

IN4MATX 285:

Interactive Technology Studio

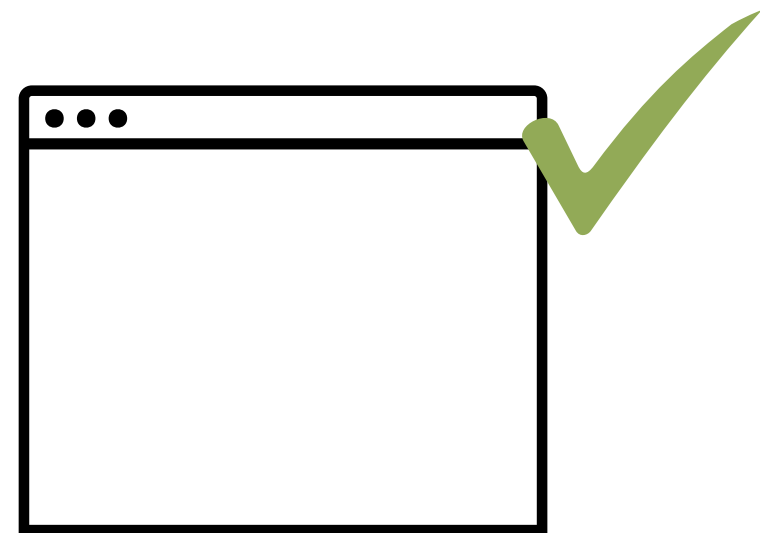
Programming: HTML

Today's goals

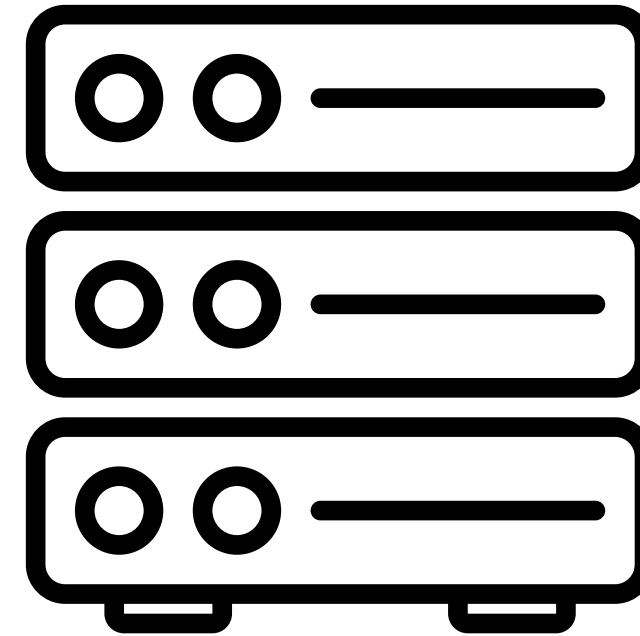
By the end of today, you should be able to...

- Describe the fundamentals of web communication
- Identify the syntax of HTML tags and attributes and describe their roles
- Create basic HTML content, like lists and headings
- Use a code editor to create HTML

Client-side web development

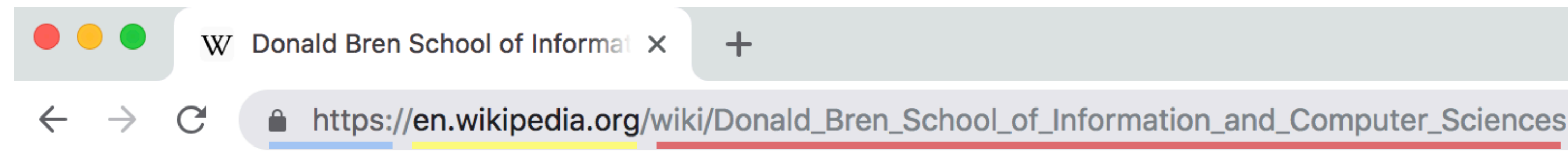


Your browser



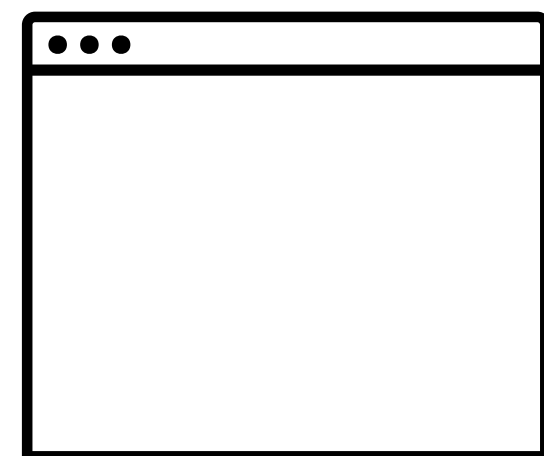
Web server

Using the internet



Protocol	Host	Resource
(how to handle info)	(who has info)	(what info you want)

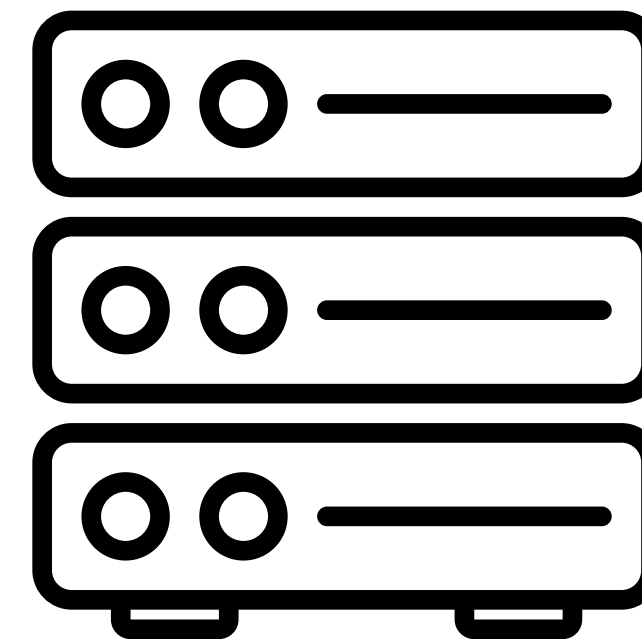
“Hey Wikipedia, I’d like to see the page for the school of ICS!”



