Your Own Web Site with HTML, CSS & JavaScript

Part 2 - Cascading Style Sheets

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 2 (today) we'll cover CSS

  ▸ We'll convert an existing Web site by adding CSS

▸ In Part 3 you'll add JavaScript to your Web site

▸ First, though, we'll spend 5-10 minutes to make sure everyone had their Github account, the starter code, and CodeSpace configured

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
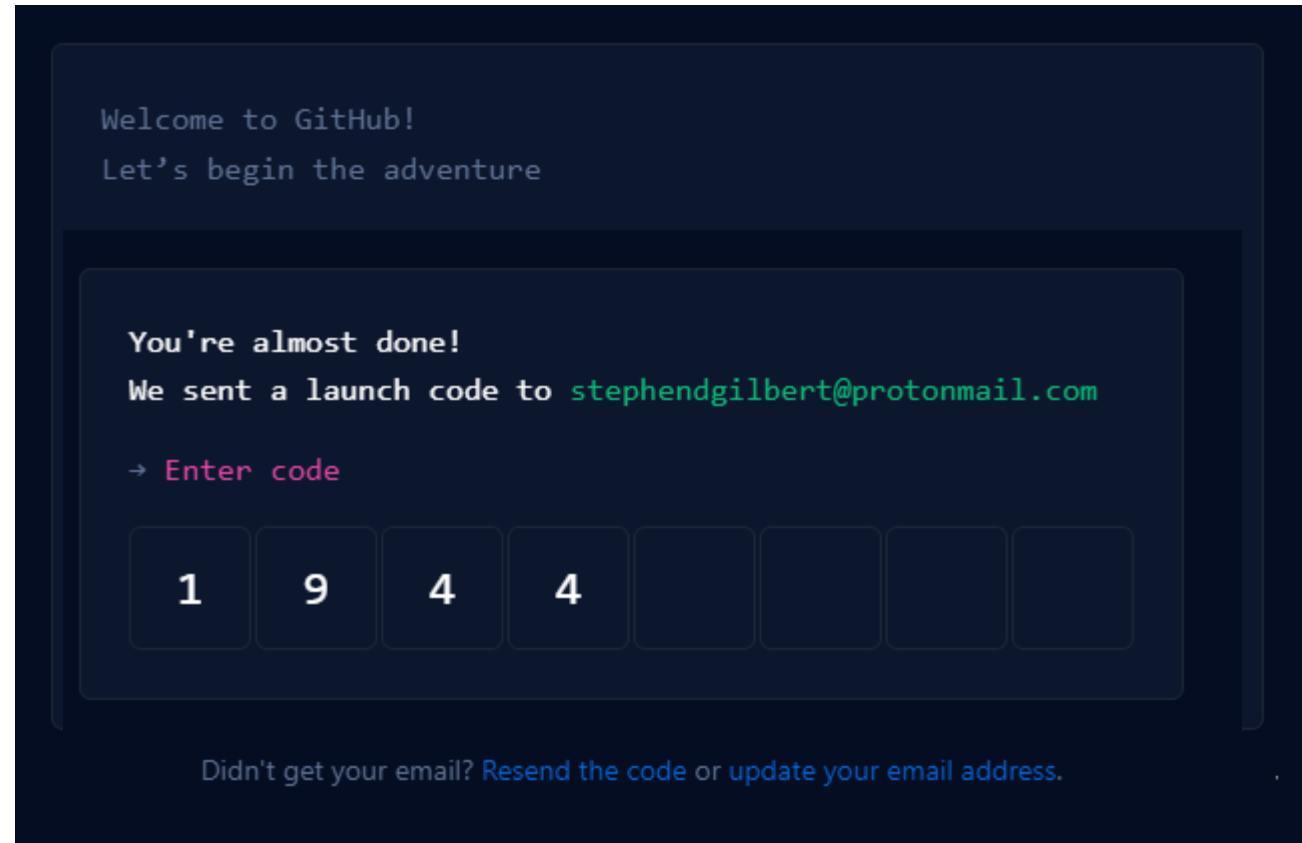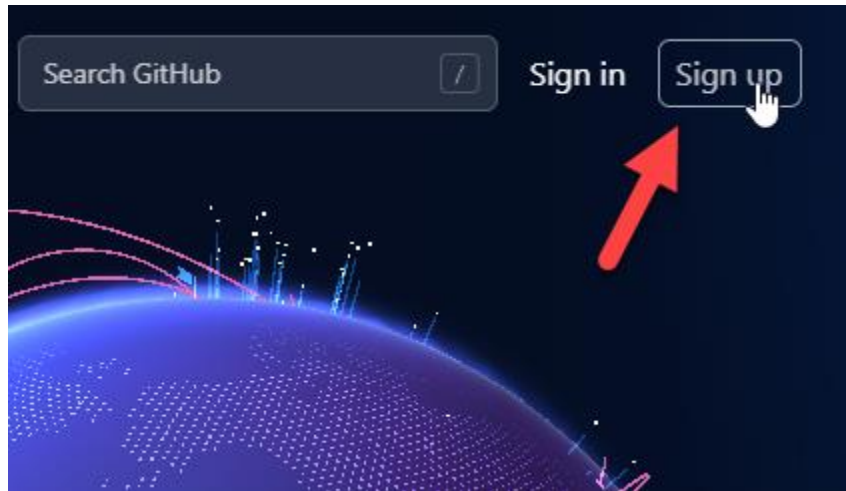
Step 2 - Create Your Starter Repository

