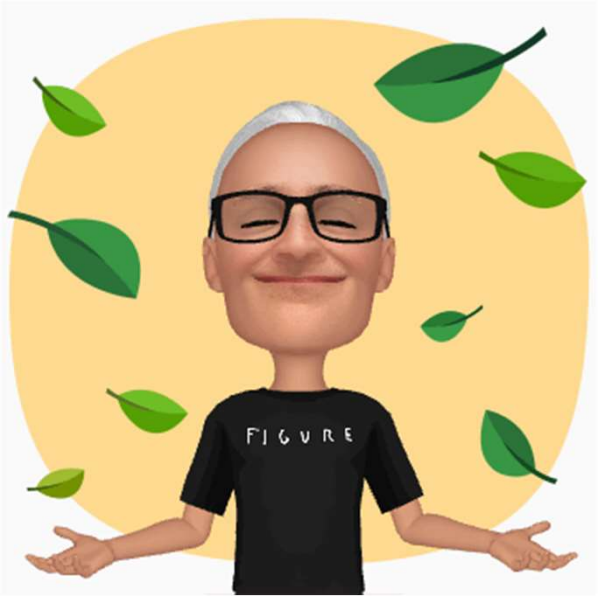
MODULE 3

Introduction to CSS Selectors and Layout Positioning

Welcome to Module 3!

- Front end layout
- MVC Design pattern
- Forms and forms processing
- Sessions, Flash scope.
- Data validation.
- User Authentication.



Let’s talk HTML

- What does HTML stand for?
- HMTL: Hypertext Markup Language
- Document is hierarchical in design.

Simple HTML Document

```
<!DOCTYPE html>
<html>
<head>
   <title>Box Model</title>
   <link rel="stylesheet" href="box-model.css" />
</head>
<body>
   <img id="slug_1" src="img/slug.png" alt="slug">
   padding: 50px; margin: 0; border: 3px solid black;
   <div>
      <img id="slug_2" src="img/slug.png" alt="slug">
      padding: 0; margin: 50px; border: 3px solid black;
   </div>
</body>
</html>
```

CSS: Cascading Style Sheet

- CSS is the way we tell HTML how to format itself
- CSS can be included by:
  - inline on any element with a style attribute
  - within the document under a <style> element
  - external document reference with a <link href=> element

CSS in document

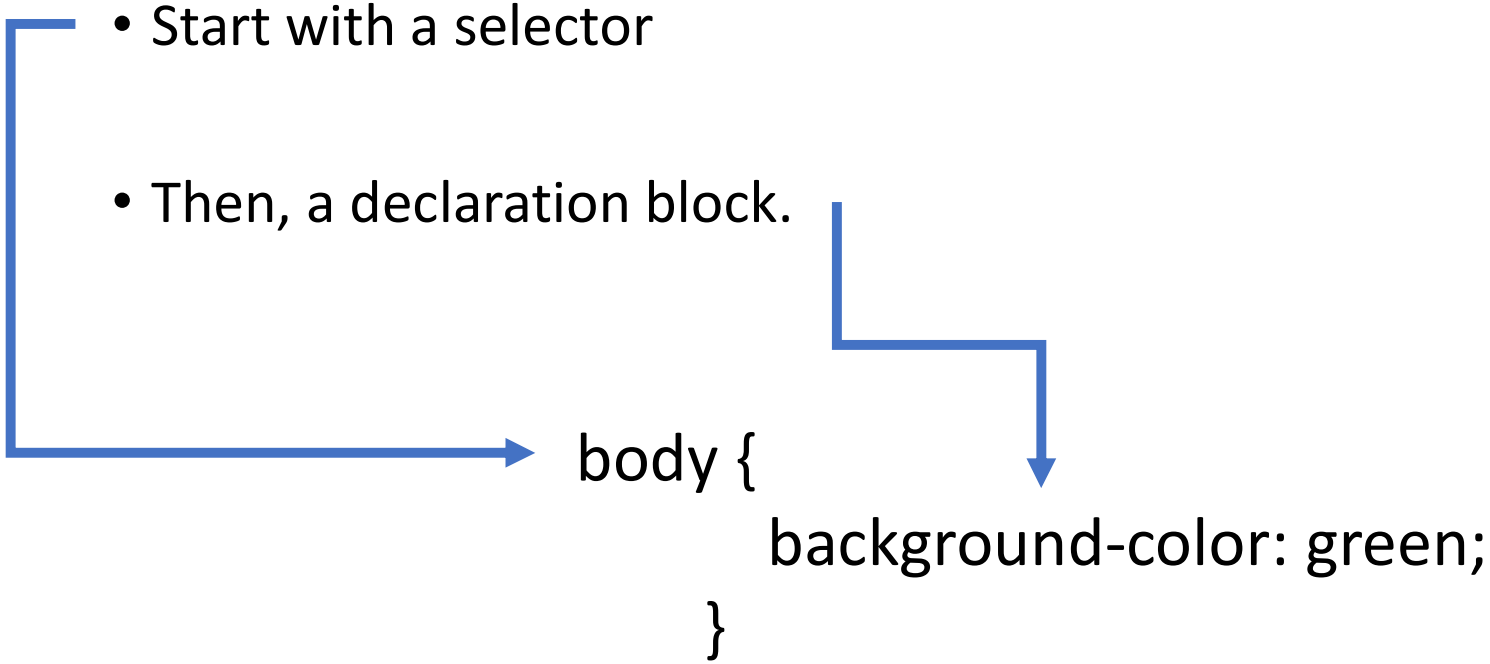
```
<body style="background-color:green;">

 <style>
       body {
          background-color: green;
       }
    </style>

 <link rel="stylesheet" href="box-model.css" />
```