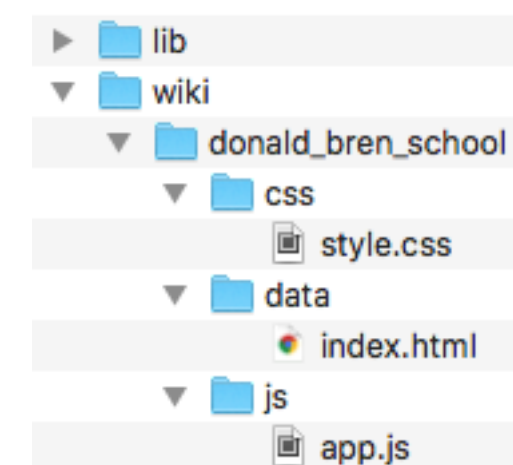
Your browser

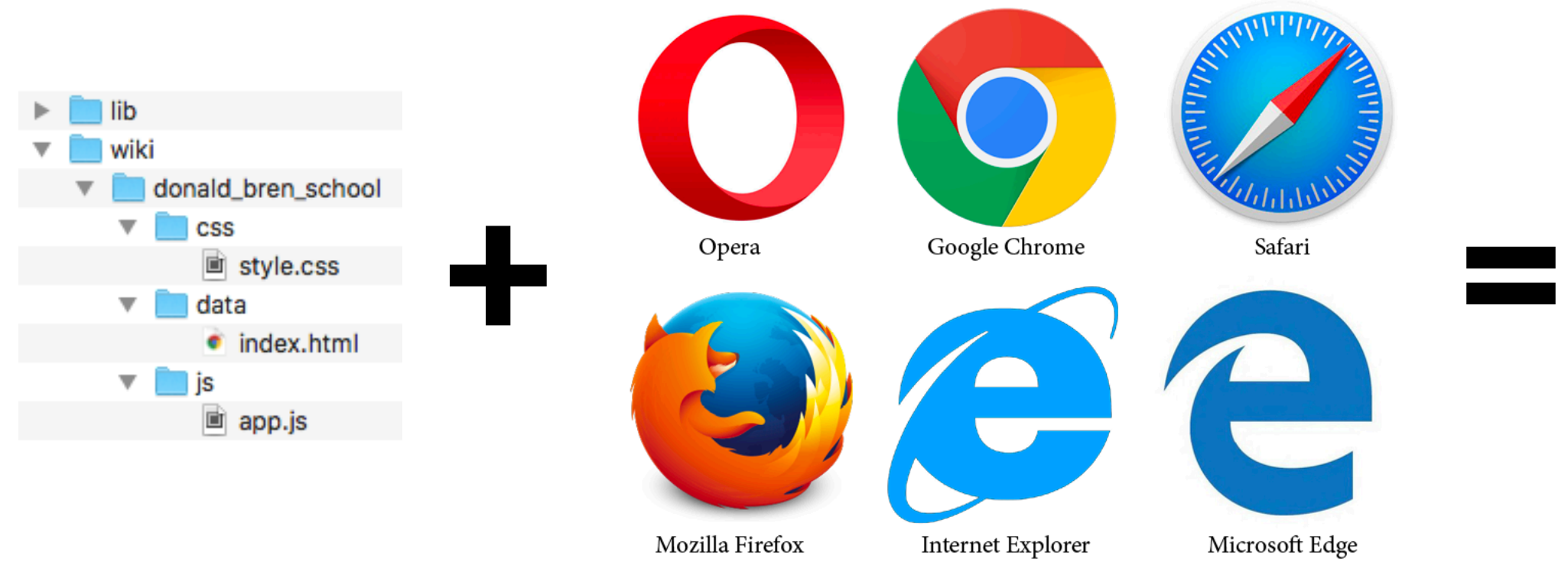
Request

Response



Web server





W Donald Bren School of Informa x +

← → ↻ 🔒 https://en.wikipedia.org/wiki/Donald_Bren_School_of_Information_and_Computer_Sciences ☆

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

Wiki Loves Monuments: The world's largest photography competition is now open! Photograph a historic site, learn more about our history, and win prizes.

