Your Own Web Site with HTML, CSS & JavaScript

Part I - GitHub, VSCode and HTML 5

Professor Stephen Gilbert

Where We're Going

- This is going to be a 3-part, hands-on series of presentations
  - You don't need to be a CS Major or have any programming experience
  - Goal is to give you practical tips you can use whatever your major
  - Every tool and Web site we'll use today is completely free
- In part 1 (today) we'll cover HTML
  - Sign up for a GitHub account to store your Web site
  - Copy the repository of starter pages that you will work on
  - Learn how to use Visual Studio Code online with CodeSpaces
  - Learn some HTML basics: pages, links, images, tables and forms
  - Get some resources for learning more
- In Part 2 and Part 3 you'll add CSS and JavaScript to your Web site

The Three Pillars of Web Design

What's the Difference?

HTML — Hypertext Markup Language

Create the structure
· Controls the layout of the content
· Provides structure for the web page design
· The fundamental building block of any web page

CSS — Cascading Style Sheet

Stylize the website
· Applies style to the web page elements
· Targets various screen sizes to make web pages responsive
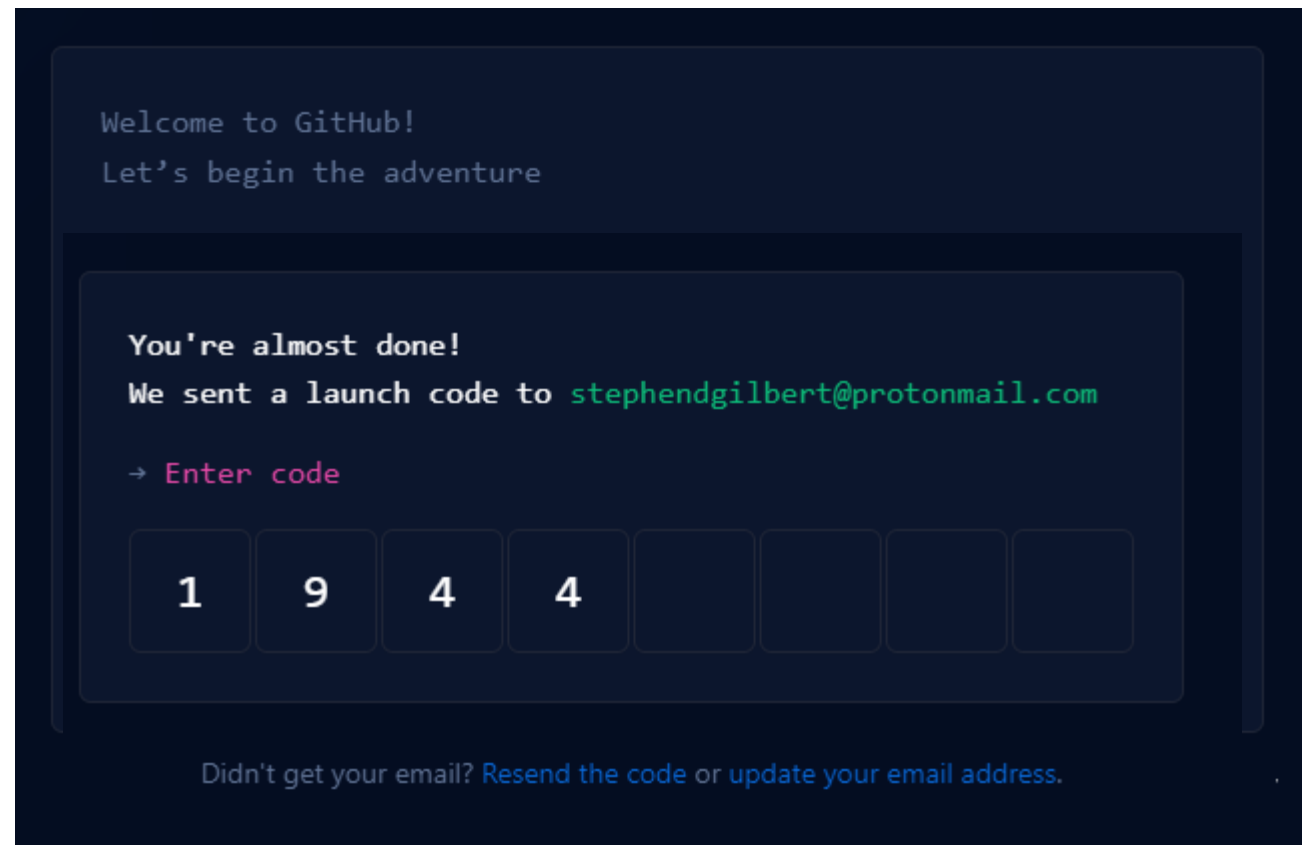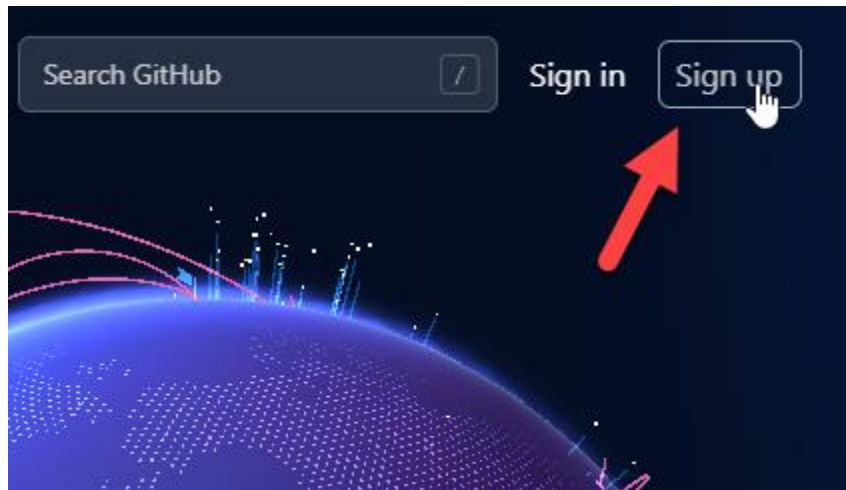· Primarily handles the "look and feel" of a web page