▸ The starter code is at: <u>classroom.github.com/a/fBfvck63</u>

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
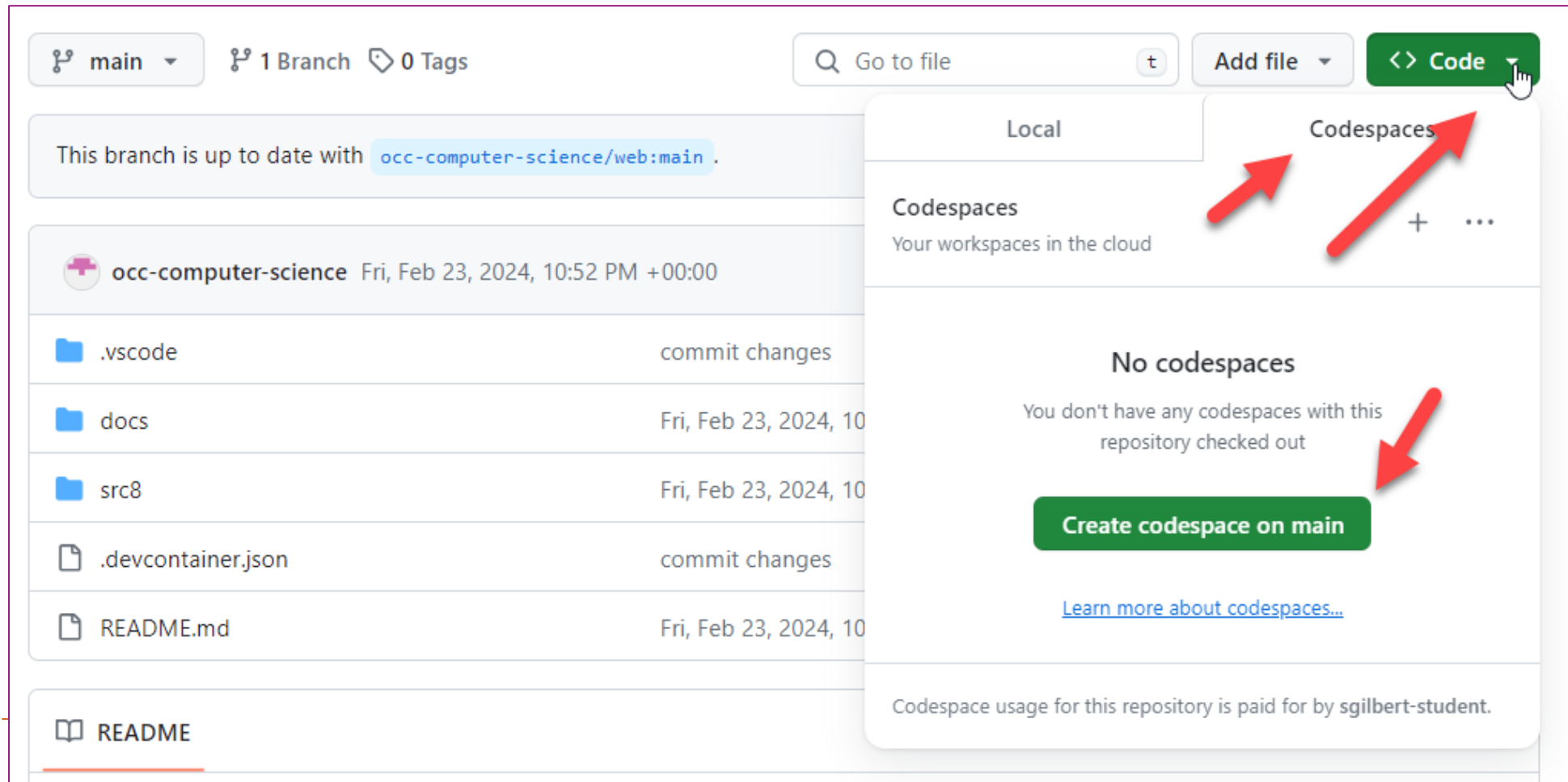
Step 5 - Your VSCode CodeSpace
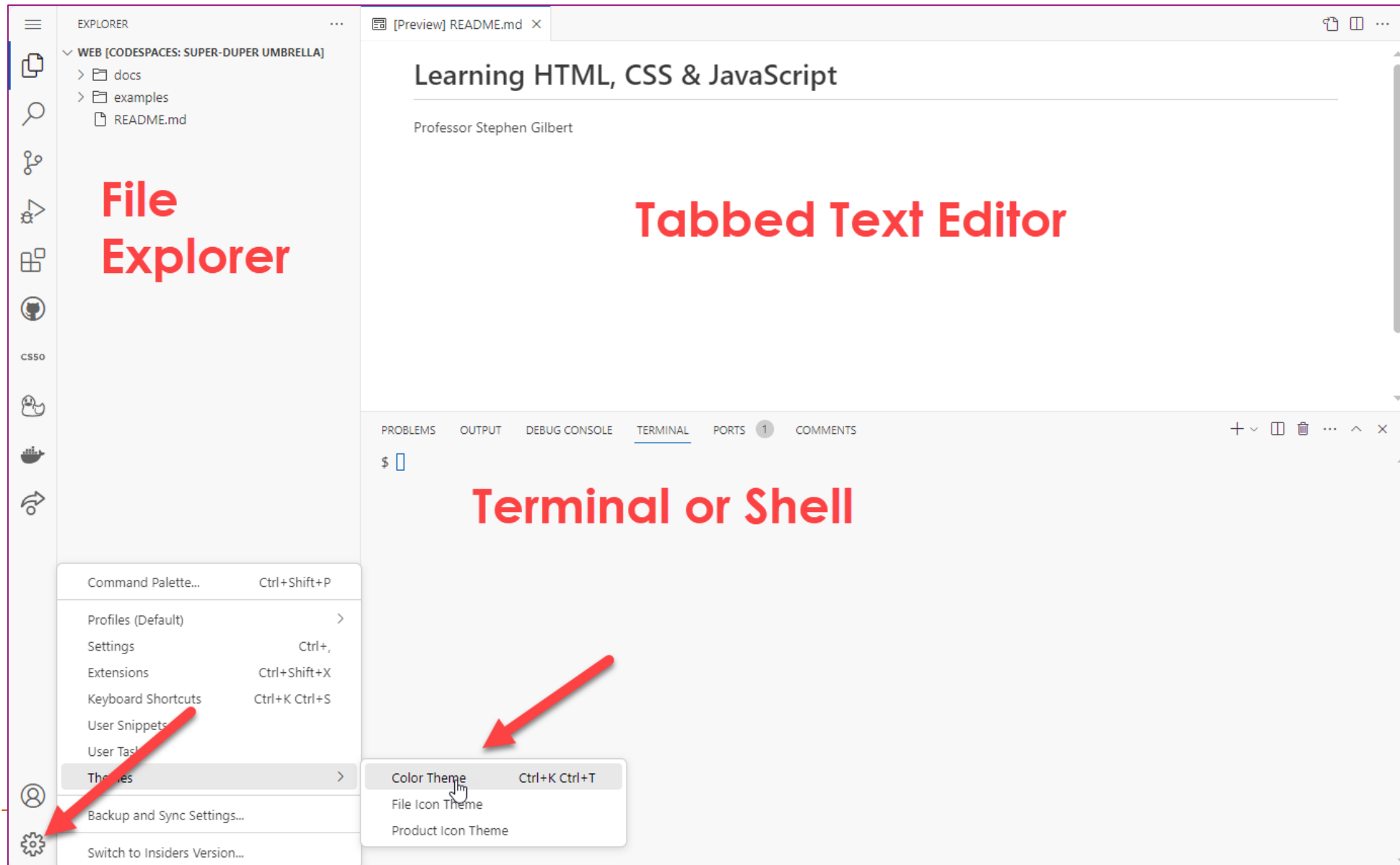
Working with CSS