Donald Bren School of Information and Computer Sciences

From Wikipedia, the free encyclopedia Coordinates: 33.6432°N 117.842°W

This article has multiple issues. Please help [improve it](#) or discuss these issues on the [talk page](#). *(Learn how and when to remove these template messages)*

- This article **contains content that is written like an advertisement**. *(April 2016)*
- This article **may rely excessively on sources too closely associated with the subject**, potentially preventing the article from being **verifiable** and **neutral**. *(January 2015)*

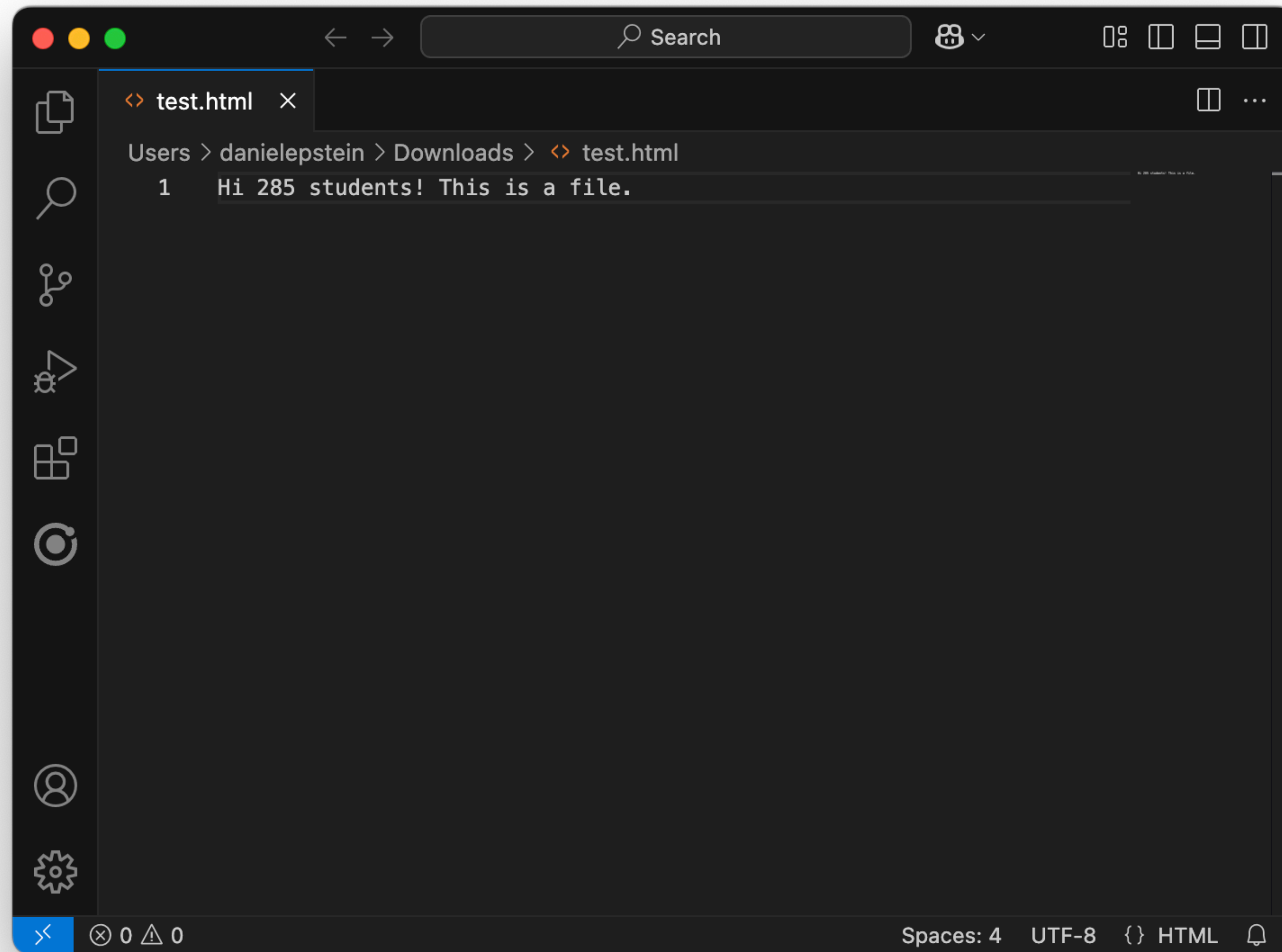
The **Donald Bren School of Information and Computer Sciences**, also known colloquially as UCI's **School of ICS** or simply the **Bren School**, is an academic unit of **University of California, Irvine** (UCI), and the only dedicated school of **computer science** in the **University of California** system. Consisting of nearly three thousand students, faculty, and staff,^[2] the school maintains three buildings in the South-East artery of UCI's undergraduate campus, and maintains student body and research affiliations throughout UCI.^{[3][4]}

The school of ICS consists of three departments: Computer Science, **Informatics**, and **Statistics**. The combined groupings focus the school around the fields of **computing** and processing of information. The departments confer eight undergraduate, eleven masters, and seven doctoral degrees in total, with some degree programs cooperating with affiliated schools.^[5]