Javascript

Increase interactivity
· Adds interactivity to a web page
· Handles complex functions and features
· Programmatic code which enhances functionality

Step 1 – Create a GitHub Account

▸ Where you'll store your code as you work on it

  ▸ Will also be where your Web Site will live. github.com

Step 2 - Duplicate the Starter Code

▸ The starter code is at: github.com/occ.computer.science/web

  ▸ Click the **Fork** button to get your own copy of the code

Step 3 - Activate GitHub Pages for your Site

▸ Click Settings at the top of the page

  ▸ Choose Pages on Setting page

  ▸ Choose Main branch, docs folder, Save
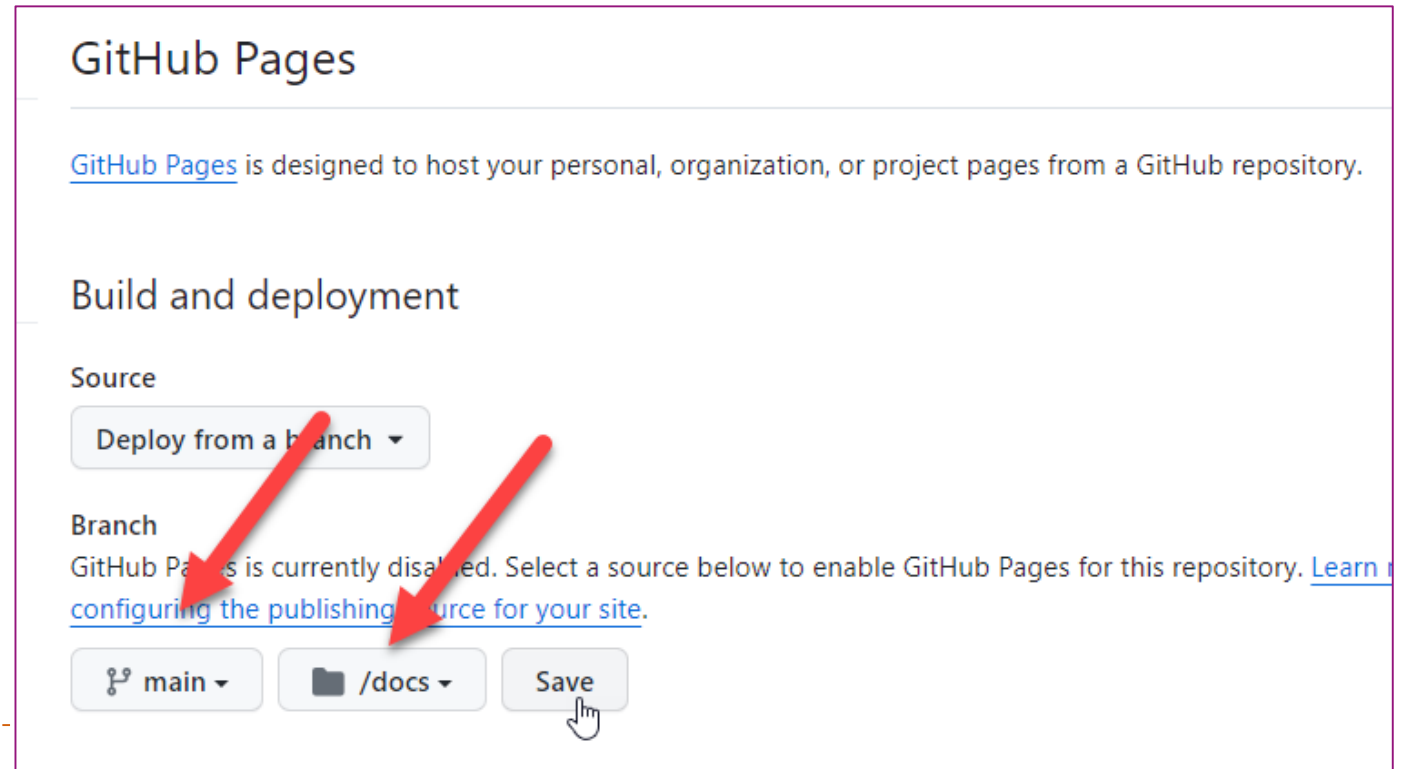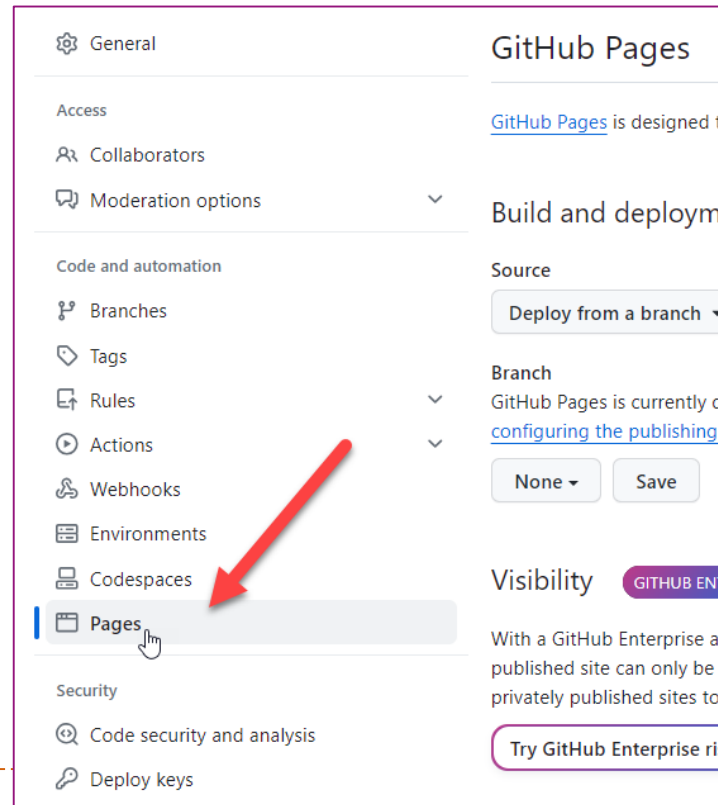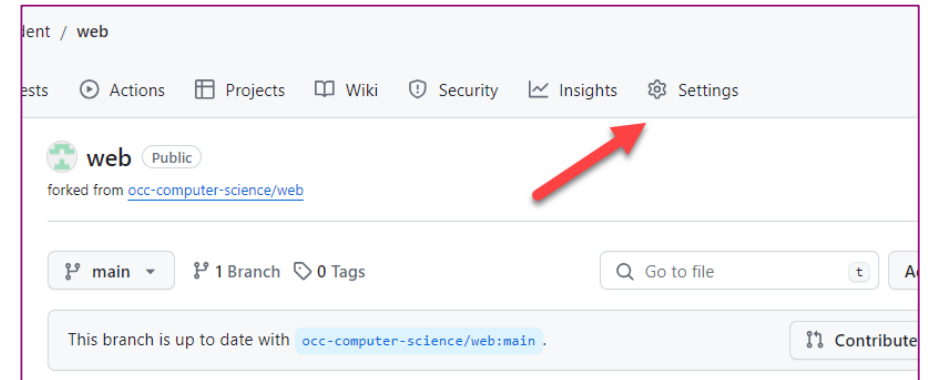
Step 4 - Open the VS Code IDE (CodeSpaces)

