

Module 3 - Lecture 2

# CSS Selectors and Layout



# CSS: Selectors



<ul> element is  
parent of <li>  
elements and child  
of <main>

<li> elements are  
children of <ul>

<li> elements are  
siblings of each  
other

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>
  <header>
    <h1>Header</h1>
  </header>

  <main>
    <ul>
      <li>Item One</li>
      <li>Item Two</li>
      <li>Item Three</li>
    </ul>
  </main>
</body>

</html>
```

## Aside: HTML id attribute

- A unique, case-sensitive name to identify an element.
- Enables CSS and JavaScript to select elements directly.
- Anchor elements can navigate directly to another element by id.

```
<a href="#paraObviousPoint">Link to p</a>
```

```
<p id="paraObviousPoint">
```

```
  Above point sounds a bit obvious.
```

```
  Remove/rewrite?
```

```
</p>
```

## Aside: HTML class attribute

- A space separated list of case-sensitive class names for an element.
- Enables CSS and JavaScript to select elements that share a class name.

```
<p class="note editorial">
```

```
Above point sounds a bit obvious.
```

```
Remove/rewrite?
```

```
</p>
```

# CSS Selector Types

Name	Syntax	Example	Description
All / Wildcard	*	*	Every element
Element	element	h1	All h1 elements
ID	#elementId	#application	Element with the id “application”
Class	.className	.btn-primary	Elements with the class “btn-primary”
Descendant	element element	main li	li elements that are descendants of a main element