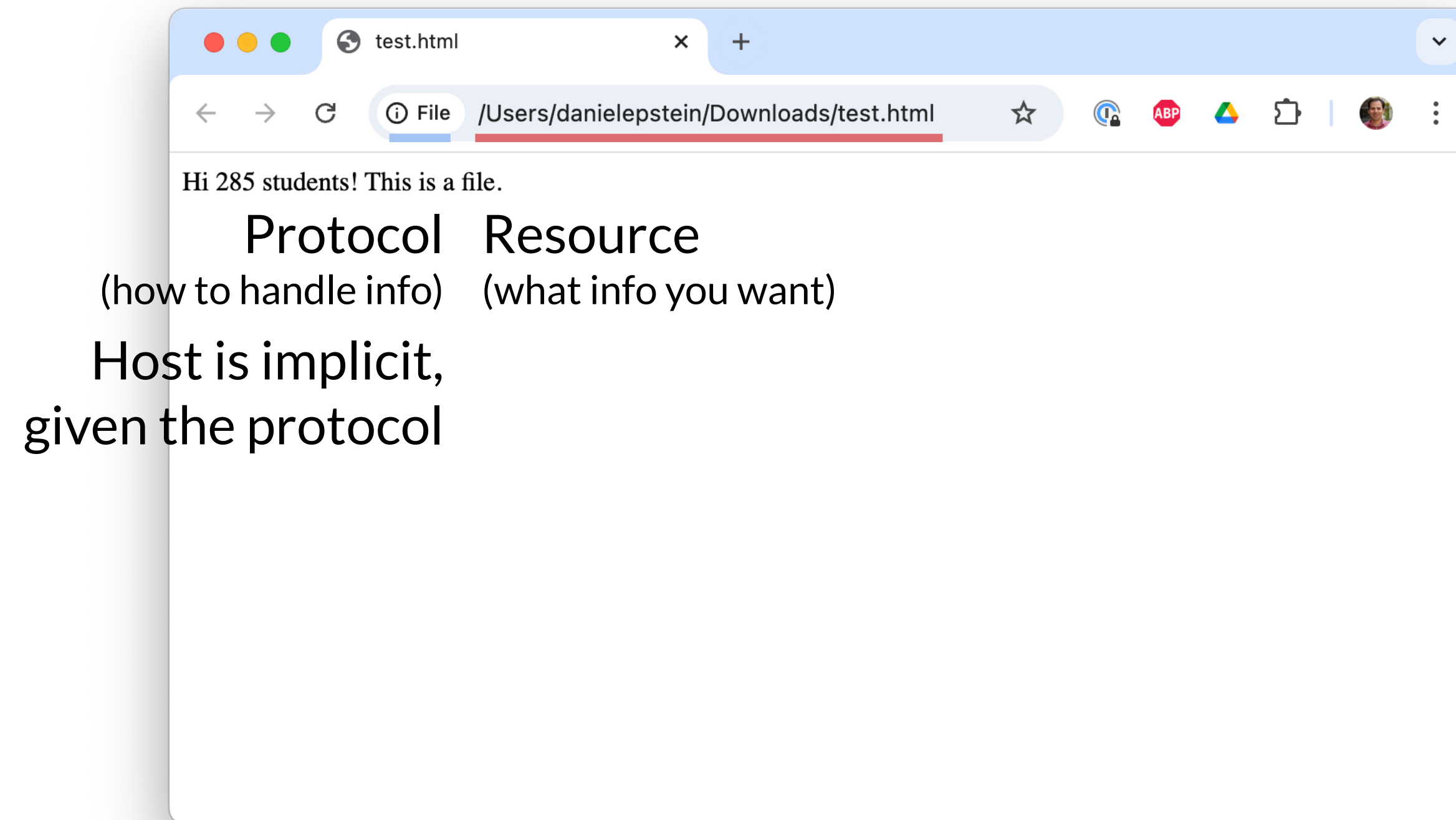
Donald Bren Hall, one of the buildings on the campus of the Bren School^[1]

**Fundamentally, the web is
designed to send files around**

So what does a file on the web look like?



Same file, opened in Chrome



**What if we wanted to specify
how the content is rendered?**

HTML (HyperText Markup Language)

- Adds meaning to text
- Links documents to one another
 - Vannevar Bush, hypertext vision



Tags

`<div>` ← Open/start tag

Content goes here. ← Content

`</div>` ← Close/end tag

Whitespace and tag case are ignored

Some common tags

`<h1>`Heading level `1</h1>`

`<h2>`Heading level `2</h2>`

...

`<p>`A paragraph`</p>`

`<!--A comment-->`

`` An image

`` An unordered list (bullets)

`` A list item

`<table>` A data table

`` Important content (**bolded**)

`` Emphasized content (*italicized*)

`<div>` A division (section) of content

Tags

- There are hundreds of tags!
- You may not use them all, but it's good to explore them
- Search on Google or W3C to understand each tag's purpose
- <https://www.w3schools.com/tags/>