▸ Click the drop-down arrow on the green Code button

  ▸ Choose the CodeSpaces tab and click the green Create button
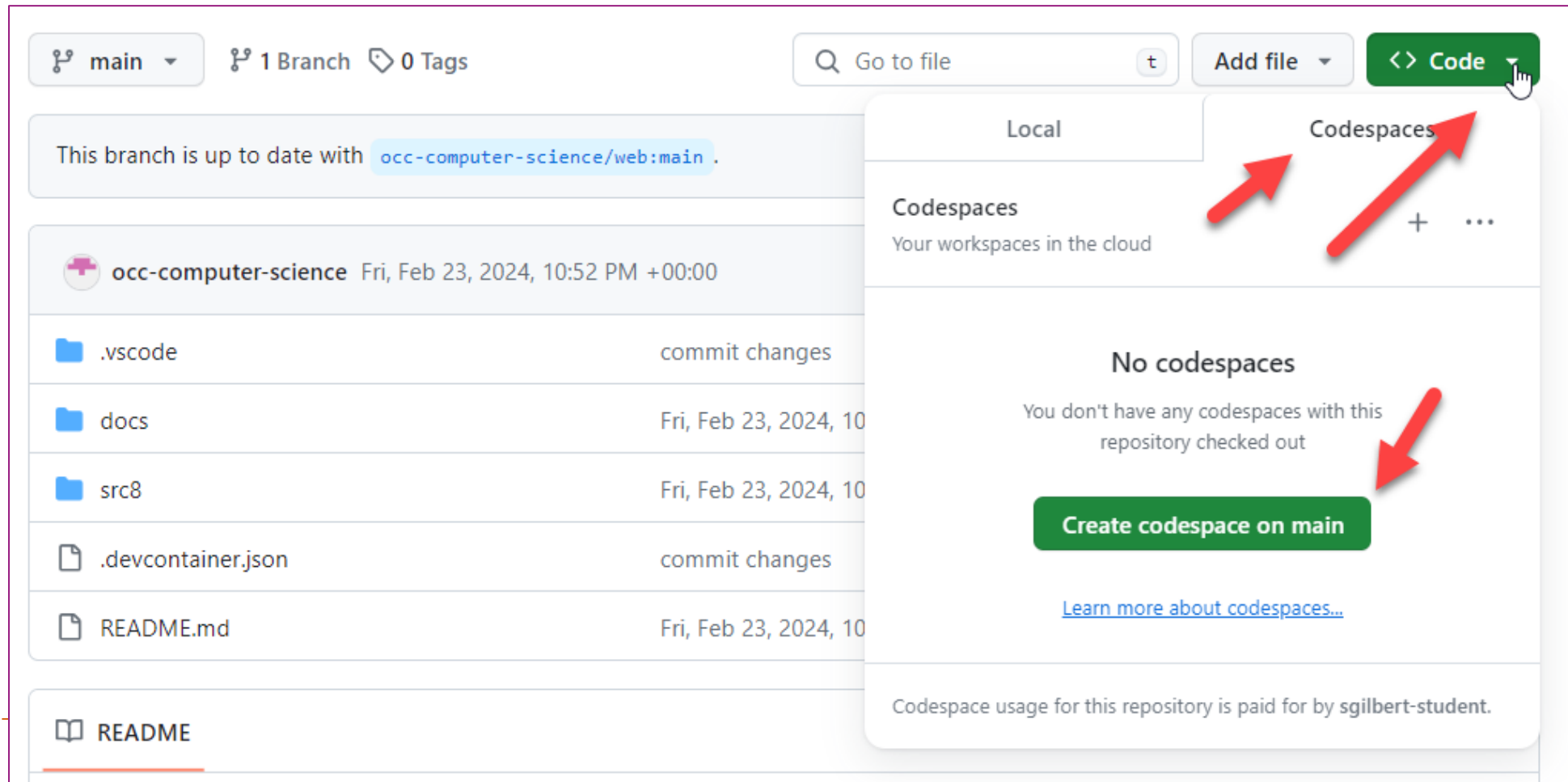
Step 5 - Your VSCode CodeSpace

