

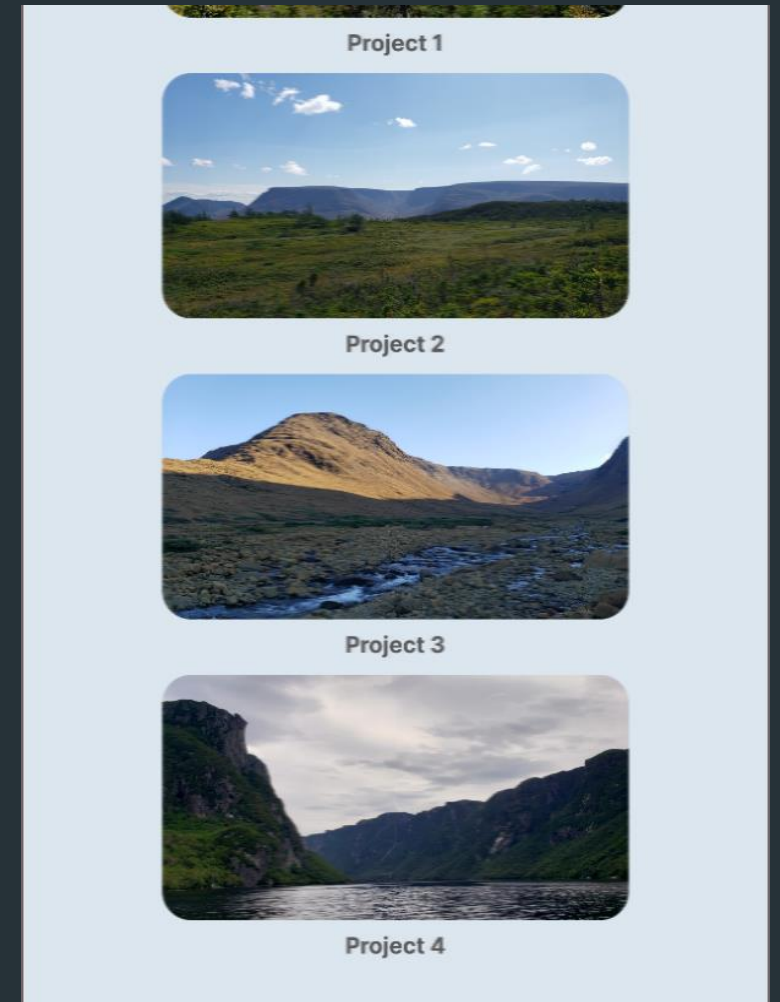
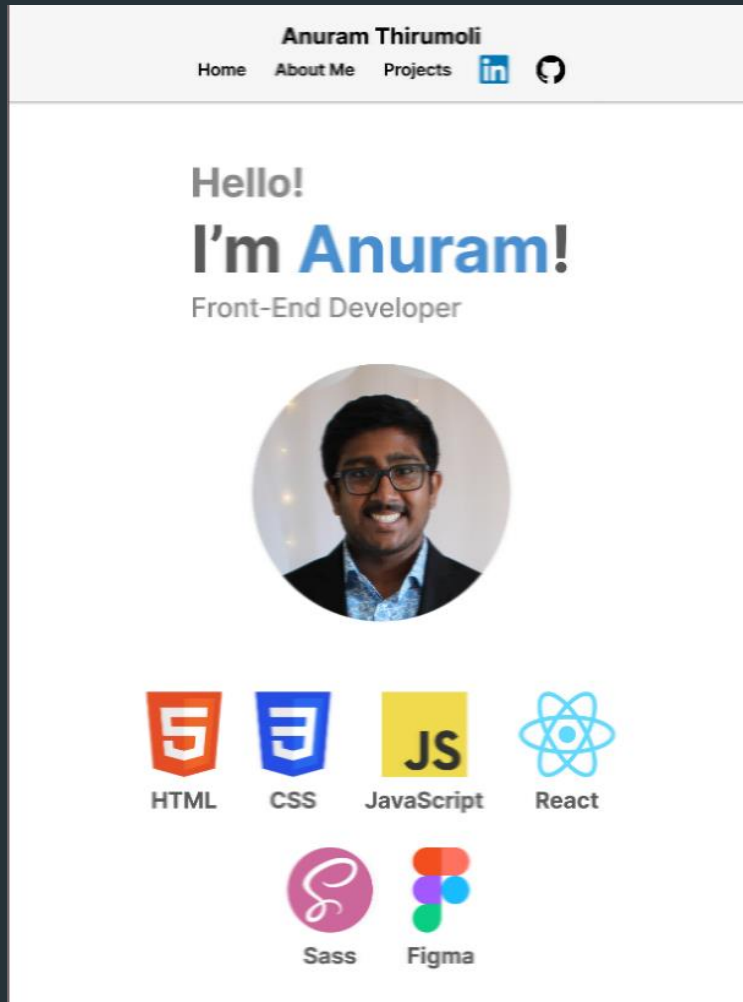
HTML & CSS

Portfolio Website Workshop

- Website Design
- HTML
 - Making Content?
 - HTML Elements
 - HTML Skeleton Code
 - Self-Closing vs. Closing Elements
 - Header, Body, and Main
 - Semantic Elements
 - Attributes
- File Structure, Relative Linking
- CSS
 - Responsive Design
 - Typical Screen Sizes
 - CSS Syntax
 - Cascading in CSS
 - CSS Box Model
 - Styling Links
 - CSS Flexbox
 - CSS Grid
 - Media Queries
- Future Learning



Website Design



HTML

Hyper Text Markup Language



use links to get from one page to another

adds meaning to text or other content
(marking content)

Marking Content?

What We Want

Introduction

In this paragraph we are introducing how HTML, CSS, and JavaScript works. We will be creating a Portfolio Website!