HTML 5 NEW TAG		TAG NOT SUPPORTED IN HTML 5	
<!--...-->	Define a comment	<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink href, hreflang, media, ping, rel, target, type	<abbr>	Defines an abbreviation
<acronym>	Used to define an embedded acronym	<address>	Defines an address element
<applet>	Used to define an embedded applet	<area>	Defines an area inside an image map alt, coords, href, hreflang, media, ping, rel, shape, target, type
<article>	Defines an article cite, pubdate	<aside>	Defines content aside from the page content
<audio>	Defines sound content autobuffer, autoplay, controls, src		Defines bold text
<base>	Defines a base URL for all the links in a page href, target	<basefont>	Used to define a default font-color, font-size, or font-family for all the document
<bdo>	Defines the direction of text display dir	<big>	Used to make text bigger
<blockquote>	Defines a long quotation cite	<body>	Defines the body element
 	Inserts a single line break	<button>	Defines a push button autofocus, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value
<canvas>	Defines graphics height, width	<caption>	Defines a table caption
<center>	Used to center align text and content	<cite>	Defines a citation
<code>	Defines computer code text autobuffer, autoplay, controls, src	<col>	Defines attributes for table columns
<colgroup>	Defines groups of table columns span	<command>	Defines a command button checked, disabled, icon, label, radiogroup, type
<datalist>	Defines a dropdown list	<dd>	Defines a definition description
	Defines deleted text cite, datetime	<details>	Defines details of an element open
<dialog>	Defines a dialog (conversation)	<dfn>	Defines a definition term
<dir>	Used to define a directory list	<div>	Defines a section in a document
<dl>	Defines a definition list	<dt>	Defines a definition term
	Defines emphasized text	<embed>	Defines external interactive content or plugin height, src, type, width
<fieldset>	Defines a fieldset disabled, form, name	<figure>	Defines a group of media content, and their caption
	Used to define font face, font size, and font color of text	<footer>	Defines a footer for a section or page
<form>	Defines a form accept-charset, action, autocomplete, enctype, method, name, novalidate, target	<frame>	Used to define one particular window (frame) within a frameset
<frameset>	Used to define a frameset, which organized multiple windows (frames)	<h1> to <h6>	Defines header 1 to header 6
<h1>	Defines header 1 to header 6	<head>	Defines information about the document
<header>	Defines a header for a section or page	<hgroup>	Defines information about a section in a document
<hr>	Defines a horizontal rule	<html>	Defines an html document manifest, xmlns
<i>	Defines italic text	<iframe>	Defines an inline sub window height, name, sandbox, seamless, src, width
	Defines an image alt, src, height, ismap, usemap, width	<input>	Defines an input field accept, alt, autocomplete, autofocus, checked, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, step, type, value, width
<ins>	Defines inserted text cite, datetime	<keygen>	Defines a generated key in a form autofocus, challenge, disabled, form, keytype, name
<kbd>	Defines keyboard text	<label>	Defines an inline sub window for, form
<legend>	Defines a title in a fieldset		Defines a list item value
<link>	Defines a resource reference href, hreflang, media, rel, sizes, type	<map>	Defines an image map name
<mark>	Defines marked text	<menu>	Defines a menu list label, type
<meta>	Defines meta information charset, content, http-equiv, name	<meter>	Defines measurement within a predefined range high, low, max, min, optimum, value
<nav>	Defines navigation links	<noframes>	Used to display text for browsers that do not handle frames
<noscript>	Defines a noscript section	<object>	Defines an embedded object data, form, height, name, type, usemap, width
	Defines an ordered list reversed, start	<optgroup>	Defines an option group label, disabled
<option>	Defines an option in a drop-down list disabled, label, selected, value	<output>	Defines some types of output for, form, name
<p>	Defines a paragraph	<param>	Defines a parameter for an object name, value
<pre>	Defines preformatted text	<progress>	Defines progress of a task of any kind max, value
<q>	Defines a short quotation cite	<rp>	Used in ruby annotations to define what to show browsers that do not support the ruby element
<rt>	Defines explanation to ruby annotations	<ruby>	Defines ruby annotations
<s>, <strike>	Used to define strikethrough text.	<samp>	Defines sample computer code
<script>	Defines a definition list async, type charset defer, src	<section>	Defines a section cite
<select>	Defines a selectable list autofocus, disabled, form, multiple, name, size	<small>	Defines small text
<source>	Defines media resources media, src, type		Defines a section in a document
	Defines strong text	<style>	Defines a style definition type, media, scoped
<sub>, <sup>	Defines sub/super-scripted text	<table>	Defines a table summary
<tbody>	Defines a table body summary	<td>	Defines a table cell colspan, headers, rowspan
<td>	Defines a table cell colspan, headers, rowspan	<textarea>	Defines a text area autofocus, cols, disabled, form, maxlength, name, placeholder, readonly, required, rows, wrap
<tfoot>, <thead>	Defines a table footer / head	<th>	Defines a table header colspan, headers, rowspan, scope
<time>	Defines a date/tim datetime	<title>	Defines the document title
<tr>	Defines a table row datetime	<tt>	Used to define teletype text
<u>	Used to define underlined text		Defines an unordered list
<var>	Defines a variable	<video>	Defines a video autobuffer, autoplay, controls, height, loop, src, width

HTML5 TAG CHEAT SHEET
Created by WebsiteSetup.org

Nesting

- The **Content** of a tag can contain other HTML tags

```
<div><p>I <strong>love</strong> HTML!</p></div>
```


Nesting: lists

```
<h2>Professor Epstein's shopping list:</h2>
```

```
<ul>
```

```
<li>Milk</li>
```

```
<li>Eggs</li>
```

```
<li>Sandwich ingredients:
```

```
<!--Lists can be nested!-->
```

```
<ul>
```

```
<li>Bread</li>
```