5-Mar-24
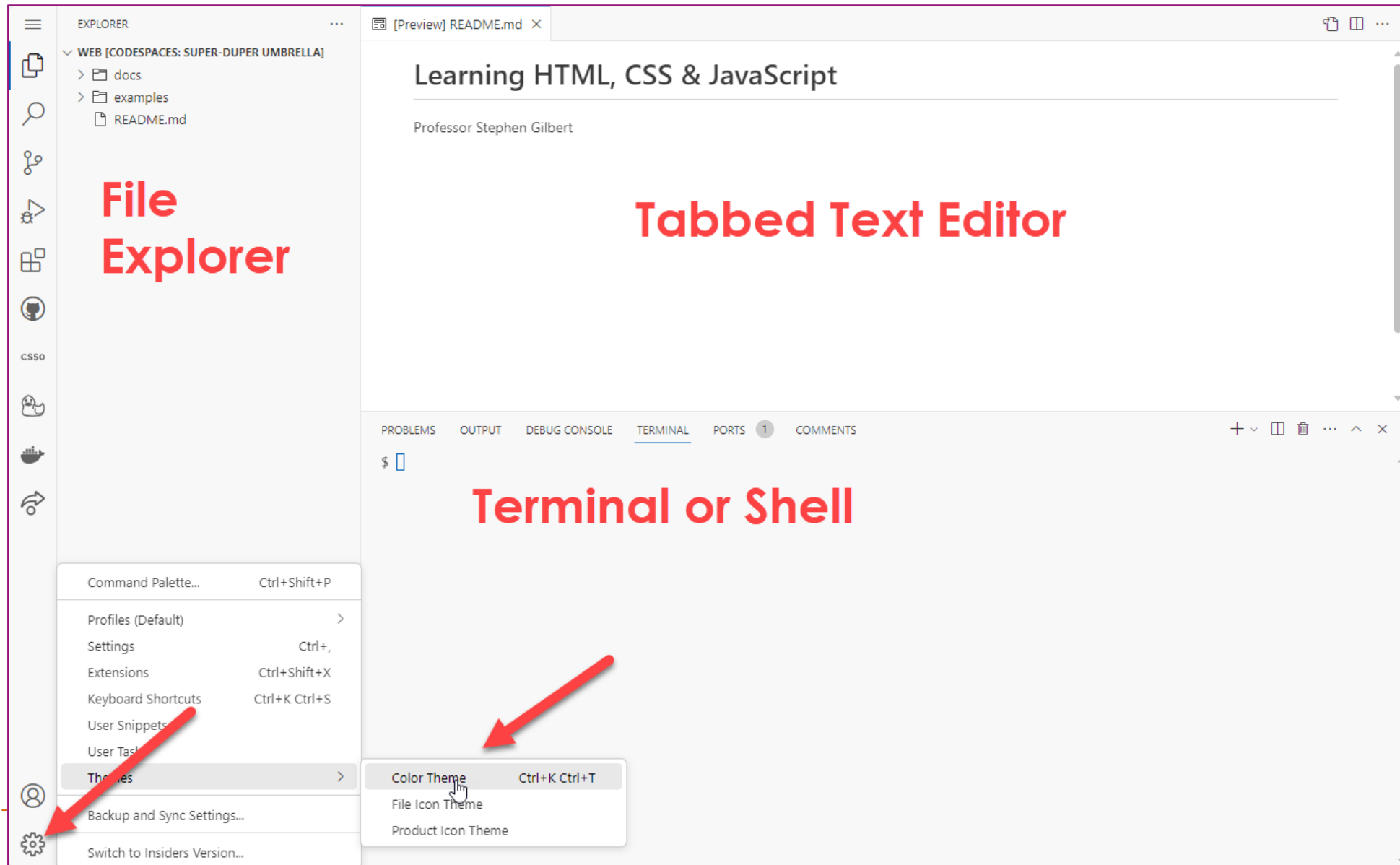
Creating Web Pages

▸ Let's create a group of linked Web pages

  ▸ Right click the `html-tutorial` folder

  ▸ Select Open in integrated terminal

  ▸ Type `http-server` and press Enter

  ▸ When the server starts, click Open URL

  ▸ Index listing. Click `index.html`

The Curbside Thai Project

- This project is borrowed from CIS 123 here at OCC
- Web site for a Thai restaurant
  - "Landing" page is complete (`index.html`)
  - Some of the links are "dead"
  - Others go to incomplete pages
  - We'll complete the missing pieces
- The copy has already been written, so we'll just focus on HTML

Curbside Thai • 411 Belde Drive, Charlotte NC 28201 • 704-555-1151

Your First Web Page

▸ Open another terminal window

▸ In the terminal, type
  `code about.html` and press Enter

▸ In the editor, type `html` and
  press `html:5` + Control-space

▸ Choose and VSCode
  will write the HTML "boilerplate"

▸ Between the `<title>` tags
  add `About Curbside Thai`

```
 about.html  X

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" cont
6      <title>About Curbside Thai
7  </head>
8  <body>
9
10  </body>
11  </html>
12  |
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TE

