

01. Intro to HTML and CSS

by Abdul Rahman Rehmtulla

Housekeeping:

- GitHub Link: <https://github.com/bijuice/kamilimu-webdev>
- Lessons will be available on branches.
- Recordings of the lessons will be found [here](#)
- Feel free to unmute and ask questions.
- There are no dumb questions.
- You can always ask me to slow down.

Intro to Webdev

- The internet runs on HTML, CSS and JavaScript
- HTML is a markup language - defines structure of content
- CSS is a styling language - defines the look and feel of the site
- JavaScript is a scripting language - adds interactivity
- Think of it like a car:
 - The body is HTML
 - The styling and paint is CSS
 - JavaScript is the engine.

HTML

- HTML consists of elements.
- Elements are defined within tags e.g
<h1>Hello World </h1>
- The structure of an HTML document can be seen in the figure
- Some tags are self-closing e.g.
- Many different types of elements exist

```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5  </head>
6  <body>
7
8  </body>
9  </html>
```

Semantic HTML

- Most HTML content can be put in a `<div>` element. This is a bad idea.
- Semantic HTML is a practice of writing HTML in a way that conveys meaning
e.g heading = `<h1>`, footer = `<footer>`, section = `<section>`, sidebar = `<aside>`,
navigation = `<nav>`, article = `<article>`
- This is important because:
 - a. Accessibility.
 - b. Search Engine Optimization (SEO).
 - c. Code readability and maintainability.

CSS

- CSS stands for Cascading Style Sheets.
- Allows us to determine the look and feel of our content.
- Easy to learn but hard to master.
- Can be frustrating but it helps to understand what is going on.

CSS Syntax

- Defined using a selector and a block.
- Properties are separated with a semicolon.
- Some properties have shorthands to declare multiple values.

```
h1 {  
  color: ■ green;  
  border: solid 1px ■ red;  
}
```

Including CSS in HTML

- CSS can be written in three places (listed in order of priority)
- In-Line CSS - this is within a tag inside the style attribute
- Internal CSS - defined within the <style> tag within the <head> element.
- External CSS - defined in a separate file and imported using the <link> element in the <head>

```
<body>
  <h1 style="color: green; font-size: 20px">
    |   Hello World
  </h1>
</body>
```

CSS Selectors - Basics

- CSS selectors are used to determine which element we are targeting
- CSS selectors can be grouped by separating them with a “,”
- Universal selector - “*”
- Type selector e.g p, h1, div, img
- Class selector- prefixed with “.” e.g .card, .custom-header
- ID selector - prefixed with a “#” e.g. #person-card, #avatar
- Attribute selector - encased in square brackets “[]” e.g [type=”text”]