Child	element > element	ul > li	li elements that are children of a ul element
Attribute	[attribute=value]	[id=application]	Elements with an attribute id=”application”
Surrounding	+ - or ~		

# Combining CSS Selectors

- Selectors can be combined together to indicate multiple selectors have to be true (AND)

## **#content div > ul.list**

- SELECT a ul element with class name “list”
  - AND the ul element is a child of a div element
  - AND the div element is a descendant of an element with an id “content”
- 
- Selectors can be combined together with a comma to indicate this selector or that selector have to be true (OR)

## **div, span, #content**

- SELECT div elements, span elements, or an element with id “content”

# Pseudo-Class Selectors

- Targets an element based on an element's state. Prefaced with a colon.

Some examples:

`a:visited` -> anchor elements that have been visited

`input:disabled` -> input elements that are currently disabled

`table tr:nth-child(even)` -> even numbered rows in a table



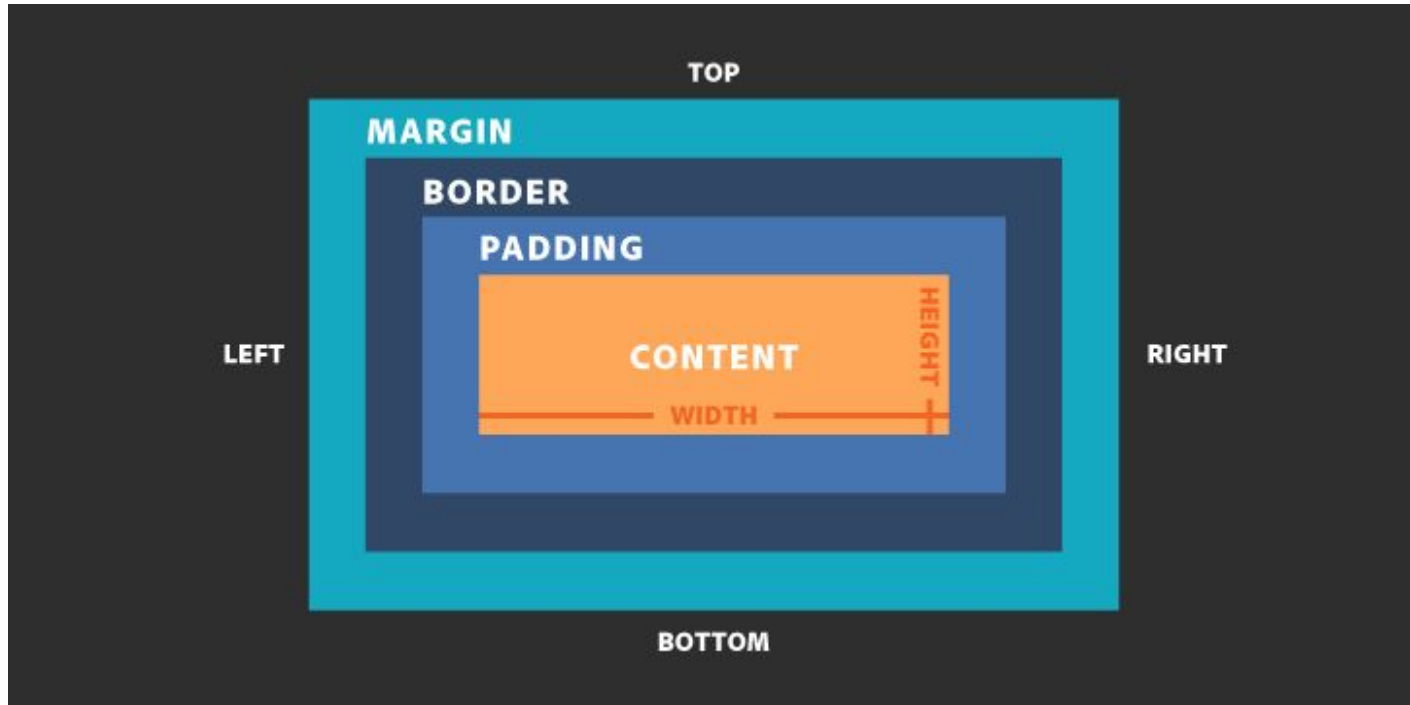
# Specificity

- Specificity is a weight that is applied to a given CSS declaration, determined by the number of each selector type in the matching selector. When multiple declarations have equal specificity, the last declaration found in the CSS is applied to the element.
- Hierarchy of specificity (weight):
  - Inline styles (1000)
  - ID selectors (100)