```
html-tutorial/ $ code about.html
html-tutorial/ $ ▯
```

Understanding HTML Tags

- Web pages are written in Hypertext Markup Language or HTML
- HTML pages have two concepts: tags and attributes
    - `<title>About Curbside Thai</title>`
    - `<title>` starts and `</title>` ends the title section
- A tag is like the quotes in this sentence: Johnny said "Hi Mom"
    - Quotes are delimiters that surround the content: Hi Mom
    - They tell you, the reader, that they contain are the words Johnny said
- Similarly, a tag tells the browser about what it contains
    - `<title>` tells the browser "Look out. Here's the title of this Web page."

Understanding HTML Structure and Attributes

▸ Tags are used to give an HTML document a hierarchical structure

  ▸ The `DOCTYPE` describes which version of HTML is used

  ▸ The `<html></html>` tags surround the entire document

  ▸ This tag has an attribute describing the page language, `lang="en"`

  ▸ Attributes consist of keyword (`lang`) and values (`"en"`) (key-value pairs)

▸ Inside the `<html>` tags are two sections, `<head>` and `<body>`

▸ The `<head>` tag contains metadata-information the browser can use

  ▸ `<meta charset=utf-8>` says the page is written in a version of Unicode

  ▸ The next `meta` tag allows your page to respond to different screen sizes

  ▸ The `<title>` is typically used by the browser when displaying the page

  ▸ Note that the `meta` tags don't require a closing `</meta>` tag

Body Sections

- The `<body>` contains the content
- Modern HTML organizes the body using different semantic tags
  - `<header>` and `<footer>`
  - `<main>` main content of page
  - `<article>` for self-contained content
  - `<aside>` for a sidebar
  - `<nav>` for navigation links
- Go ahead and add the structure to your about page

14

```html
 6          <title>About Curbside Thai<
 7 </head>
 8 <body>
 9      <header>

11      </header>
12      <main>
13          <article>

15          </article>
16      </main>
17      <footer>

19      </footer>
20 </body>
21 </html>
22
```

Adding Text to the Body

‣ The text for your page has already been written

    ‣ Open `pages.txt` in a new tab

    ‣ Copy address into `<footer>` section

    ‣ Copy the first section into your `<article>` tag

‣ View the page. Doesn't look very good!

    ‣ HTML reduces all whitespace to a single space

    ‣ Text needs to be inside tags

    ‣ `<p>` around each paragraph

    ‣ `<h1>` around the heading

---

about.html ✕   pages.txt

```
10  |
11      </header>
12      <main>
13          <article>
14              About Us
15
16              Curbside Thai bri
17              North Carolina. M
18              years of experien
19              that same fine di
20              with our modern m
21
22              This is not bland
23              paper boxes! Samp
24              mobile locations
25              11 a.m. to 7 p.m.
26              11 a.m. to 11 p.m
27              difference! If yo
28              Curbside Thai wil
```

All Books (s ✕ | t.s. eliot free ✕ | gutenberg.c ✕ | Your Repos ✕ | sgilbert-stu ✕ | about.html ✕ | About Curb ✕ | should I use ✕ | The main e

https://fuzzy-space-orbit-j47vgr7x79v35q4r-8080.app.github.dev/about.html

About Us Curbside Thai brings the rich flavor of Thailand to North Carolina. Master Chef Sajja Adulet, with over 35 years of experience at the House of Asia, now offers that same fine dining food packaged in greasy paper boxes! Sample his acclaimed cuisine at our various mobile locations throughout downtown Charlotte from 11 a.m. to 7 p.m. on Monday through Thursday, and from your desk, Curbside Thai will deliver. Contact us to cater your next party and experience what Carolina Traveler calls the finest Asian food on or off the streets of Charlotte.

Text-level Elements

- Mark words or phrases
  - `<strong>` (or `<b>`)
  - `<em>` (or `<i>`)
  - `<cite>` and `<q>`
  - `<mark>`
  - `<sub>` or `<sup>`
  - `<code>`, `<kbd>`, `<tt>`
- Add this to your page
  - Bold Curbside Thai in 1st line
  - Emphasize CT will deliver in 2nd
  - Cite Carolina Traveler, and quote

Linking to a Style Sheet

▸ Formatting of your page is done with CSS

  ▸ We'll cover writing CSS in the second lesson

▸ Inside your HTML page you need to link to a style sheet

  ▸ `<link href="styles/base.css" rel="stylesheet">`
    `<link href="styles/layout2.css" rel="stylesheet">`

  ▸ The `href` attribute requires a URL (absolute or relative)

  ▸ I've placed the stylesheets in their own folder

  ▸ The `rel` (relationship) attribute says what kind of link this is

▸ Add these lines to the `<head>` element and view results

```
6    <title>About Curbside Thai</title>
7    <link href="styles/base.css" rel="stylesheet">
8    <link href="styles/layout2.css" rel="stylesheet">
9  </head>
```

-24

Adding Special Characters

Character	Entity	Note
&	&	Interpreted as the beginning of an entity or charact
<	<	Interpreted as the beginning of a tag
>	>	Interpreted as the ending of a tag
"	"	Interpreted as the beginning and end of an attribut
		Interpreted as the non breaking space.
–	–	Interpreted as the en dash (half the width of an em
—	—	Interpreted as the em dash (equal to width of an "n
©	©	Interpreted as the copyright sign.

▸ Not all characters can appear in an HTML page; they are interpreted as HTML syntax

▸ To enter these, use an HTML entity

  ▸ &, <, >,

  ▸ Older syntax: • (avoid)

  ▸ Complete list of named entities

▸ Add a couple bullets and a non-breaking space to the address line