▸ Our first project adds styling to a set of existing Web pages

  ▸ Right click the css-`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`
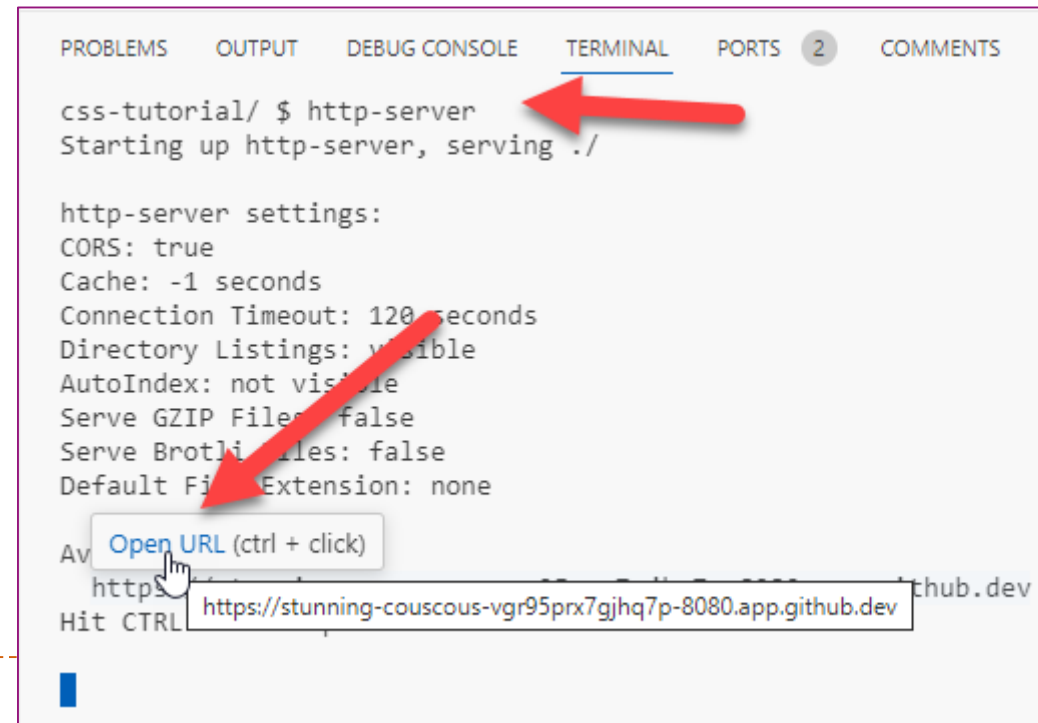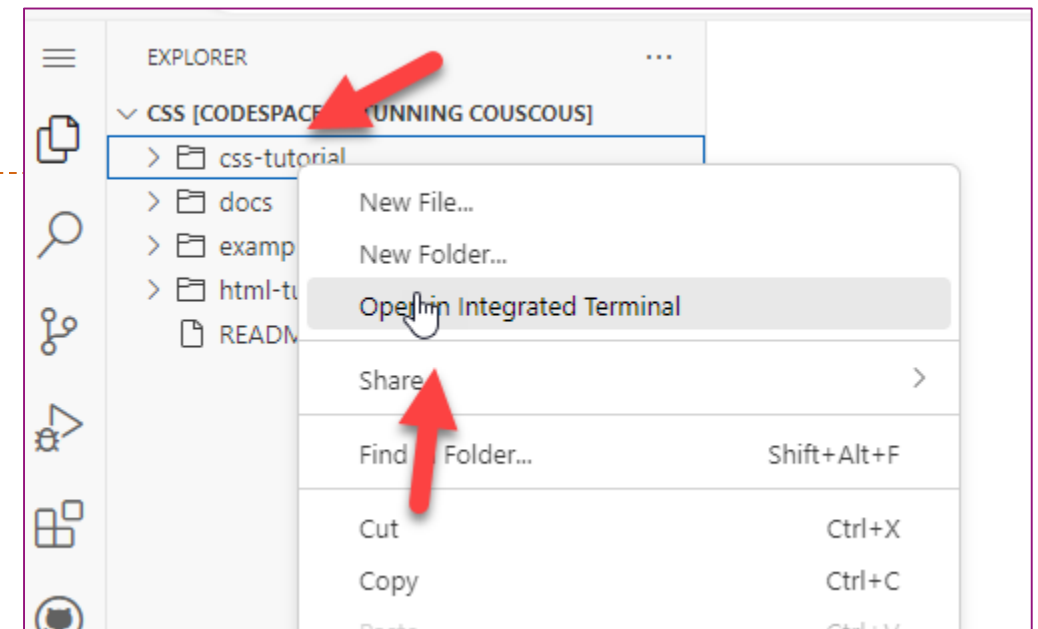
The T&SS Project

▸ The starter site contains 4 pages

  ▸ All of them are un-styled

▸ We will style them with CSS or Cascading Style Sheets

  ▸ Our two style sheets are already linked in the head of each page

  ▸ In `layout.css` we'll add rules to arrange each section on the page

  ▸ In `styles.css` we'll control colors, typography, and other elements.