- HTML
- CSS
- JavaScript

Markup

```
<h1>Introduction</h1>
```

```
</p>In this paragraph we are introducing  
how HTML, CSS, and JavaScript works.  
We will be creating a Portfolio  
Website!</p>
```

```
<ul>
```

```
    <li>HTML</li>
```

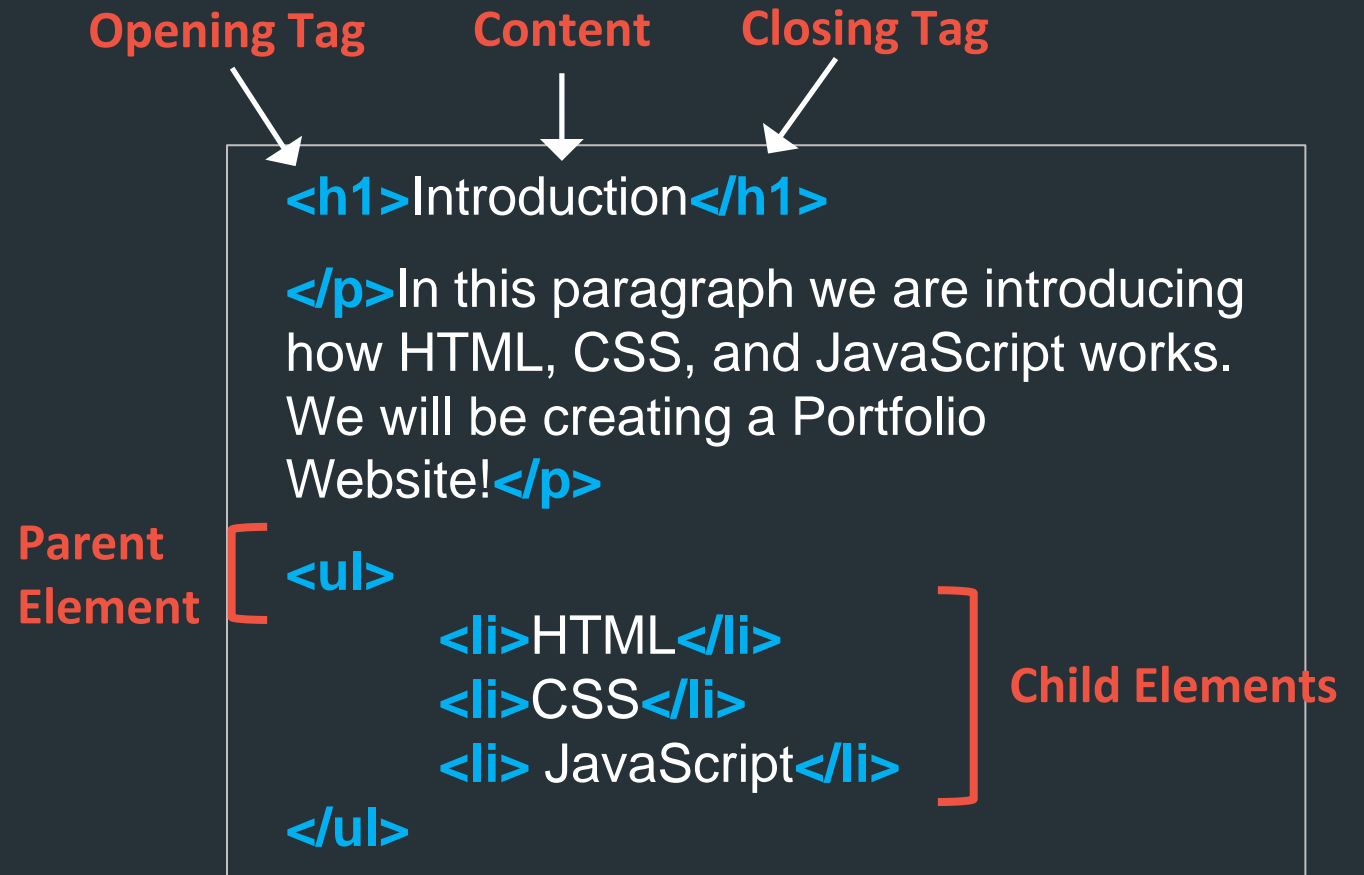
```
    <li>CSS</li>
```

```
    <li> JavaScript</li>
```

```
</ul>
```

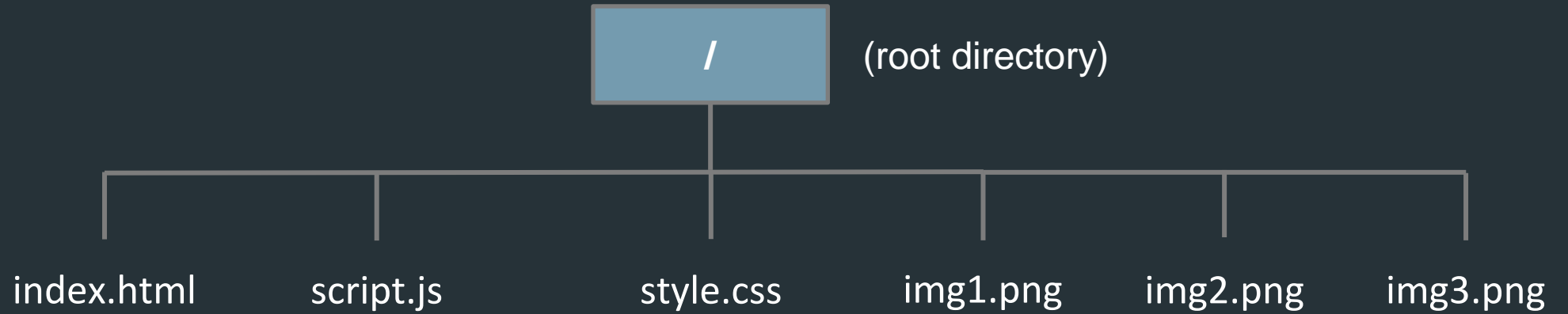
HTML Elements

- HTML elements allow us to give meaning to content.
- Elements are written using tags.
- Elements that enclose content need opening and closing tags.
- You can have elements nested inside other elements.