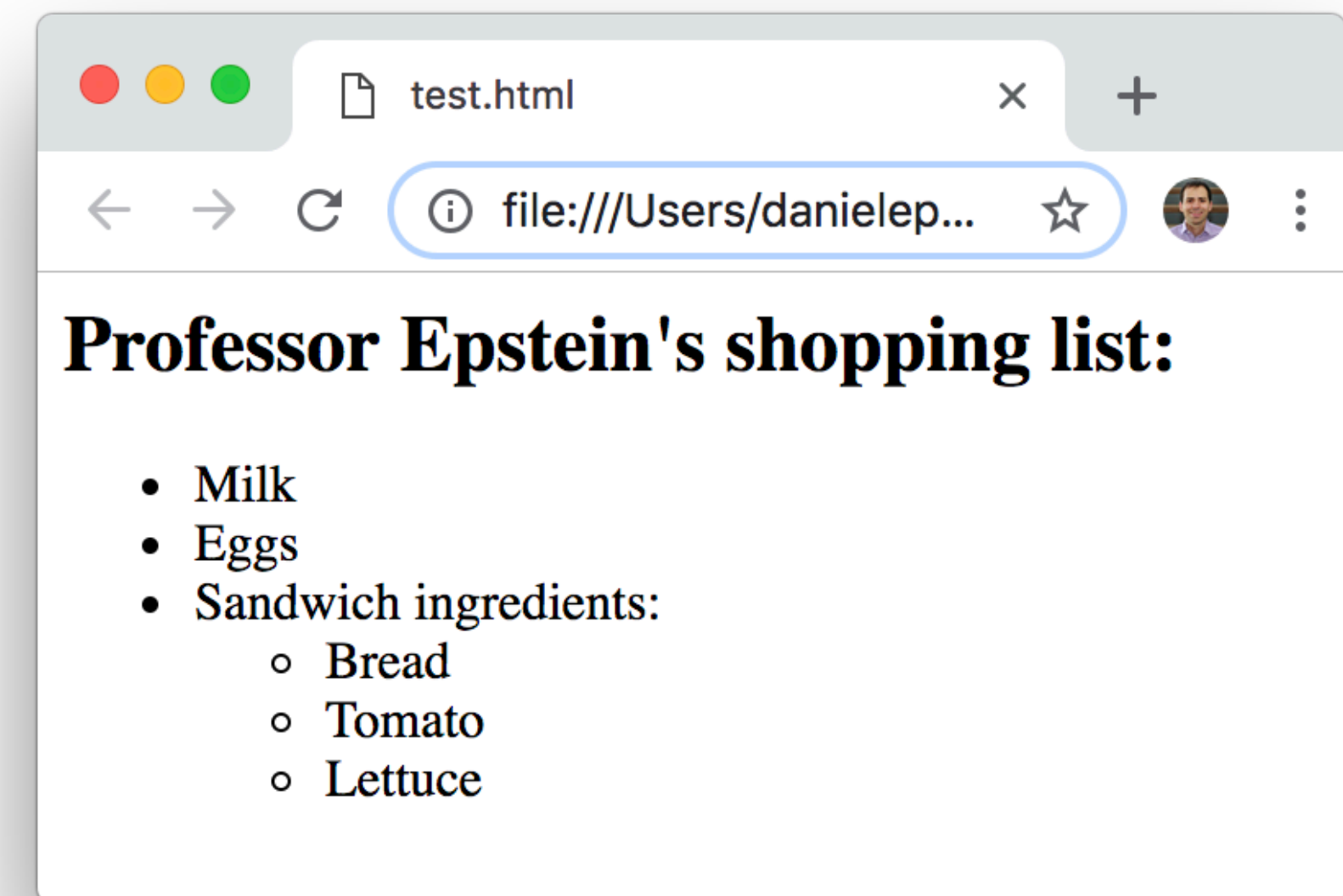
```
<li>Tomato</li>
```

```
<li>Lettuce</li>
```

```
</ul>
```

```
</li>
```

```
</ul>
```



Nesting: HTML

- By convention, HTML is specified via the **Content** of an `<html>` element.

`<html>` ← Start of HTML document
 `<body>` ← Start of body (visible) content
 `<h1>`Hello, IN4MATX 285!`</h1>`
 `<p>`HTML is ``great``!`</p>`
 `</body>` ← End of body content
`</html>` ← End of HTML document

Attributes

- Attributes specify options and add meaning
- Attributes are space-separated lists of names and values.
 - Kind of like variables
 - Almost always Strings

```
<div attributeA="valueA" attributeB="valueB">  
  Content goes here  
</div>
```

Attributes

```
<a href="http://depstein.net/">Daniel Epstein</a>
```



anchor
(hyperlink)