- triathlon.org
- usatriathlon.org
- Texas Triathlons
- CapTex Triathlon
- Triathlon Calendar
- Triathlete.com
- Trifuel.com

About TSS

Since 2002, **Tri and Succeed Sports** has provided Austin with a first class training center for athletes of all abilities and g

Want to swim? Great! Interested in improving your cycling? Fantastic! Want to tackle a triathlon? We're there for you: be what you start.

Classes

```
 6    <meta name="viewport" content="width=device-width, initial-sca
 7    <title>Tri and Succeed Sports</title>
 8    <link href="styles/layout.css" rel="stylesheet">
 9    <link href="styles/styles.css" rel="stylesheet">
10  </head>
```

What are CSS Rules?

▸ CSS rules have two concepts: selectors and properties

```
h1 {                        // this is a selector
   color: red;              // the color property
   text-align: center;      // the text-align property
}
```

▸ Selectors specify which elements should be styled

▸ You can specify several selectors: `h1, h2 { color: blue; }`

▸ Properties are "key-value" pairs, separated with a colon

▸ The "key" names the property: `color`

▸ The value says what to set it to: `blue`

▸ Must end with a semicolon

Where Do We Put the Rules?

‣ In an external, linked style sheet that usually has the extension `.css`

‣ In an internal style sheet, placed in the head of the document

  ‣ `<link href="styles/styles.css" rel="stylesheet">`
    `<style>`
      `h1 { color: green; text-align: left; }`
    `</style>`

  ‣ The internal style sheet overrides styles in the external style sheet

  ‣ That's what the "cascading" part of CSS is about

‣ As an inline style attached to a particular html tag

  ‣ `<h1 style="color: pink; text-align: right;">Hot!</h1>`

  ‣ This overrides any previously defined style for this `<h1>` tag

Layout and the CSS Box Model

- The properties you'll use to control the layout of your page are called the CSS box model

- For the layout, we're concerned with `margin` and `width`

- Go ahead and open `layout.css` and add rules for the `body`, `header`, `nav`, `aside`, `footer`, `#about` and `.syllabus`

  - `#about` is called an id selector

  - `.syllabus` is called a class selector
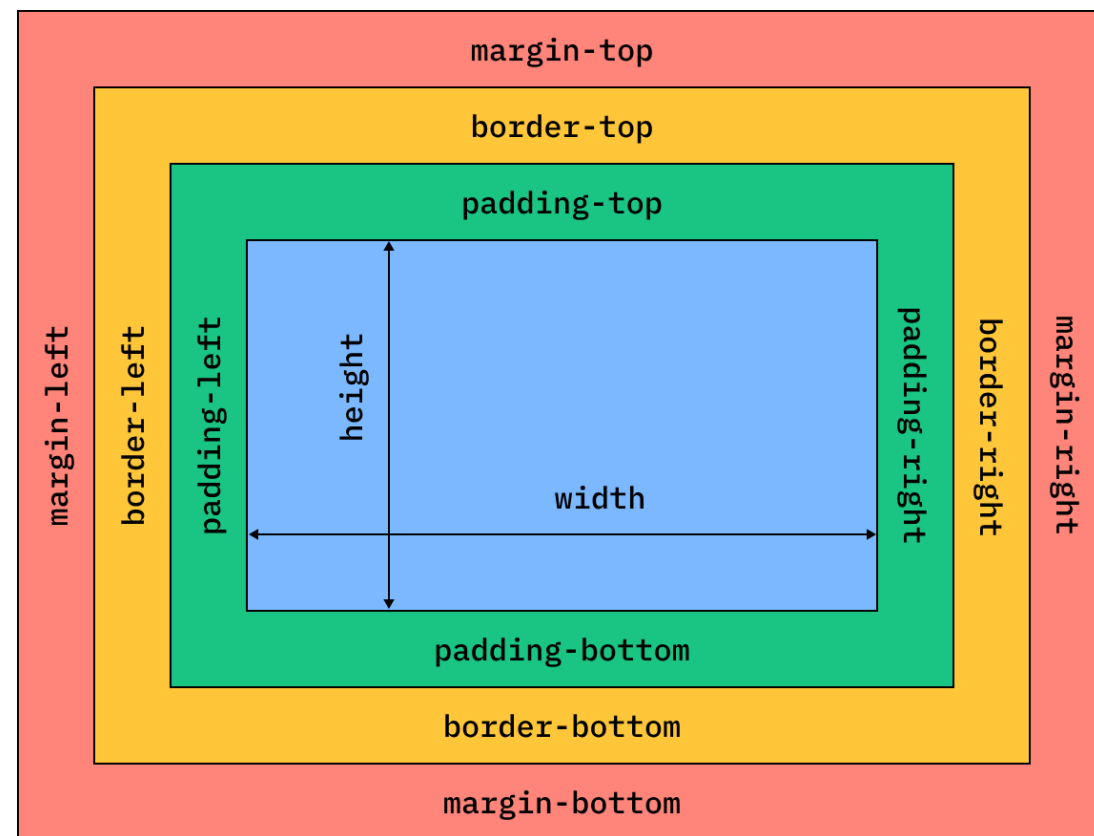
The width Property

