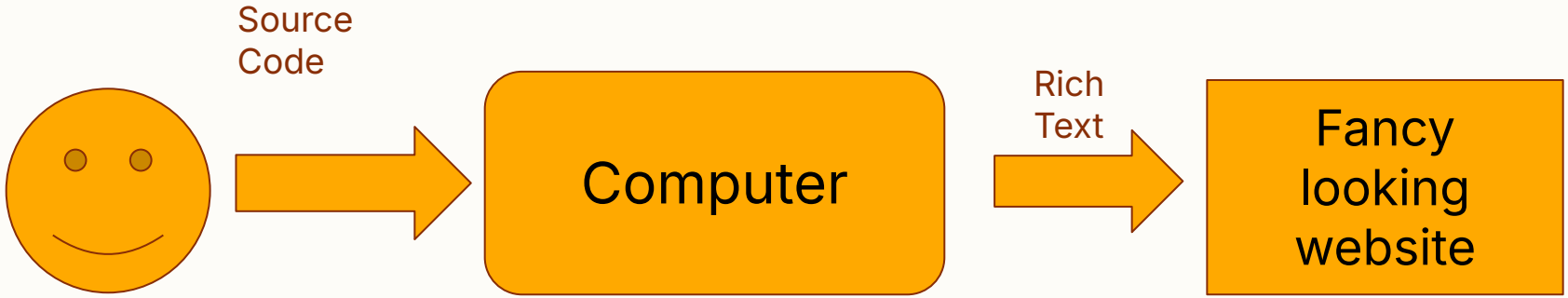
- Plaintext
  - As opposed to rich text

- Computers are very specific
  - We need a simple, concrete way of giving it text
  - Code

HTML

- We provide the source code (plaintext)

- Computer “compiles” out code
- Software abstraction



HTML

- Give the computer plaintext
- It gives us rich text
- Plaintext
  - Just plain ASCII (unicode) character
- We need a special plaintext editor for this
- Cannot use Word

HTML

- Use any of these
- Most full-time devs use VSCode or Vim

- We will support **Notepad++** for the labs

HTML

- To create a new HTML document
- We have to tell the browser what we are trying to do

- “This is a web page that has the following content”

HTML

- We need all of this
- Into our HTML file before we can start
- Boilerplate
- Bear with me for a minute

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML

- HTML works by defining a series of “tags”
- Usually between angle brackets
- Start with **<TAG>**
- End with **</TAG>**
- Content goes in between
- <h1> Heading </h1>

HTML

- Let’s go back and re examine all that boilerplate
  - Some rules have exceptions
  - Head follows those rules
  - Things can go inside inside of those other elements
- Head is primarily “metadata”
- Doesn’t actually render anything

HTML

- Can open the file in your browser to see what it looks like
- There is nothing here yet

- Let’s add something
- Use HTML standard tags
- Make sure to save your file

HTML

- Heading (1)
- Largest heading
- **<h1> Your heading here </h1>**

- Will render as a large heading text
- Use sparingly
- Put this in the **BODY** of your HTML

HTML

- You can make headings smaller
- H1 ⇒ Biggest heading
- **<h2> My heading</h2>** ⇒ smaller heading
- Can repeat with bigger values

- There is a limit ⇒ Let’s find it

HTML

- Paragraph
- Let’s you give a large bulk of paragraph text

- ** Your text here **
- Do this in the body
- Outside of any headers

HTML

- Line breaks
- Provide a line break between elements
- Does NOT need a closing tag

-
- Let’s see that
- Paragraph did the breaking for us

HTML

- Anchor Tag
- Hyperlink!
- Has an “attribute” ⇒**href**

- **Link text**
- Provide links to other web pages
- Anchors you to the “net”

HTML

- Images
- Render images to your website
- Does not need a closes
- Two Attributes
  - src= Image source
  - alt= Alternative image text
- Accessibility (+ semantic web)

HTML

- Your own image
  - Put the image in the same folder as your html file
- Someone else images
  - Get the link to **JUST** the image

```
<img src="my-image.png" alt="description">
<img src="[link-to-image]" alt="description">
```

Nested Components

Reese Hatfield

HTML

- So far, we've just seen "top level" components
- Secretly nestled within a body

- But web apps are more than just a bunch of linear elements

HTML

- Lists
- Two kinds of lists in HTML

- Ordered (numbered)
- Un-Ordered (bullets)

HTML

- Unordered Lists
- ** [list content] **

- Need to specify the list content
- Organize with list items
- ** item content **

HTML

- Ordered Lists
- ** [list content] **

- Need to specify the list content
- Organize with list items
- ** item content **
- Do the numbering for you