hypertext
reference

```

```



source



alternative text
for screen readers



img tags have no (text)
content, so no closing tag

```
<html lang="en">
```

...

```
</html>
```

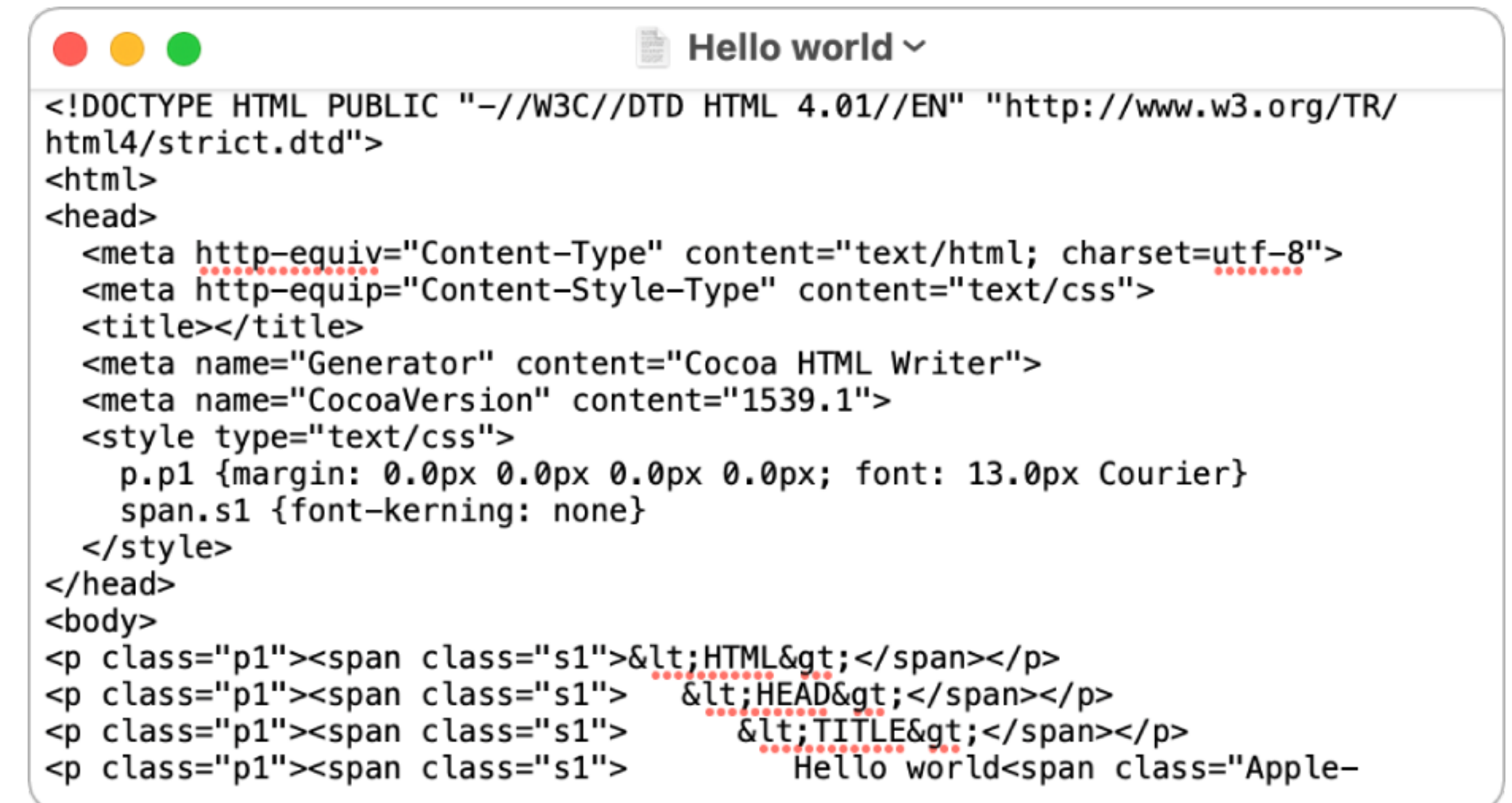


Language of document
is English

So, how should I go about creating an HTML file?

Code editors

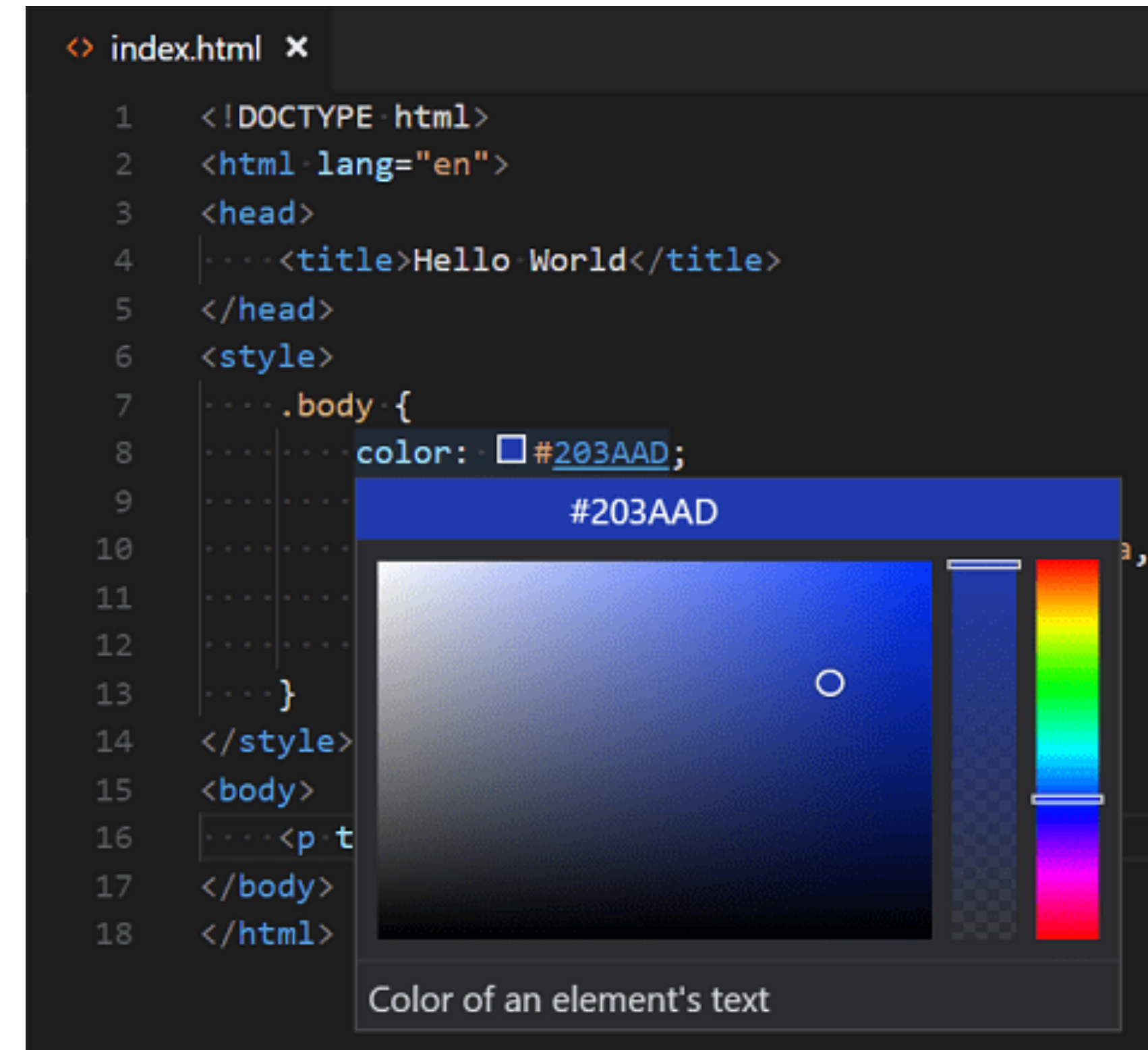
- Since HTML is just text, you can use any text editor to edit HTML
 - TextEdit on Mac, Notepad on Windows
- But, doing so gets annoying very quickly
 - Lack of formatting
 - Lots of repetitive typing (open tags, close tags)