▸ The `width` of an element can be specified
in percentages, pixels, or other units, like ems
  ▸ `body {`
    ```
    width: 100%;
    max-width: 1020px;
    min-width: 640px;
    }
    ```
  ▸ May also specify `max-width` and `min-width`
  ▸ Useful for responsive design
▸ Go ahead and add width's to your selectors

Introdu

```css
 4  body {
 5      width: 100%;
 6      max-width: 1020px;
 7      min-width: 640px;
 8  }
 9
10  header {
11      width: 100%
12  }
13
14  nav {
15      width: 20%
16  }
17
18  aside {
19      width: 20%
20  }
21
22  footer {
23      width: 100%;
24  }
25
26  #about {
27      width: 51%
28  }
29
30  .syllabus {
31      width: 74.5%
32  }
```

The margin Property

▸ You can write the margin property in several different ways

  ▸ `margin: 0;`   All 4 margins are 0

  ▸ `margin: 0 auto;`   Top-bottom 0, rest auto

  ▸ `margin: 0 3% 0 3%;`   Top-bottom 0, left-right 3%

▸ Go ahead and add margins

Introduction to

```
 4  body {
 5      margin: 0 auto;
 6      width: 100%;
 7      max-width: 1020px;
 8      min-width: 640px;
 9  }
10
11  header {
12      margin: 0;
13      width: 100%
14  }
15
16  nav {
17      width: 20%;
18      margin: 0 0 0 1.5%;
19  }
20
21  aside {
22      width: 20%;
23      margin: 0 1.5% 0 0;
24  }
25
26  footer {
27      width: 100%;
28  }
29
30  #about {
31      width: 51%;
32      margin: 0 3% 0 3%;
33  }
34
35  .syllabus {
36      width: 74.5%;
37  }
38
```

The float Property

- Now that we've provided width's for each element, that element is in a "box"
  - Unless we do something, those boxes are stacked on top of each other like building blocks
- The `float` property allows the browser to arrange the boxes next to each other
  - You can `float` an item to the `left` or `right`
  - Add `float: left;` to the `nav`, `aside`, `#about` and `.syllabus` rules
- Now, only the picture looks out of place

Introducti

```css
11  header {
12      margin: 0;
13      width: 100%
14  }
15
16  nav {
17      width: 20%;
18      margin: 0 0 0 1.5%;
19      float: left;
20  }
21
22  aside {
23      width: 20%;
24      margin: 0 1.5% 0 0;
25      float: left;
26  }
27
28  #about {
29      width: 51%;
30      margin: 0 3% 0 3%;
31      float: left;
32  }
33
34  .syllabus {
35      width: 74.5%;
36      float: left;
37  }
38
```

Floating the Image

- We want the image to float to the right, but not all images
- Specify exactly which element is selected with these patterns
  - `header h1 { ... }` any `h1` inside a `header` section (descendant)
  - `article > p { ... }` any `p` immediately inside an `article` section
  - `h1 + h2 { ... }` any `h2` immediately preceded by an `h1`
- Our image is inside the `article`, which has `id="about"`
  - ```
    #about img {
        float: right;
        width: 40%;
        margin: 0 4% 0 0%;
    }
    ```

Styling the Content

▸ You're going to add styles to `styles/styles.css`

▸ We'll start by specifying colors in the page: here are different ways

    ▸ Named colors: 147 predefined solid colors ([Wikipedia link](#))

    ▸ RGB colors: `rgb(255, 20, 147)`, `rgb(100%, 10%, 55%)`, `#FF1499`

    ▸ HSL (hue, saturation, luminance) colors ([Color Picker](#))

        ▸ `hsl(320, 100%, 50%)` - hue: 320 degrees, 100% saturation, 50% luminance

    ▸ Both RGB and HSL have an A variant to specify transparency

Properties: color & background-color

- Let's start by specifying colors
  - You can specify both foreground and background for any element
  - For `html`, use a color of `hsl(27, 72%, 72%)` or pick another at: hslpicker.com
  - For `body`, use a color of `rgb(91, 91, 91)` and a background of `ivory`
- Use white and two different shades of orange for the `<h1>` and `<h2>` headings

```
 4    /* HTML and Body Styles */
 5
 6    html {
 7        background-color: hsl(27, 72%, 72%);
 8    }
 9
10    body {
11        color: rgb(91, 91, 91);
12        background-color: ivory;
13    }