HTML

- Ordered Lists
- ** [list content] **

- Also support a type attribute
- **type=[a, A, i, I]**
- Optionally also a start attribute
- **start=[c]**

HTML

- Want formatted text
- Within other elements
- Good start would be
  - Bold
  - Italics
  - Etc.
- Remember, we only have plaintext

HTML

- Bold
- **Two ways**
- ** Important **
  - Strong importance
  - Accessibility
- ** Bold text **
  - Purely visual

HTML

- Italics
- *Two ways*
- ** Important **
  - Emphasis or stress
  - Accessibility
- ** Bold text **
  - Purely visual

HTML

- Underlined
- Text
- Only one way*
- **<u> Text to be underlined </u>**
  - Underlines the contained text
  - Generally not used as often
  - Other ways are preferred
    - Styled text

HTML

- Strikethrough
- ~~Text~~
- **<s> Changed text </s>**
  - No longer accurate
- **<del> Deleted text </del>**
  - Other ways are preferred
  - Align with semantic meaning

HTML

- Can get us pretty much all purely textual formatting
- Can nest these together to create more complex text formatting
- Let’s see that

HTML

- Web is just a bunch of linked documents
    - See anchor tags
    - How else can be connect with other pages?
    - Iframe!

HTML

- IFrame = a “portal” to another website
- **<iframe>** tag
- Used like
- Has a src attribute
  - Similar to img
- Show other websites within yours

HTML

- Syntax:
  <iframe src="url here">
  Fallback text if iframe is unsupported
  </iframe>
- Some websites don't allow iframe
- Some encourage them!
  - Youtube ⇒ share⇒embed

HTML

- Tables
- Tables are significantly more complex to represent as plaintext

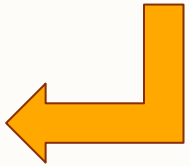
- Let’s look at the anatomy of a table

HTML

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

HTML

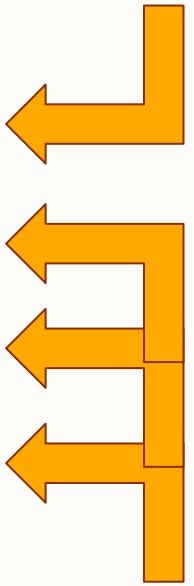
Header



Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

HTML

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology



HTML

- Table header and table data are fundamentally different
- Header defines *what* the data actually means

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

HTML

- When we are encoding a table into plaintext
- We need to consider this separation

- Let’s start with a basic declaration

HTML

- Declare tables with
- **<table> ... </table>**
- How do we load data into that table?

- Well we first need to define the header for our data
- Acts like a schema for the data

HTML

- Each row in our data has a meaning
- Define a table row as a **<tr> ... </tr>**

<table>

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

</table

HTML

- Each row in our data has a meaning
- Define a table row as a **<tr> … </tr>**

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

<table>

<tr> …. </tr>

<tr>…. </tr>

<tr>….</tr>

<tr>….</tr>

</table

HTML

- For our header row
  - Special element for table header
  - <th> HEADER </th>

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

```
<table>

<tr>
  <th> Class </th>
  <th> Instructor </th>
  ...
</tr>

...

</table
```

HTML

- For our data rows
  - Different HTML tag (table data)
  - <td> HEADER </td>

Class	Instructor	Level	Department
Intro-To-CS	Reese Hatfield	1150	Computer Science
Discrete-Math	Ron Taylor	2200	Math
Microbiology	Stacey Hundley	3120	Biology

```
<table>

<tr>
  <th> Class </th>
  <th> Instructor </th>
  ...
</tr>
<tr>
  <td> MicroBio </td>
  <td> Stacey H. </td>
</tr>
</table
```

HTML

- Let’s implement the full thing
- Using
  - **<table>**
  - **<tr>**
  - **<th>**
  - **<td>**

HTML

- Tables can get significantly more complicated than what we will have time to cover
  - “colspan” and “rowspan” attributes
  - Etc.
- Should be all you need for the lab

HTML

- Last HTML elements we will cover
- Non-semantic element grouping

- We want a way to be able to **div**ide our HTML code out into logical sections
- Right now, it’s all in one group

HTML

- Block Dividers
- ** ... **
- Styles as a vertical block on the page (kinda)
- This is the **most important** HTML element we will cover
  - Useful for styling your page

HTML

- Inline Dividers
- ** ... **
- Styles as a inline elements (kinda)

- Not used as much as block level divisions
- Still very common

HTML

- There are *many* more HTML elements
- We only covered the most common ones
- Web Development explores these ideas more thoroughly
- Let’s peek at some of the weird ones