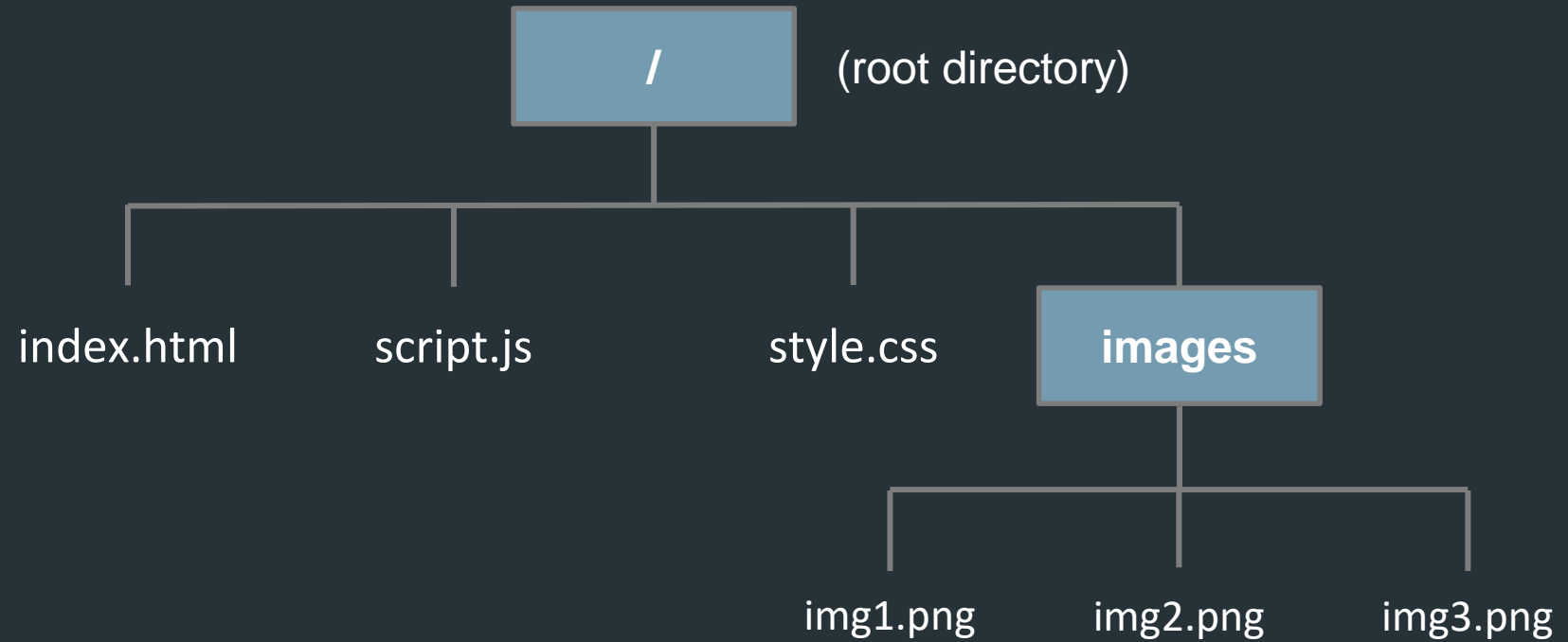
File Structure

File structure is important!

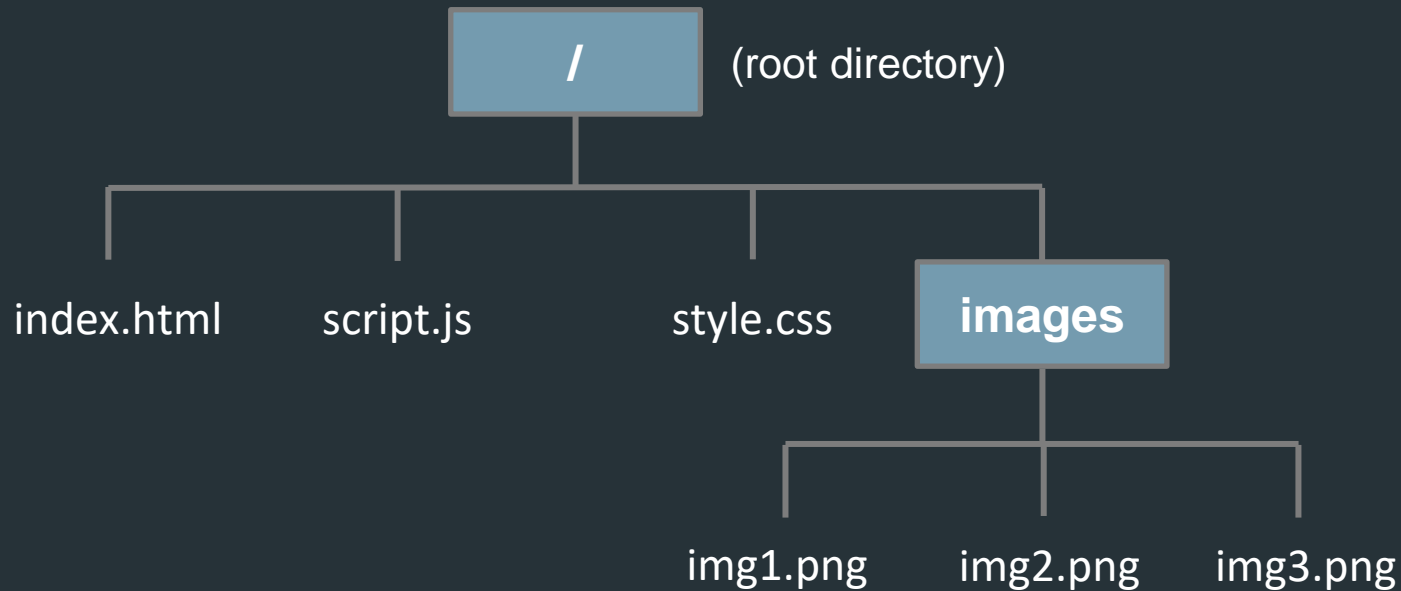


File Structure

File structure is important!



Relative Linking



To access subfolders:

`subfolder/file.html`

`subfolder/subfolder/image.png`

`images/img1.png` (from root folder)

To access parent folders:

`../`

`../../`

`../script.js` (from images folder)

HTML Skeleton Code

```
<!DOCTYPE html>
<html lang="en-ca">

  <head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Page Title</title>

    <!--Additional elements for browsers and robots go here goes here-->

  </head>

  <body>
    <!--Elements visible to users go here-->
  </body>

</html>
```

Self-Closing vs. Closing Elements

Self Closing

`<meta charset="UTF-8">`

Opening Tag

Content

Closing Tag

`<h1>Introduction</h1>`

`</p>`In this paragraph we are introducing how HTML, CSS, and JavaScript works. We will be creating a Portfolio Website!`</p>`

Opening Element

``

`HTML`

`CSS`

` JavaScript`

Content

Opening Element

``

Header, Body, and Main

`<header></header>`

Introductory Content
(navbar)

`<main></main>`

Primary Content

`<footer></footer>`

Content About Webpage
(year of publication, author info,
sitemap)

Semantic Elements

Elements whose names convey the meaning or type of content in the element.

Always prioritize
using semantic
elements whenever
possible!

Why?

1. Helps developers understand your HTML code faster.
2. Allows users to use accessibility features such as screen readers to understand what is on your website.
3. Allows browsers to read your content, therefore improving the SEO (Search Engine Optimization) of your website and bringing it to the top of search results.

Semantic Elements – Examples

Elements whose names convey the meaning or type of content in the element.

Element	Description
<code><head></code>	Contains metadata (machine-readable information) about the document.
<code><meta></code>	Represents metadata that cannot be represented by other HTML metadata elements.
<code><title></code>	Defines the title of the HTML document, which is shown in a browser tab.
<code><body></code>	Represents the content of an HTML document.
<code><link></code>	Specifies that there is a relationship between the HTML document and an external resource, commonly used for CSS files.
<code><header></code>	Section representing introductory content.
<code><main></code>	Section representing the main content of the document.
<code><footer></code>	Section representing informational content about the webpage.