CSS Selectors - Pseudo Classes

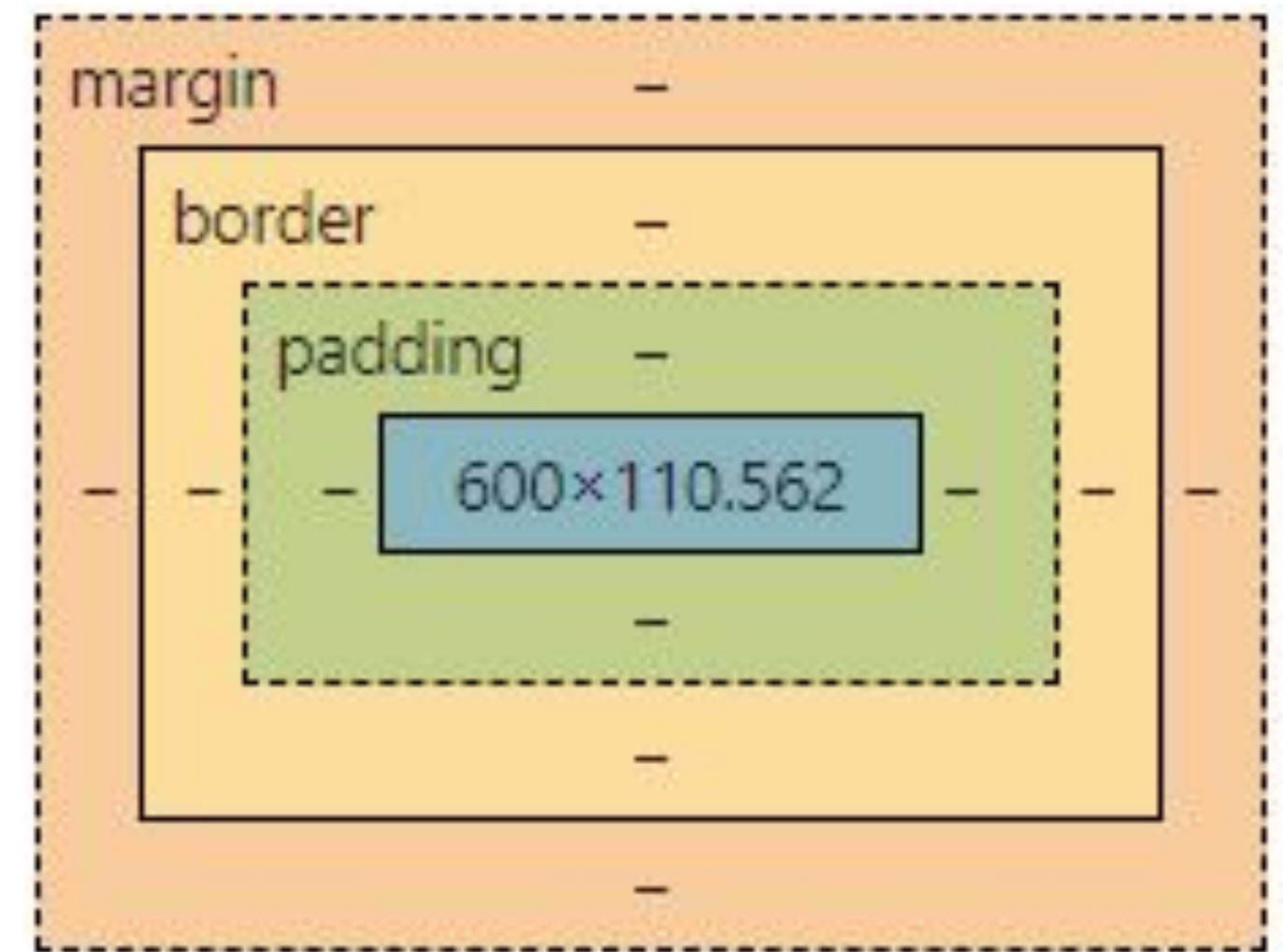
- Pseudo-classes are used to define a special state of an element.
- Separate a selector using a colon e.g p:hover, input:valid
- :hover - active when an element is hovered over
- :valid or :invalid - when an input is valid or invalid
- :disabled - when an element (such as a button) is disabled
- :first-child or :last-child - when an element is the first or last element in its group
- :even or :odd - to select even or odd elements
- Learn more [here](#)

CSS Selectors - Pseudo Elements

- Different from Pseudo Classes
- These are used to style specific parts of an element
- Are denoted with a double colon :: e.g p::before, div::after, h1::first-letter
- Learn more [here](#)

CSS Box Model

- Every HTML element has a **margin**, **border**, **padding**, and internal space.
- These govern the size of each element.
- **Margin**: space outside the element
- **Padding**: space inside an element
- **Border**: an outline of the element