```
</main>
<footer>
    Curbside Thai &bullet; 411 Belde Drive, Charlotte   NC 28201 &bullet; 704-555-1151
</footer>
```

Images and Attributes

- The `<img>` tag does not have a closing tag
- All of the information is supplied with attributes
  - Key-value pairs: `<html lang="en">`
- Different `<img>` tag attributes
  - Picture to display: `<img src="url">` where url is:
  - In same folder as Web page: `src="cat.jpg"`
  - In a subfolder: `src="images/cat.jpg"`
  - Out on the Web: `src="https://occ-cs.com/steve.png"`
  - Alternative text: `alt="Your instructor, Stephen Gilbert."`
  - Change size with: `height="42" width="64"`

Insert an Image

▸ Let's add a banner image and a photo of the restaurant

  ▸ Note that both pictures are inside the images folder

  ▸ The stylesheet determines how the images are displayed

```
<header>
    <img src="images/logo2.png" alt="Curbside Thai Banner">
</header>
<main>
    <article>
        <h1>About Us</h1>
        <img src="images/photo1.png" alt="Dining in style!">
        <p>
            <strong>Curbside Thai</strong> brings the rich flavor
```

The blockquote Element

▸ Block quotes allow you to group a series of paragraphs and visually set them apart

▸ Open `reviews.html` and insert the reviews section from `pages.txt`
  ▸ Add paragraph tags around each

▸ Then, surround each review with a `<blockquote>` pair of tags

▸ Notice how the stylesheet styles the blockquote elements

```html
<article>
    <h1>Reviews</h1>
    <p>The culinary art of the food truc
    improving over the last 10 years wit
    Antonio's Italian, Organic on the Ro
    Sajja Adulet is to be congratulated
    and authentic Thai food at a reasona
    Curbside Thai provides the finest As
```

```html
<h1>Reviews</h1>
<blockquote>
 <p>The culinary art of the food truck vendo
    improving over the last 10 years with fi
    Antonio's Italian, Organic on the Road,
    Sajja Adulet is to be congratulated for
    and authentic Thai food at a reasonable
    Curbside Thai provides the finest Asian
    the streets of Charlotte.</p>
    <p>Carolina Traveler</p>
</blockquote>

<blockquote>
 <p>Steaming rice with a spicy Asian sauce i
    and we all accept its limitations in exc
    convenience of a quick bite on the run.
    don't have to compromise. Building on hi
```

The Contacts Page and Address Element

- Open `contacts.html`
- Paste text
- Use `<address>` around address
  - End each line with a `<br>`
- Add `<p>` tags to others

```
<article>
    <h1>Contact Us</h1>
    <p>Contact Curbside Thai for your
    out when our mobile truck will ne
    Employment opportunities availabl

    <address>
        Curbside Thai<br>
        Belde Drive<br>
        lotte NC 28201<br>
    ress>

    all: (704) 555-1151</p>
    mail: curbside.thai@example.
    e>

<footer>
```

Contact Us

Contact Curbside Thai for your next event or just to find out when our mobile truck will next be in your area. Employment opportunities available now!

Curbside Thai
411 Belde Drive
Charlotte NC 28201

Call: (704) 555-1151

Email: curbside.thai@example.com

Definition Lists

- List of terms and their descriptions
  - `<dl>`
    ```
    <dt>Term</dt>
    <dd>Description</dd>
    </dl>
    ```
- Open `menu.html` and paste text
  - Wrap all inside a `<dl>` tag
  - Put each menu item in a `<dt>`
  - Each menu description in a `<dd>`
- View the page in your browser

```
<article>
    <h1>Mobile Menu</h
    <dl>
        <dt>Basil Beef Se
        <dd>
            Spicy Angus be
            red cabbage, c
            sesame vinaigr
        </dd>
        <dt>Curbside Rice
        <dd>
            Stir-fried ri
            carrots, garni
            and fried garl
            or shrimp ($9.
        </dd>
        <dt>Garlic Pepper
        <dd>
            Marinated pork
            served with st
            cucumbers, sca
        </dd>
        <dt>Pad Thai</dt>
        <dd>
            Stir-fried ri
            garnished with
            scallions, lim
            add chicken ($
        </dd>
        <dt>Thai Red Curr
        <dd>
```

Mobile Menu

Basil Beef Sesame Salad
Spicy Angus beef and sweet basil on top
and tomatoes; served with sesame vinai

Curbside Rice
Stir-fried rice with onions, red bell pepp
cucumbers, scallions, and fried garlic ($

Garlic Pepper Pork
Marinated pork stir-fried with fresh gar
cabbage, carrot, cucumbers, scallions, a

Pad Thai
Stir-fried rice noodles with bean sprout
scallions, lime, and crushed peanuts ($7

Thai Red Curry
Traditional red curry sauce cooked in co
Thai chili and served on a bed of steam
($9.50)

Curbside Thai • 411 Belde Dr

Ordered and Unordered Lists

Unordered list: bullet points