List of HTML Elements:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

Attributes

Changes the features of HTML elements.

One Attribute

```
[<a href="index.html">Anuram Thirumoli</a>
```

No Attributes

```
[<nav>  
  <ul>
```

Multiple Attributes

```
[    <li><a href="#home" class="nav-links">Home</a></li>  
    <li><a href="#about-me" class="nav-links">About Me</a></li>  
    <li><a href="#projects" class="nav-links">Projects</a></li>  
  </ul>  
</nav>
```

Semantic Elements – More Examples

Element	Description
<code><p></code>	Represents a paragraph.
<code><a></code>	Creates hyperlinks to other webpages, files, email addresses, content within the page, or any URL.
<code><nav></code>	Section representing navigational links.
<code></code>	Represents an unordered list of items.
<code></code>	Represents an item in a list, used within a parent element such as <code></code> (ordered list), <code></code> (unordered list) or <code><menu></code> .

```
<a href="index.html">Anuram Thirumoli</a>
```

```
<nav>
  <ul>
    <li><a href="#home" class="nav-links">Home</a></li>
    <li><a href="#about-me" class="nav-links">About Me</a></li>
    <li><a href="#projects" class="nav-links">Projects</a></li>
  </ul>
</nav>
```

CSS

Cascading Style Sheets



```
graph TD; CSS[Cascading Style Sheets] --> A[defined styles cascade, higher precedence with overwrite rules of a lower precedence]; CSS --> B[sets the style or appearance of HTML content (visual appearance)]
```

The diagram illustrates the concept of Cascading Style Sheets (CSS). At the top, the word 'CSS' is written in large red letters. Below it, the full name 'Cascading Style Sheets' is written in white. Two white arrows point downwards from the underlined 'Cascading' and 'Style' parts of the title to two separate text blocks. The left block explains that defined styles cascade, with higher precedence rules overwriting lower precedence ones. The right block states that CSS sets the style or appearance of HTML content, which is its visual appearance.

defined styles cascade, higher precedence with overwrite rules of a lower precedence

sets the style or appearance of HTML content (visual appearance)

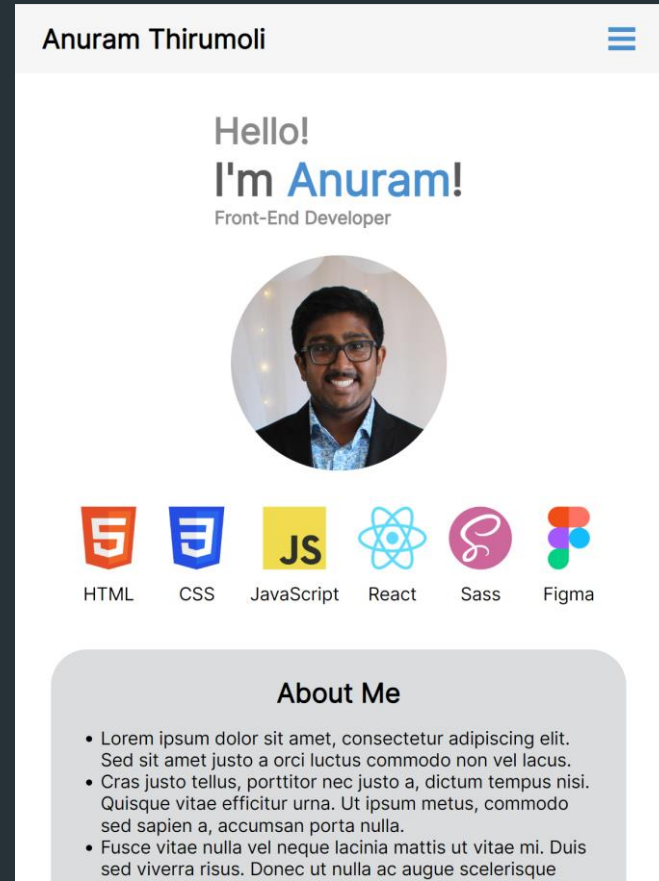
Why CSS? To have a standard stylesheet that applies to our website across all its pages.

Responsive Design

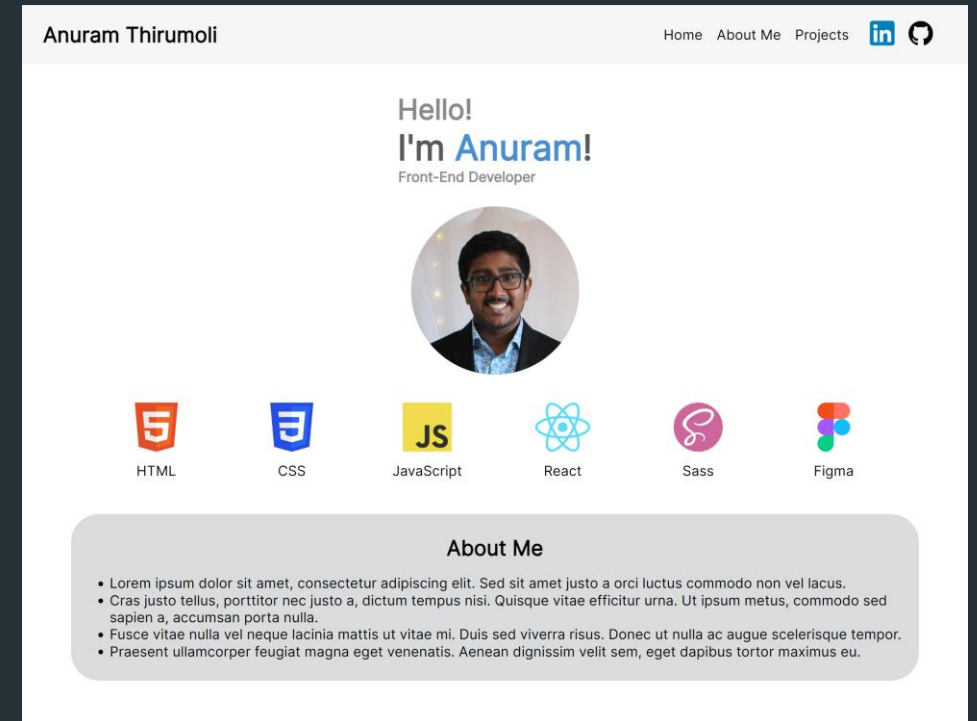
The approach to designing websites that allow web pages to look good on all or most display sizes.



iPhone SE



iPad Pro



Desktop

Typical Screen Sizes

Different ranges to develop layouts or change content size for.