CSS Units

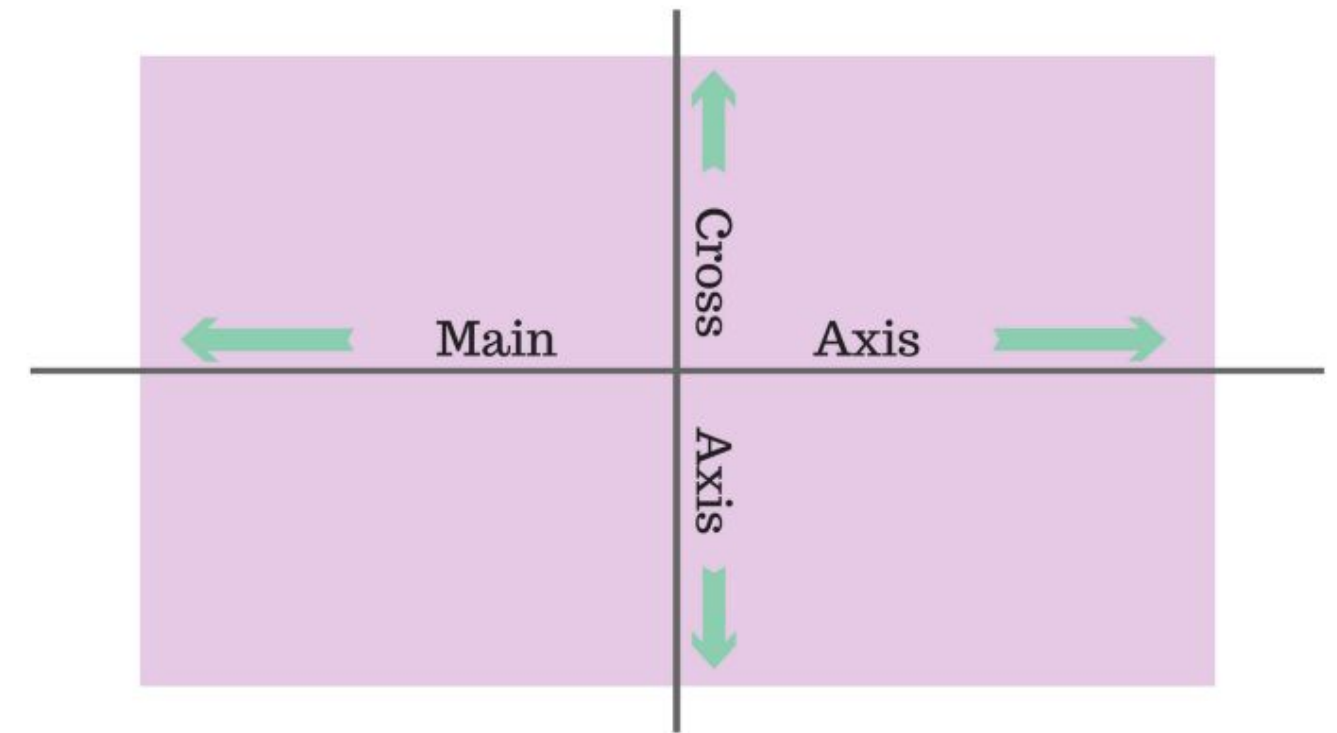
- CSS units determine the dimensions of HTML elements and their contents
- **Absolute** units are constant and do not change depending on scale of the document e.g **px** (pixels), **cm** (centimeters), **in** (inches), **pt** (points)
- **Relative** units scale with the elements and the document scale e.g **em** (font-size of parent element), **rem** (font-size of root element), **vw** (view width), **vh** (view height), **%** (percent), **ch** (characters)
- Prefer using relative units (**em**, **rem**) for height most of the time to make your app responsive.
- Use **ch** when determining widths of elements with text
- Use **%**, **vh**, **vw** for ratios
- Use **px** when you're sure something should be the exact height or width e.g images

CSS Flex

- The normal flow is where elements are placed one after the other vertically
- Flex allows us to use a flexible responsive layout
- Properties of note:
- **flex-direction** - the direction of flow of elements. e.g row, col, row-reverse, col-reverse
- **gap** - the spacing between elements
- Arranging content within a flexbox is determined by **justify-content** and **align-items**
- **flex-wrap**

CSS Flex: Justify & Align

- **justify-content** arranges elements along the main axis
- **align-items** arranges elements along the cross axis
- The main axis and cross axis change with the flex direction. **row** - (main axis = horizontal, cross axis = vertical). **col** - (main axis = vertical, cross axis = horizontal)
- Possible values: **start, end, space-between, space-around** . Learn more [here](#)



CSS Transitions

- Used to animate changes to an element.
Usually when an event occurs (such as hovering over an element).
- Determined by **transition-property**, **transition-duration**, **transition-timing-function**, **transition-delay**
- Just use **transition** property for shorthand.
- Transitions can be grouped with a comma

```
.box {  
  width: 100px;  
  height: 100px;  
  background-color: blue;  
  transition: width 2s, height 2s, transform 2s;  
}  
.box:hover {  
  width: 200px;  
  height: 200px;  
  transform: rotate(45deg);  
}
```


Performance Considerations

- Most transition properties (such as **height**, **width**, and **position**) cause the entire page to be **repainted** and **reflowed** this means that the browser must recalculate all properties of the elements on the page.
- Limit animation properties to **transform** and **opacity**
- If you really need to animate reflow properties consider the **FLIP** method. Learn more about that [here](#)

Questions?

Homework

- Create a portfolio using CSS and HTML
- Get inspiration from:
 - a. <https://dribbble.com/search/portfolio>
 - b. And if you're feeling adventurous... awwwards.com