16
17    /* Heading Styles */
18    h1 {
19        color: white;
20        background-color: rgb(222, 128, 60);
21    }
22
23    h2 {
24        color: white;
25        background-color: rgb(235, 177, 131);
26    }
27
28
```

ID Selectors

- In your HTML page, you can add an `id` attribute to any element
  - Each `id` needs to be unique on a particular page
- In CSS, you use `#id-name` as the selector for the rule
- Let's modify our `h1` and `h2` rules so they only affect those in the "about" section
  - Refresh to see the result

```
44      <article id="about">
45          <h1>About TSS </h1>
46          <img src="images/photo1.png" alt="" />
47          <p>Since 2002, <strong>Tri and Succeed Sp
48      with a first class training center for at
49      goals. We specialize in helping you reach
50      us what you want to do; we work to fulfil
51          <p>Want to swim? Great! Interested in imp
52      Want to tackle a triathlon? We're there f
53      race. Or do you just want to get more fit
54      instruction to match your goals. And you
16
17  /* Heading Styles */
18  #about h1 {
19      color: white;
20      background-color: rgb(222, 128, 60);
21  }
22
23  #about h2 {
24      color: white;
25      background-color: rgb(235, 177, 131);
26  }
27
```

Class Selectors

- Like `id`, `class` attributes can be added to any HTML element
  - `class` attributes do not need to be unique, however, like `ids` do
  - The `swim`, `run` and `bike` pages all use the `syllabus` class

- In CSS, a `class` selector starts with a dot: `.syllabus {...`

- Add this rule for the `syllabus` headings

```
25
26      <article class="syllabus">
27          <h1>Swim for Fitness</h1>
28          <p>TSS coaches take a scientific app
29          providing knowledge, motivation, con
30          athlete who want to improve his or
31          of swimming lies in proper technique
32          the water with speed and efficiency
33          our coaches can maximize the time sp
34          <p>We offer three instruction times
35
```

```
27
28  .syllabus h1, .syllabus h2 {
29      color: rgb(101, 101, 101);
30      background-color: rgb(255, 185, 255);
31  }
```

The font-family Property

- To specify the font to use for an element, use `font-family`
  - Follow with a comma-separated list of fonts to use (font-stack)
  - End with a generic font name
  - `serif`, `sans-serif`, `monospace`, `cursive`, `fantasy`
- "Web-safe" fonts are those that are available on most computers
  - This is the list from Google
  - You computer probably has more

```
10  body {
11      color: rgb(91, 91, 91);
12      background-color: ivory;
13      font-family: Geneva, Verdana, sans-serif;
14  }
15
16
```

Arial
Verdana
Tahoma
Trebuchet MS
Times New Roman

Georgia
Courier New
Brush Script MT
Impact

Google's Web Fonts

- If you want more fonts, then you can download fonts and place them in your site
  - Or, you can link directly to the font
- Visit **fonts.google.com**
  - Type **Quicksand** and click sample that is displayed
  - Click the Get Font button
  - Click Get Embed Code
  - Click **@import** button and copy code into your CSS
- Now add to **h1** and **h2**

The font-size Property

▸ You can specify the size of a font using different measures

  ▸ `font-size: 12pt;` (points)

  ▸ `font-size: 14px;` (pixels)

  ▸ `font-size: 200%;` (percentage)

  ▸ `font-size: 1.2em;` (em units)

▸ Generally you want to use *em* units to specify font sizes

  ▸ Makes it relative to the size of the containing element

▸ Add sizes to the fonts

```
17  /* Heading Styles */
18
19  h1 {
20      font-size: 2.2 em;
21  }
22
23  h2 {
24      font-size: 1.1 em;
25  }
26
```

```
46  /* Aside and Blockquote Styles */
47
48  aside {
49      font-size: 0.8em;
50  }
51
```

```
58  /* Navigation Styles */
59  nav {
60      font-size: 0.8em;
61  }
62
```

Introduction to CSS   15-Mar-24

Kerning, Tracking and Leading

- Controls space between letters, words, and lines of text
    - `letter-spacing: value;`
    - `word-spacing: value;`
    - `line-height: value;`
- Set kerning to `0.1em` for `h1` and `h2` headings
- Set leading to `2em` (double-spaced) for `nav > ul` elements
    - Refresh and see changes

```
27  h1, h2 {
28      font-family: "Quicksand", Geneva, V
29      letter-spacing: 0.1em;
30  }
31
```