Mobile: 320px to 480px

Tablets: >480px to 768px

Notebooks: >768px to 1024px

Large Screens: >1024px to 1200px

Extra Large Screens: >1200px

CSS Syntax

Selector: what html element should be styled

```
[ html {  
    font-size: 16px;  
}]
```

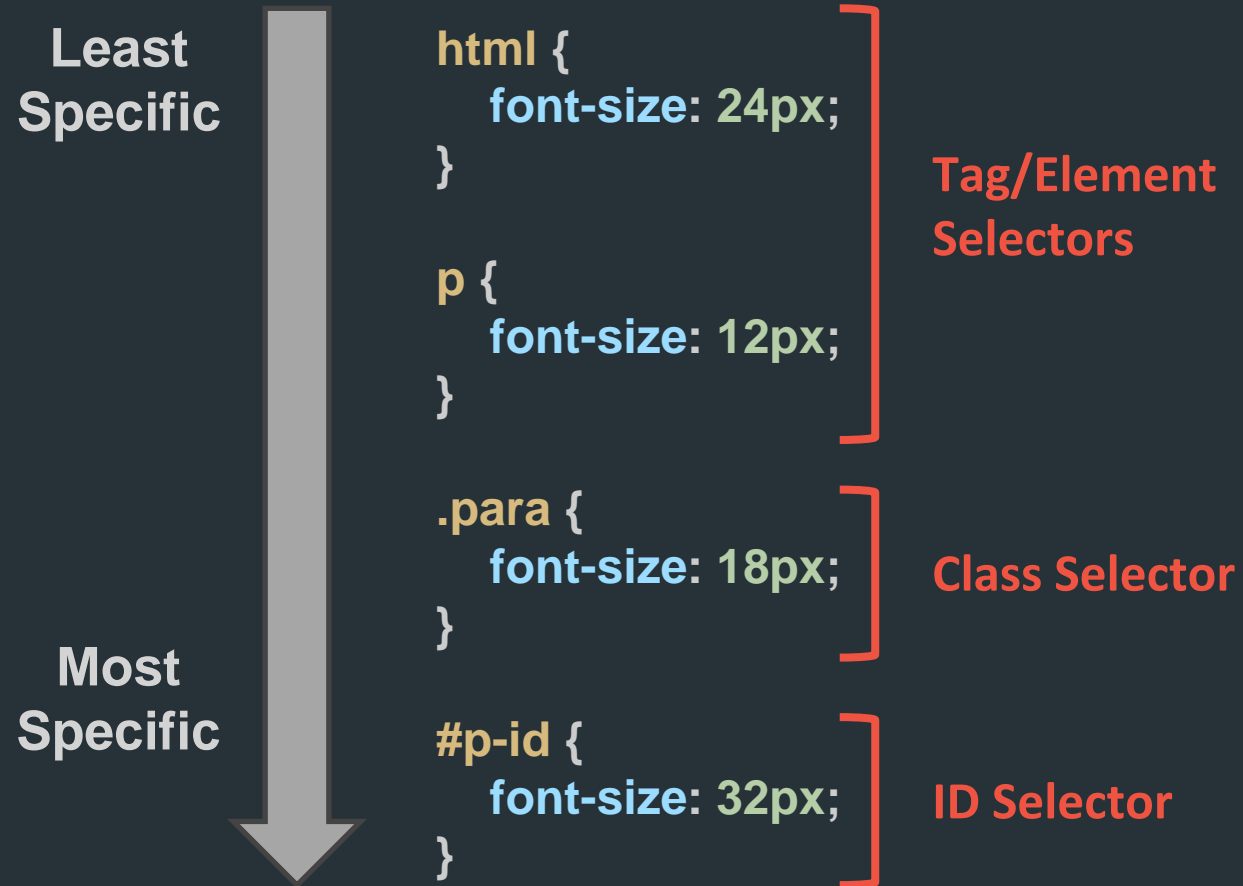
Declaration: how to style the element

Property: What property of the element to adjust

Value: What to change the property to

Cascading in CSS

Specificity



What Will the Font Size of Each Element Be?

`Hello!` **24px**

`<p>Hello!</p>` **12px**

`<p class="para">Hello!</p>` **18px**

`<p id="p-id" class="para">Hello!</p>` **32px**

Cascading in CSS

Order of Declaration

```
html {  
  font-size: 16px;  
  color: #FF0000;  
}
```

```
.para {  
  font-size: 32px;  
  color: #00FF00;  
}
```

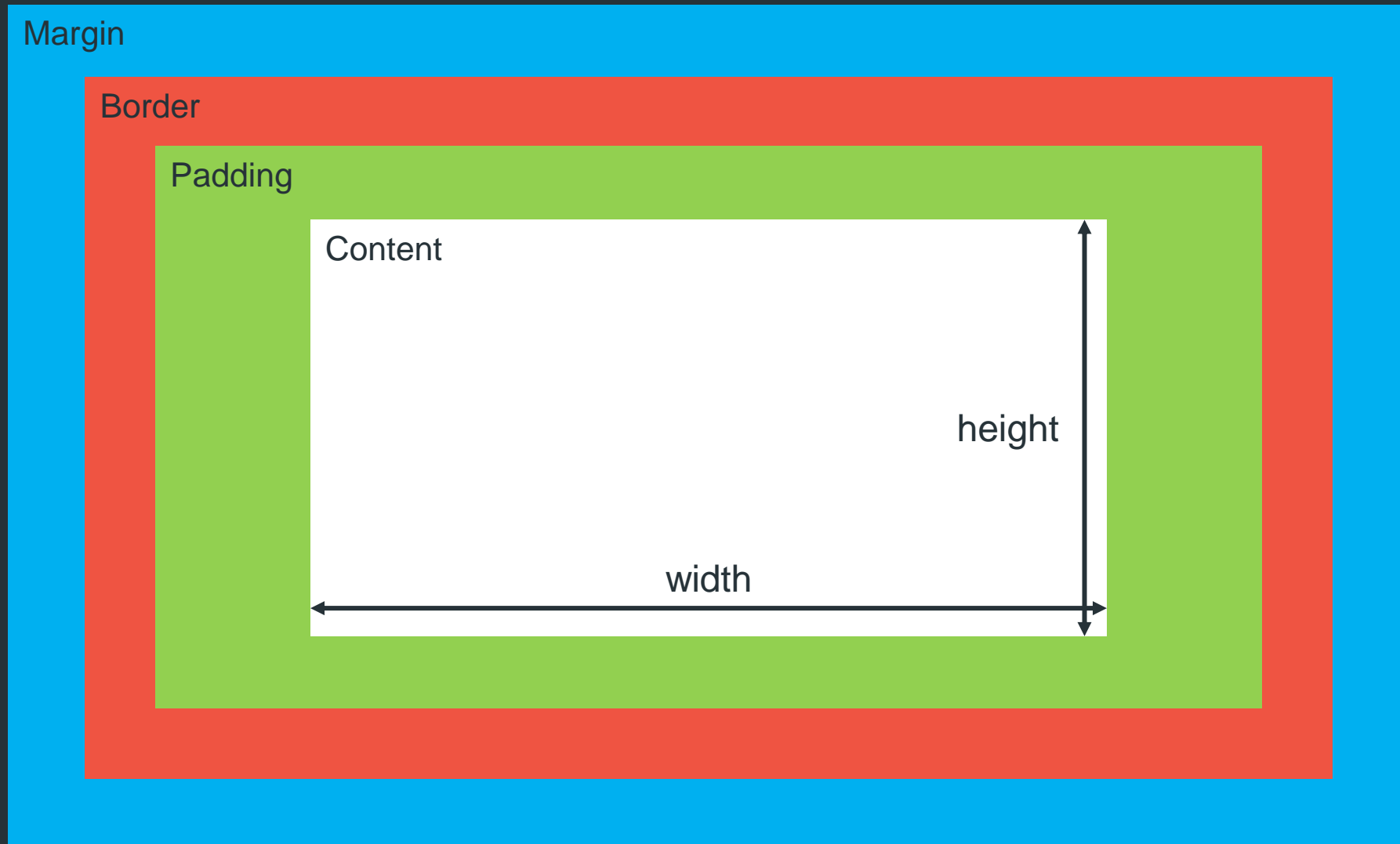
```
.para1 {  
  color: #0000FF;  
}
```

```
.para2 {  
  font-size: 24px;  
}
```