Well, that escalated quickly…

- Start with a selector
- Then, a declaration block.



```
body {
    background-color: green;
}
```

CSS Selectors

Type	Example
Universal	*

Element	body {
	background: green;
	}

Id	#slug_1

Class	.bigBorder{
	border: double;
	}

Fancy CSS Selectors

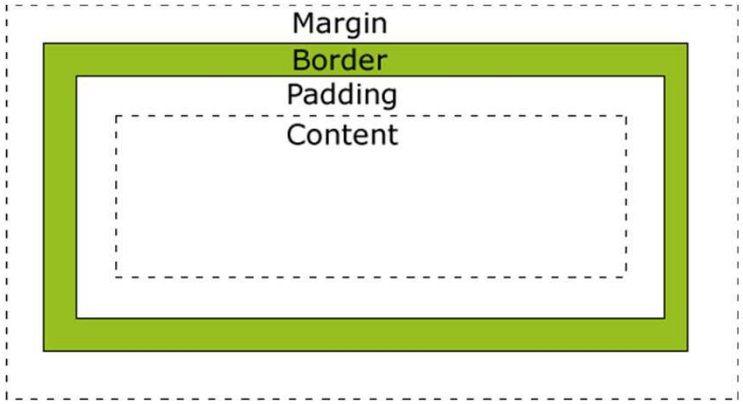
- Descendant selector: [a space]
  - ul li <= applies to list elements inside an unordered list
  - div .foo <= applies to any element with the class foo inside a div
- Direct Child selector: >
  - div > span <= applies to any span that is a **direct child** of a div
- Adjacent Sibling selector: +
- General Sibling selector: ~