```
59  /* Navigation Styles */
60  nav {
61      font-size: 0.8em;
62  }
63
64  nav > ul {
65      line-height: 2em;   /* double spaced */
66  }
67
68
```

Other Font Properties

▸ Some other font/text properties

  ▸ `font-style`: normal or italic

  ▸ `font-weight`: normal or bold

  ▸ `font-variant`: normal or small-caps

  ▸ `text-decoration`: none, underline, overline or strikethrough

  ▸ `text-transform`: capitalize, uppercase or lowercase

▸ Can combine these in one `font` property

  ▸ `font`: *style variant weight size/line-height family;*

  ▸ Only `size` and `family` are required

```
67
68  /* Footer Styles */
69  footer address {
70      background-color: rgb(222, 128, 60);
71
72      /* Text is 75% transparent */
73      color: rgba(255, 255, 255, 0.7);
74
75      font: normal                    /* style      */
76            small-caps                /* variant    */
77            bold                      /* weight     */
78            0.9em/3em                 /* size/height */
79            Quicksand, Geneva, Verdana, sans-serif;
80
81      text-align: center;
82  }
83
84
```

Formatting Lists

▸ Use `list-style-type` to specify

   ▸ Unordered: `none`, `disc`, `circle`, `square`

   ▸ Ordered: `none`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `lower-greek`, `upper-greek`

▸ Use descendant selectors to create an outline style

```css
56 nav > ul {
57     line-height: 2em;
58     list-style-type: none;
59 }
60
61
62 /* List Styles */
63 .syllabus ol {
64     list-style-type: upper-roman;
65 }
66
67 .syllabus ol ol {
68     list-style-type: upper-alpha;
69 }
70
71 .syllabus ol ol ol {
72     list-style-type: decimal;
73 }
74
```

Using Images as List Markers

▸ For unordered lists, you can use your own image for the "bullet"

  ▸ `list-style-image`: `url(../images/runicon.png)`

▸ Use pseudo-classes to give each bullet a different image

  ▸ `li:first-of-type`

  ▸ `li:nth-of-type(2)`

  ▸ `li:last-of-type`

```css
63  /* List Styles */
64
65  #about ul li:first-of-type {
66      list-style-image: url(../images/runicon.png);
67  }
68
69  #about ul li:nth-of-type(2) {
70      list-style-image: url(../images/bikeicon.png);
71  }
72
73  #about ul li:last-of-type {
74      list-style-image: url(../images/swimicon.png);
75  }
76
```

Padding and Margins Again

▸ Padding is the area around the content
  ▸ `padding: 0 1em;`
  ▸ `padding: 0 5px 0 10px;`
  ▸ `padding-left: 5px;`
    ▸ Set the padding for the `nav` list
▸ Margins are area around whole
  ▸ Set margins for `li.newgroup`
▸ Set color and margins for the `aside blockquote` elements

```
56  nav > ul {
57      line-height: 2em;
58      list-style-type: none;
59      padding-left: 5px;
60  }
61
62  nav > ul > li.newgroup {
63      margin-top: 20px;
64  }

48  aside blockquote {
49      color: rgb(232, 165, 116);
50      margin: 20px 3px;
51  }
52
```

Hyperlink Pseudo-classes

- You can style a hyperlink based on:
    - `a:link`   An unvisited link
    - `a:visited`   Previously visited
    - `a:hover`   Mouse hovering
    - `a:active`   Being clicked
- May change any elements, such as color, text-decoration, font style, etc.
    - Change color and text decoration of navigation links as shown here

```
70  nav > ul > li > a:link,
71  nav > ul > li > a:visited {
72      color: rgb(151, 151, 151);
73      text-decoration: none;
74  }
75
76  nav > ul > li > a:hover,
77  nav > ul > li > a:active {
78      color: rgb(255, 64, 255);
79      text-decoration: underline;
80  }
81
82
```

Links

Home

Running

Cycling

Swimming

About TSS

Since 2002, **Tri and Succeed Sports** has provided Austin with first class training center for athletes of all abilities and goals. We specialize in helping you rea

Some Decorative Quotes

- CSS can generate information
  - `element::before` { content: value; }
    `element::after` { content: value;}
  - The value can include counters, text, opening-and-closing quotes, and attributes, such as `href`
- Let's add quotes to the aside
  - `quotes: "\201C" "\201D";`
  - `aside blockquote::before`
  - `aside blockquote::after`

```
48  aside blockquote {
49      color: rgb(232, 165, 116);
50      margin: 20px 3px;
51      quotes: "\201C" "\201D";
52  }
53
54  aside blockquote::before {
55      content: open-quote;
56      font-family: serif;
57      font-size: 1.6em;
58      font-weight: bold;
59  }
60
61  aside blockquote::after {
62      content: close-quote;
63      font-family: serif;
64      font-size: 1.6em;
65      font-weight: bold;
66  }
67
```

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)