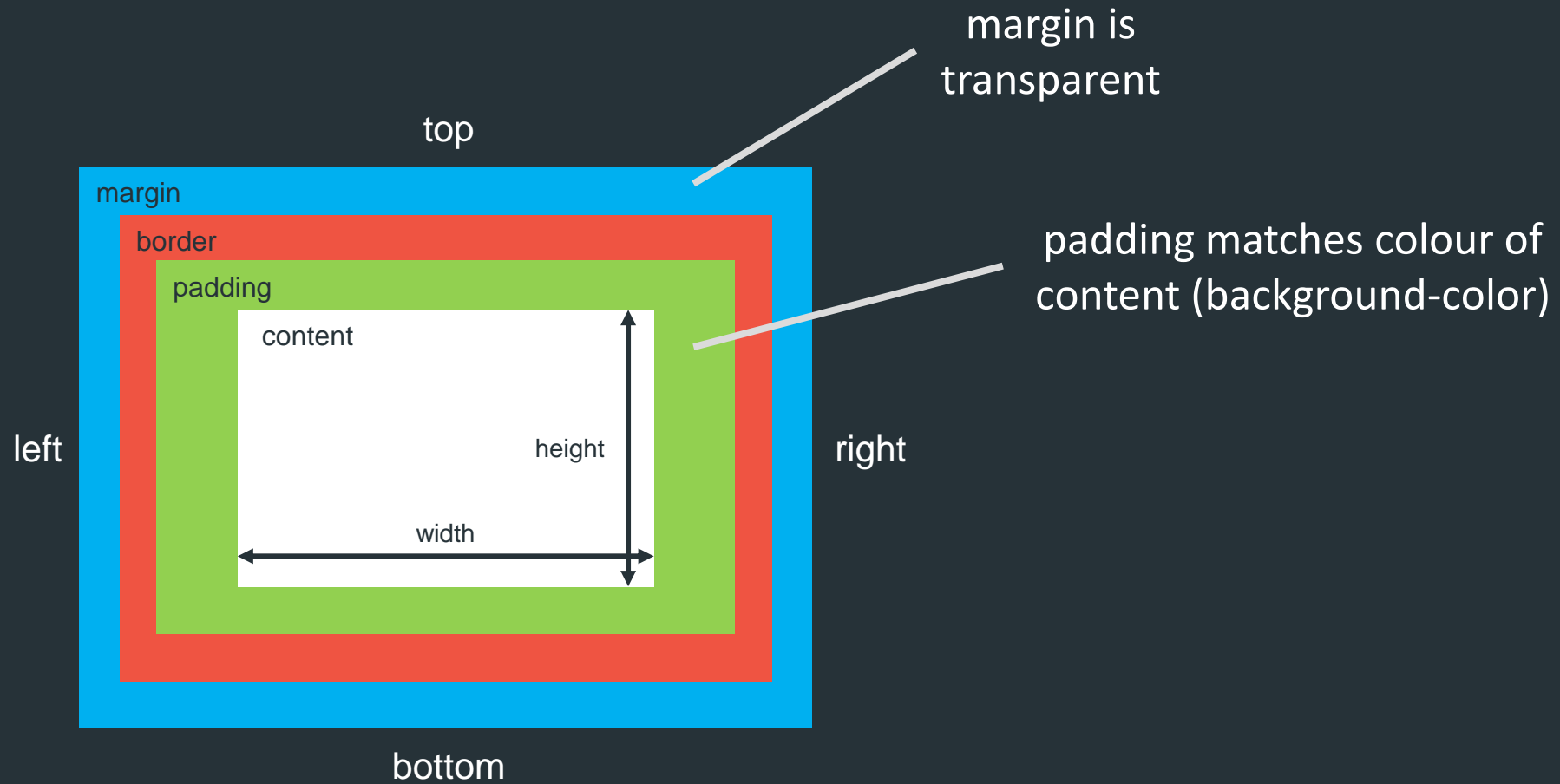
element	font-size	color
<p>Hello!</p>	16px	#FF0000
<p class="para">Hello!</p>	32px	#00FF00
<p class="para para1"> Hello!</p>	32px	#0000FF
<p class="para para1 para2"> Hello!</p>	24px	#0000FF
<p class="para1 para2"> Hello!</p>	24px	#0000FF

CSS Box Model

How elements can be sized and spaced in CSS.