  - Class, Attribute, Pseudo selectors (10 each)
  - Element selectors (1 each)
- Applying **!important** to a selector will override specificity.
  - Best to avoid using this!

# CSS: Layout

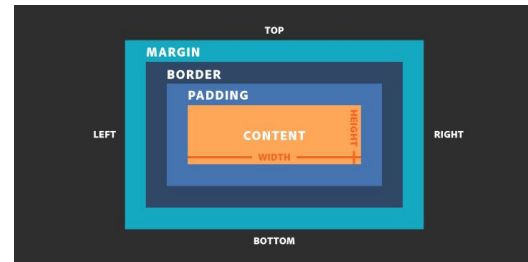


# CSS Box Model



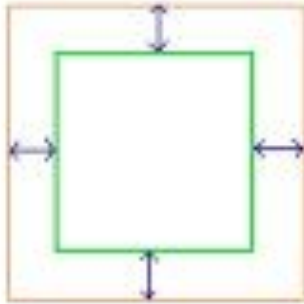
# CSS Box Model

- Every element in a browser is rendered as a box.
- Every element is made up of content, padding, border, and margin.
- Margin, Border, and Padding each have 4 component widths, a top, right, bottom, left width (think clockwise from the top).
  - Use margin to separate the block from things outside it
  - Use padding to move the contents away from the edges of the block.

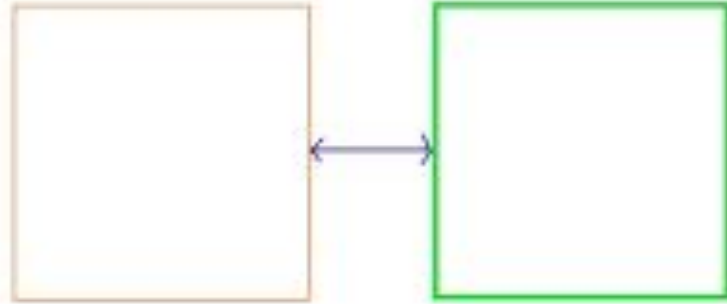


# CSS Box Model

- Use margin to separate the block from things outside it
- Use padding to move the contents away from the edges of the block.



**Padding**



**Margin**



# Inline vs Block

 **WEB BASICS** CSS

`display: inline`

inline


inline

inline

inline

Displays an element as an inline element. Any height and width properties will have no effect

[@samanthaming](#) [samanthaming.com](#) [samantha\\_ming](#)

 **WEB BASICS** CSS

`display: inline-block`


inline-block

inline-block

inline-block

Displays an element as an inline-level block container. You CAN set height and width values.

[@samanthaming](#) [samanthaming.com](#) [samantha\\_ming](#)

 **WEB BASICS** CSS

`display: block`

block

block