```
<ul>
    <li>Preprocessor</li>
    <li>Compiler</li>
    <li>Assembler</li>
</ul>
```

Ordered lists: numbered list

```
<ol>
    <li>Create</li>
    <li>Build</li>
    <li>Run</li>
</ol>
```
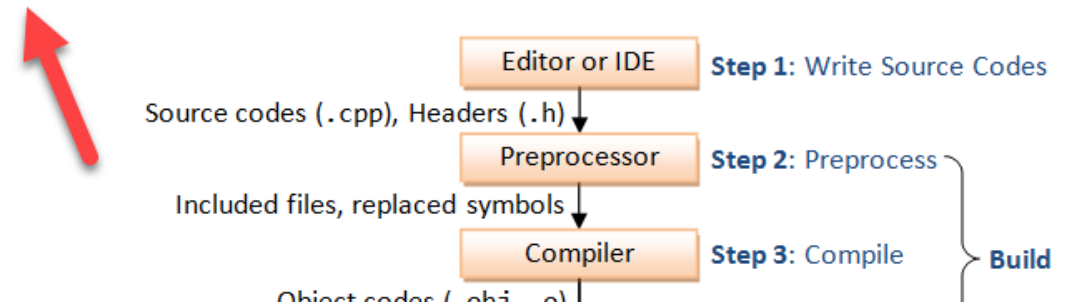
The Build Process

Once you have written your source code, the **build process** turns that source code into an executable program. The build process involves several tools:

- **Preprocessor**—performs text substitution on your source code.
- **Compiler**—generates **assembly code** from the preprocessed source code.
- **Assembler**—converts assembly code into **object** or **native machine code**.
- **Linker**—combines machine-code modules into an **executable** that runs.
- **Make**—provides instructions for **building** each program.
- **Loader**—reads the executable from disk into memory and starts it running.

Program Mechanics

To create your own C++ programs, you follow a **mechanical process**, a well-defined set of steps called the **edit-build-run** cycle. In short you:

1. **Create** your source code using a **text editor**.
2. **Build** your executable using a **compiler toolchain**
3. **Run** your program using your operating system facilities

Source codes (.cpp), Headers (.h)

Editor or IDE — **Step 1**: Write Source Codes

Preprocessor — **Step 2**: Preprocess

Included files, replaced symbols

Compiler — **Step 3**: Compile — **Build**

The <nav> Section and Navigation Lists

- To create a navigation bar, place an unordered list inside a `<nav>` element
  - Should be in a `<header>` section
- Open `about.html` and add this navigation list to the document


```
<header>
    <img src="images/logo2.png" alt="
    <nav>
        <ul>
            <li>About</li>
            <li>Locations</li>
            <li>Reviews</li>
            <li>Catering</li>
            <li>Contact</li>
        </ul>
    </nav>
</header>
<main>
    <article>
        <h1>About Us</h1>
        <img src="images/photo1.png"
        <p>
```

Hyperlinks

- Allows you to "jump" from one document to another
  - `<a href="https://www.google.com">Google</a>`
  - `<a href="about.html">About</a>`
  - Go ahead and add links to each item in the navigation list
- You can use images (instead of text) as links
  - Just wrap the `<img>` tag inside a pair of `<a>` tags
  - `<a href="index.html">`
    `<img src="images/logo2.png" alt="Curbside Thai">`
    `</a>`
  - Go ahead and add this to the About page
  - Finally, replace the header section in the other pages with this one

Learning More

- Online classes at OCC
  - CIS 123 and 129 (Web Design 1 and 2 - HTML and CSS)
  - CIS 171 and 223 (JavaScript 1 and 2)
- Free Code Camp: freecodecamp.org
- W3Schools - OK tutorials: w3schools.com
- Harvard's CS50: cs50.harvard.edu/x
  - Video lectures, IDE, etc. 8 weeks from Scratch through C, Python, SQL
  - Lecture 8 only (Internet, HTML, CSS & JavaScript)
- Harvard's CS50W (Web): cs50.harvard.edu/web/2020/
  - More detailed, requires Python. Take (or skim) CS50 first
  - CS50W Lecture 0 only (HTML & CSS)