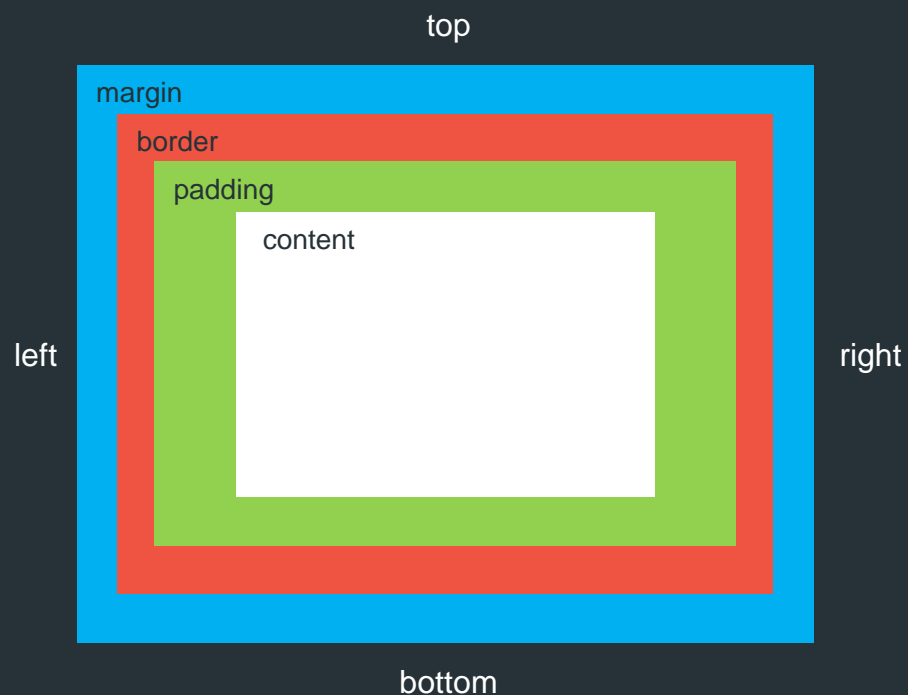


CSS Box Model



CSS Box Model

How to Change the Sizes of Padding and Margin



For margin, replace the word padding!

```
/* changes all sides */  
padding: 16px;
```

```
/* top+bottom, left+right */  
padding: 16px 12px;
```

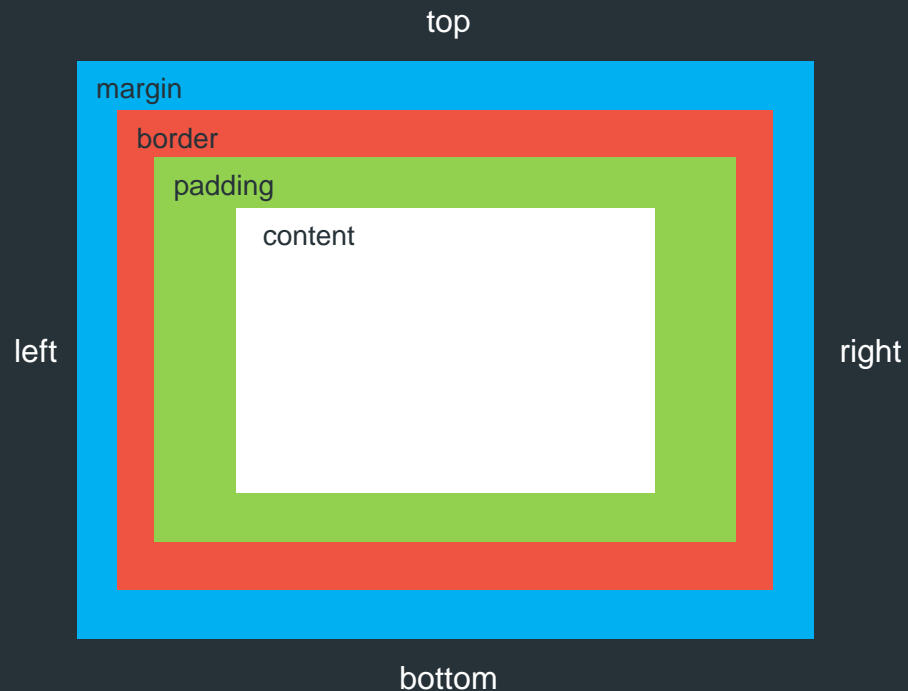
```
/* top, left+right, bottom */  
padding: 16px 12px 14px;
```

```
/* top, right, bottom, left */  
padding: 16px 15px 14px 13px;
```

```
/*changes individual sides*/  
padding-top: 16px;  
padding-right: 15px;  
padding-bottom: 14px;  
padding-left: 13px;
```


CSS Box Model

How to Change the Border



```
/* changes each property for all sides */  
border-color: #FF0000;  
border-style: solid;  
border-width: 16px;
```

```
/* for all sides: width, style, color*/  
border: 16px solid #FF0000;
```

```
/*changes individual sides (top, right,  
bottom, left */  
border-top-color: #FF0000;  
border-top-style: solid;  
border-top-width: 16px;
```

Styling Links

Default: the default style of the link

Visited: links that have already been clicked

Hover: mouse is over the link, not clicking it

Focus: when the user is using tab to select the link

Active: how it looks like when the user is clicking it

CSS Flexbox

A powerful tool to layout your elements in a single axis.

display: flex;

Only applies to the children of the element Flexbox is activated for!

CSS Flexbox

Flex Direction

```
#parent {  
  display: flex;  
  flex-direction: ____;  
}
```

```
p {  
  font-size: 32px;  
  font-weight: 1000;  
  background-color: #F0F00F;  
  padding: 10px;  
  margin: 5px;  
}
```

```
<div id="parent">  
  <p>1</p>  
  <p>2</p>  
  <p>3</p>  
  <p>4</p>  
  <p>5</p>  
  <p>6</p>  
</div>
```

flex-direction:
row;

1 2 3 4 5 6

flex-direction:
row-reverse;

6 5 4 3 2 1

flex-direction:
column;

1

2

3

4

5

6

flex-direction:
column-reverse;

6

5

4

3

2

1

CSS Flexbox

Justify Content

```
<div id="parent">  
  <p>1</p>  
  <p>2</p>  
  <p>3</p>  
  <p>4</p>  
  <p>5</p>  
  <p>6</p>  
</div>
```

```
#parent {  
  display: flex;  
  flex-direction: row;  
  justify-content: ____;  
}
```

```
p {  
  font-size: 32px;  
  font-weight: 1000;  
  background-color: #F0F00F;  
  padding: 10px;  
  margin: 5px 0px 5px 0px;  
}
```

justify-content:
flex-start;

1 2 3 4 5 6

Push all elements together, remaining space placed to the right of the elements.

justify-content:
flex-end;

1 2 3 4 5 6

Push all elements together, remaining space placed to the left of the elements.

justify-content:
center;

1 2 3 4 5 6

Push all elements together, remaining space divided by 2 and placed to the sides of all the elements.

justify-content:
space-between;

1 2 3 4 5 6

Divide remaining space by $n-1$, place between elements.

justify-content:
space-around;

1 2 3 4 5 6

Divide remaining space by $n*2$, place to the sides of each element.

justify-content:
space-evenly;

1 2 3 4 5 6

Divide remaining space by $n+1$, place between each element, to the left of the first element, and the right of the last element.