- pseudo-class
  - a: hover

Box Model

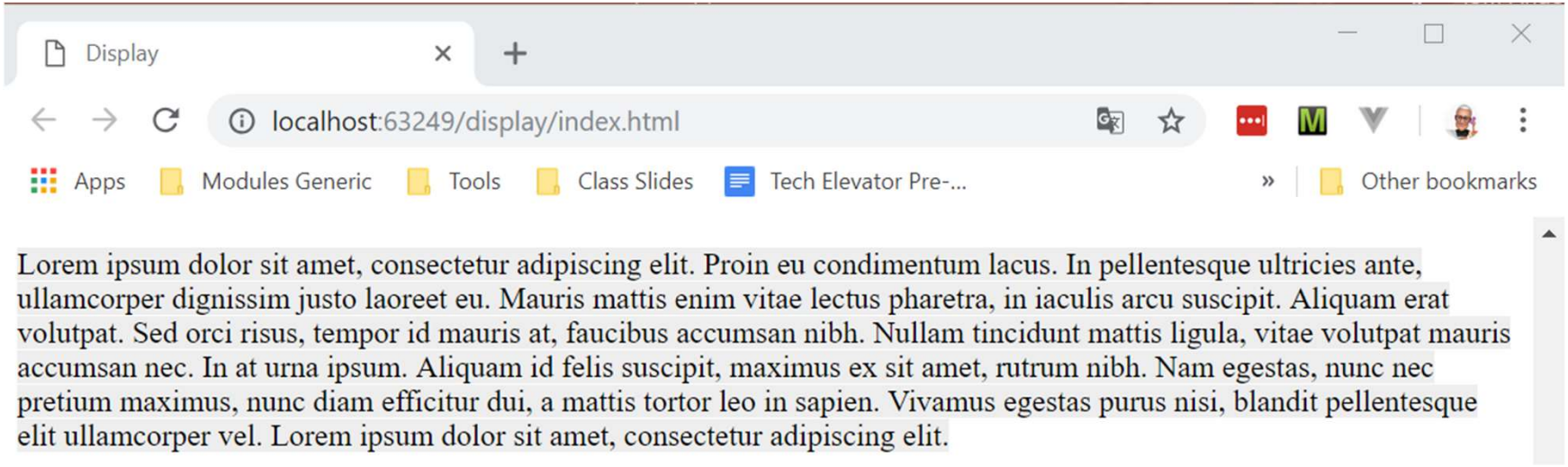
Box Model

- Every element in web design is a rectangular box.
- We use the content, padding, border, and margin to calculate the amount of space that an element takes up.
- Margin is the space **outside something**. It does not affect the size of the box but affects other content that interacts with the box.
- Padding is the space **inside something**.



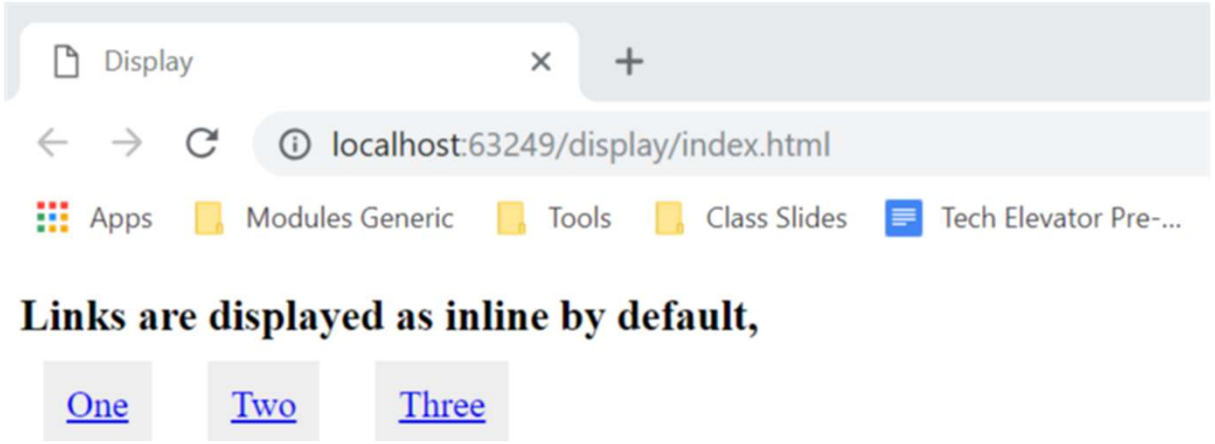
Block or Inline?

- Block elements (div, h1-h6, p, form, header, footer)
  - Always starts on a new line
  - Takes up full width available



Block or Inline?

- Inline elements (a, span, img)
  - Do not start on a new line
  - Takes up only as much width as necessary
  - It accepts margin & padding while ignoring height & width.



inline-block is very similar to inline but **it allows you to set a width and height**.

Positioning elements on a page

- **Normal** flow of a page:
  - Left to right
  - Top to bottom
- **Relative** position means **relative to where it would otherwise be positioned in the normal flow**
  - Can set top, right, bottom, and left
- **Absolute** position will place the element relative to the parent ancestor (i.e. containing element) **exactly where you specify**.
  - These elements are removed from the flow of the page.
  - Setting both *top* and *bottom*, or both *left* and *right*, you can "stretch" an element's dimensions.
- **Fixed** position is **relative to the browser window** and does not scroll with the page.
  - setting the *top, right, bottom, and left*

Finally, float.

The float property specifies if the element should be taken from the normal flow and placed along the left or right-side of the container. Text and inline elements will wrap around it.

- **none** element does not float
- **left** element floats to left of its container
- **right** element floats to right of its container
- **inherit** element inherits float direction of its parent
- Floated elements automatically display as block

WHAT QUESTIONS DO YOU HAVE?

Reading for tonight:

HTML and CSS Overview