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/
html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta http-equiv="Content-Style-Type" content="text/css">
  <title></title>
  <meta name="Generator" content="Cocoa HTML Writer">
  <meta name="CocoaVersion" content="1539.1">
  <style type="text/css">
    p.p1 {margin: 0.0px 0.0px 0.0px 0.0px; font: 13.0px Courier}
    span.s1 {font-kerning: none}
  </style>
</head>
<body>
<p class="p1"><span class="s1">&lt;HTML&gt;</span></p>
<p class="p1"><span class="s1">    &lt;HEAD&gt;</span></p>
<p class="p1"><span class="s1">        &lt;TITLE&gt;</span></p>
<p class="p1"><span class="s1">            Hello world<span class="Apple-
```

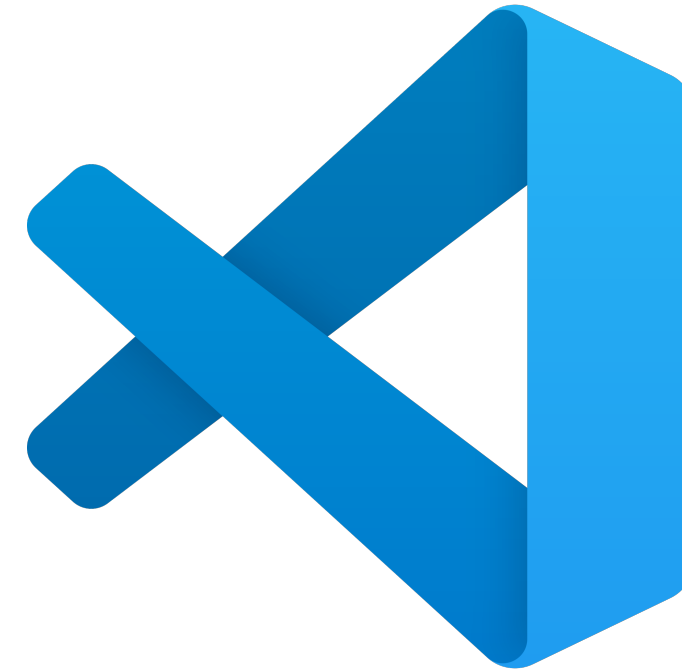
Code editors

- Instead, programmers use an Integrated Development Environment (IDE)
- IDEs provide:
 - Tools for checking syntax
 - Formatting assistance
 - Automatic text completion
 - Tools for compiling and debugging, sometimes



Code editors

- Widely-used IDEs
 - Visual Studio Code (Microsoft)
 - WebStorm (JetBrains)
 - Eclipse (Eclipse Foundation)
- If you're new to Web Development, we recommend using VS Code



Today's goals

By the end of today, you should be able to...

- Describe the fundamentals of web communication
- Identify the syntax of HTML tags and attributes and describe their roles
- Create basic HTML content, like lists and headings
- Use a code editor to create HTML

Next programming skill: styling with CSS

INF 285 - Interactive Technology Studio

[Home](#) | [Assignments](#) | [Syllabus](#) | [Resources](#) | [Calendar](#)

Assignments

- A1: Widget Factory in Adobe Spectrum
 - [Design Mockup \(Due April 5, 2025 at 12:59 AM\)](#)
 - [Code Check-in \(Due April 12, 2025 at 12:59 AM\)](#)
 - [Final submission \(Due April 19, 2025 at 12:59 AM\)](#)
- A2: Enrollment Guide in JavaScript
 - Hello, World! (Due April 26, 2025 at 12:59 AM)
 - Schedule Summarization (Due May 3, 2025 at 12:59 AM)
 - Final Submission (Due May 10, 2025 at 12:59 AM)
- A3: Accessibility Browser Extension
 - Code Check-in (Due May 24, 2025 at 12:59 AM)
 - Final Submission (Due June 7, 2025 at 12:59 AM)

Copyright © 2025 by Daniel Epstein.

This course is licensed under a [Creative Commons Attribution Non-Commercial](#) licen

INF 285 - Interactive Technology Studio - Spring 2025

Assignments

- A1: Widget Factory in Adobe Spectrum
 - [Design Mockup \(Due April 5, 2025 at 12:59 AM\)](#)
 - [Code Check-in \(Due April 12, 2025 at 12:59 AM\)](#)
 - [Final submission \(Due April 19, 2025 at 12:59 AM\)](#)
- A2: Enrollment Guide in JavaScript
 - Hello, World! (Due April 26, 2025 at 12:59 AM)
 - Schedule Summarization (Due May 3, 2025 at 12:59 AM)
 - Final Submission (Due May 10, 2025 at 12:59 AM)
- A3: Accessibility Browser Extension
 - Code Check-in (Due May 24, 2025 at 12:59 AM)
 - Final Submission (Due June 7, 2025 at 12:59 AM)

CSS



IN4MATX 285:

Interactive Technology Studio

Programming: HTML