CSS Flexbox

Align Items

```
<div id="parent">
```

```
<p>1</p>
```

```
<p>2</p>
```

```
<p>3</p>
```

```
<p>4</p>
```

```
<p>5</p>
```

```
<p>6</p>
```

```
</div>
```

```
#parent {
```

```
height: 200px;
```

```
border: 16px solid #FF0000;
```

```
display: flex;
```

```
flex-direction: row;
```

```
justify-content: space-between;
```

```
align-items: ____;
```

```
}
```

```
p {
```

```
font-size: 32px;
```

```
font-weight: 1000;
```

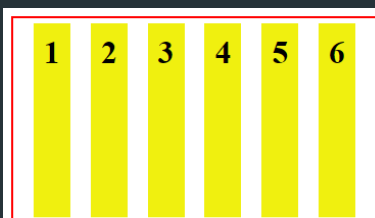
```
background-color: #F0F00F;
```

```
padding: 10px;
```

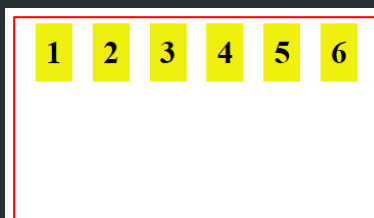
```
margin: 5px 0px 5px 0px;
```

```
}
```

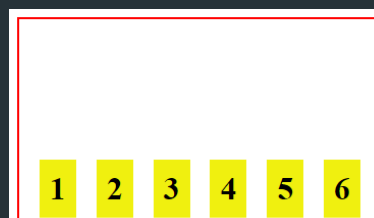
align-items:
stretch;



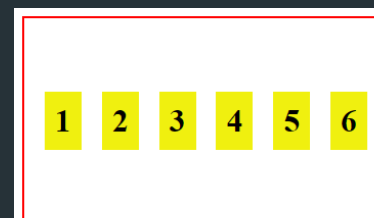
align-items:
flex-start;



align-items:
flex-end;



align-items:
center;



CSS Grid

A powerful tool to layout your elements in a 2D grid.

`display: grid;`

Only applies to the children of the element Grid is activated for!

Additional Resources:

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids

Media Queries

Using breakpoints to develop layouts or change content size for.

~~Mobile: 320px to 480px~~

Tablets: >480px to 768px

Notebooks: >768px to 1024px

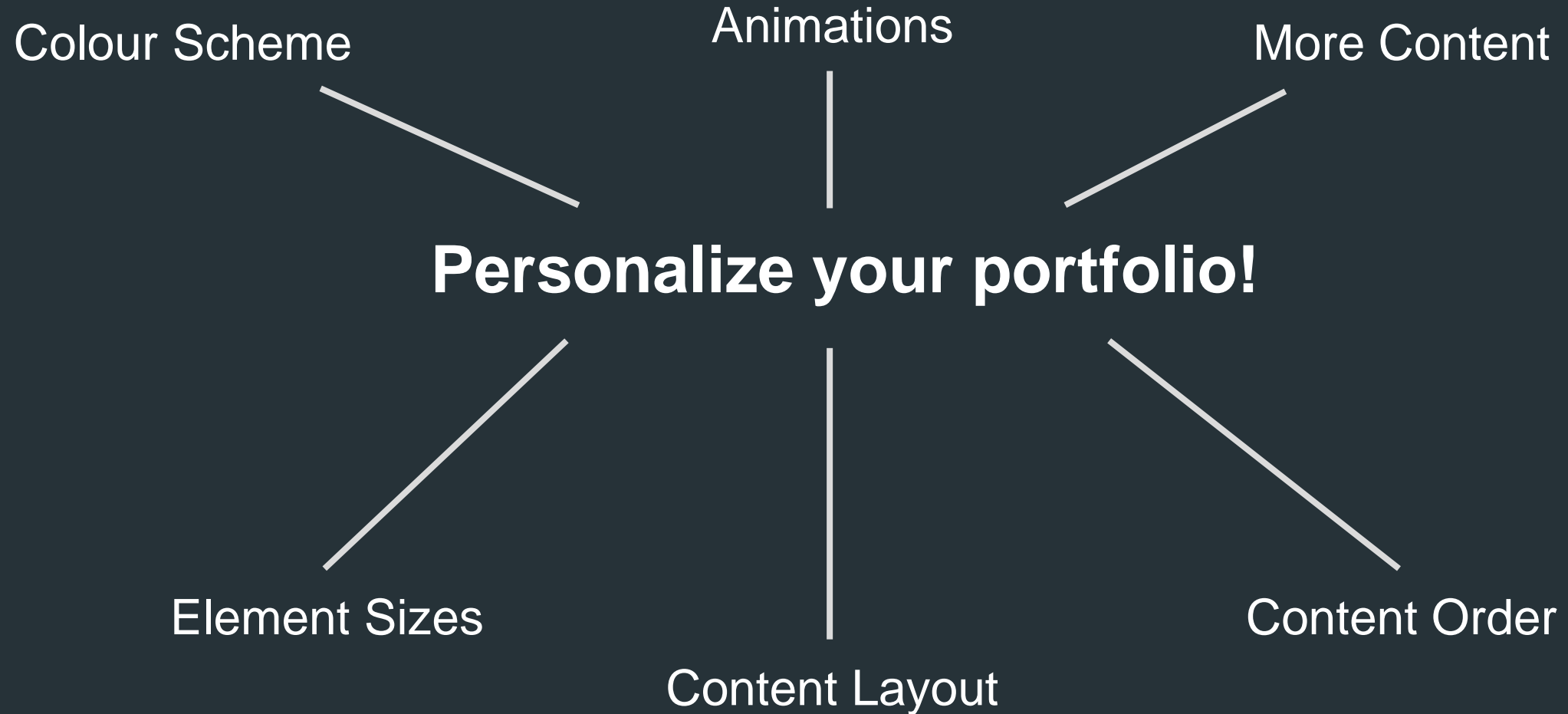
Large Screens: >1024px to 1200px

Extra Large Screens: >1200px

```
@media (min-width: 480px) {
```

```
  html {  
    font-size: 20px;  
  }  
}
```


Future Learning – Challenges



**Thank You For
Attending!**

Any Questions?