Displays an element as a block element. It starts on a new line and takes up the whole width.

[@samanthaming](#) [samanthaming.com](#) [samantha\\_ming](#)

## Default of inline\*

- span
- a
- img

\*Not exhaustive lists

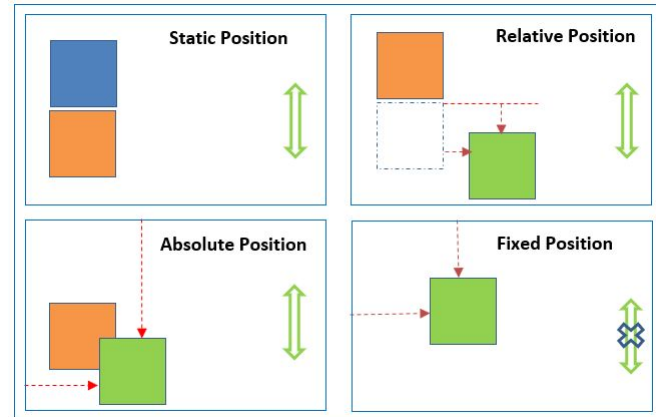
## Default of block\*

- div
- h1
- p
- section
- li



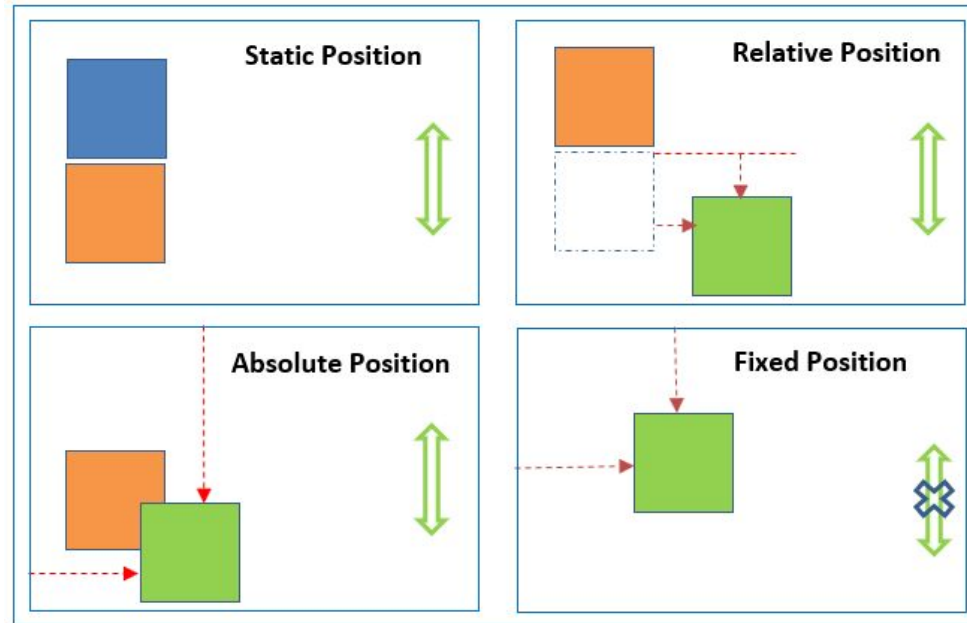
# Position

- The **normal flow** of a page is for elements to appear left to right and top to bottom based on the order in which they appear in the HTML document and the rules of block and inline display. This is also referred to as **Static**.
- There are options for other types of positioning. **Relative**, **Absolute**, and **Fixed**.



# Relative Position

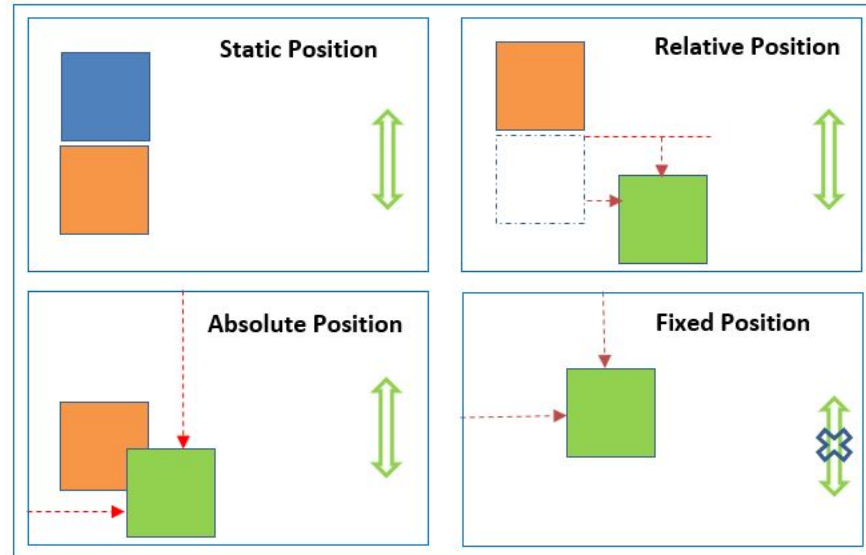
- **Relative** position means **relative to where it would otherwise be positioned in the normal flow**.





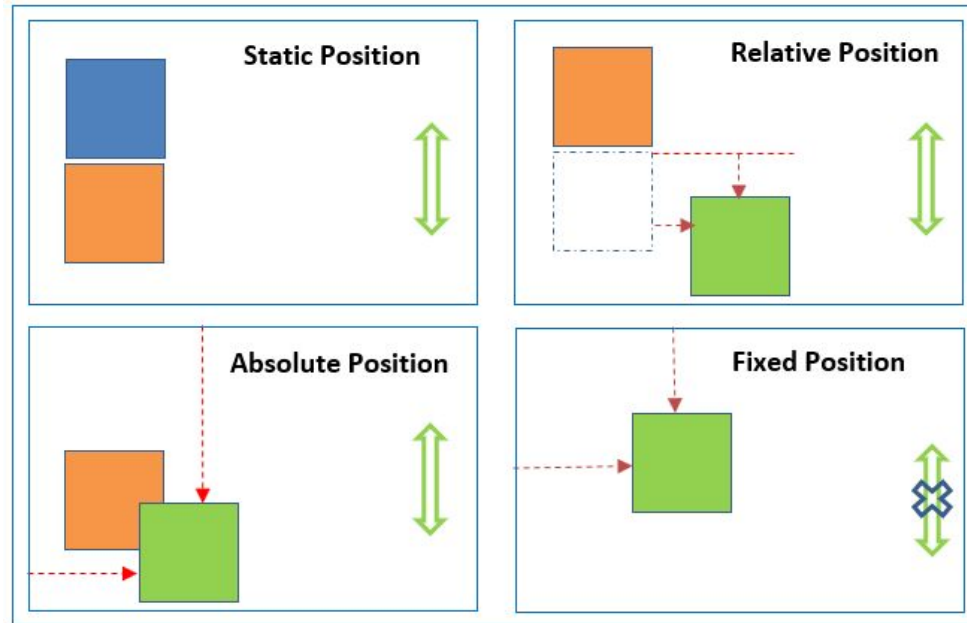
# Absolute Position

- **Absolute** position places the element relative to the parent ancestor—that is, the containing element—**exactly where you specify**.
- These elements are removed from the flow of the page.



# Fixed Position

- **Fixed** position is **relative to the browser window** and does not scroll with the page.



# 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 wrap around it.

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

## float: none;



## float: left;



## float: right;



QUESTIONS?

