

An abstract network diagram on a black background. It features numerous small circular nodes, some of which are highlighted with a white ring. These nodes are interconnected by a web of thin, curved lines in shades of orange, teal, and yellow. The lines create a sense of dynamic movement and connectivity, with some lines forming dense clusters and others extending across the frame. A vertical white line is positioned to the right of the main title text.

# INTERACTIVE WEB DEVELOPMENT

Pelin Mutanguha

# CSS BASICS



---

# WHAT IS CSS?

- Stands for Cascading Style Sheets.
  - CSS describes how HTML elements are displayed on a page.
  - Specifies how the document is styled, laid out, variations in display for different devices and screen sizes, etc...
  - CSS Styles are usually in separate (.css) files but can also be embedded in html documents
-

---

# CSS SYNTAX

- CSS is a rule based language
- CSS styles are declared as rules structured in block
- CSS rules have 3 parts:
  - a selector,
  - a property: identifier that indicates stylistic features
  - a value: indicates how to style the property
- CSS Selectors :
  - Universal Selector
  - Type Selectors (Html Elements),
  - Class Selectors,
  - ID selectors

```
selector { property: value }
```

```
h1 {  
    color: ■ #000;  
    text-align: center;  
}
```

---

# APPLYING CSS STYLES TO HTML

- CSS styles can be applied to html in 4 ways:
  - Inline styles
  - Internal styles
  - External styles
  - Import styles

---

# INLINE STYLES

- Applied directly in html elements
- Use style attribute in html element to apply styles to that element
- Inline styles are difficult to reuse on other elements and pages since they're applied on elements

```
<nav>

  <h2 style="background: ■ #ffae00; padding:5px;">Navigation</h2>

  <!-- internal link -->
  <a href="contacts/second.html" target="_blank">Second Page</a>
```

---

# INTERNAL STYLES

- Internal styles are placed within the html document
- To embed CSS styles in you HTML you use `<style></style>`
- The style tag must be placed in the `<head>` section of your html document
- Not helpful when u want to apply same styles across different pages

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My First HTML Page</title>
    <style>
      body {
        background-image: url("images/bg");
        color: #ecec;
      }
      h1{
        color: #000;
      }
    </style>
  </head>
  <body>
    <h1>My First HTML Page</h1>
  </body>
</html>
```

---

# EXTERNAL STYLES

- CSS rules are placed in a separate file (.css file)
  - HTML document is linked to the CSS file using **<link>** tag
  - Link tag is placed inside the head section of html page
  - It's the most common method for adding CSS styles to your document
  - It allows reusability of CSS styles across different pages
- 

```
*{  
    font-family:'Times New Roman', Times, serif  
}  
h1 {  
    color: ■ #000;  
    text-align: center;  
}
```

```
<head>  
    <title>My First HTML Page</title>  
    <link rel="stylesheet" href="css/styles.css">  
</head>  
<body>  
    <h1>My First HTML Page</h1>
```



---

# IMPORT STYLES

- Used to import any external CSS styles in your html document
- <style> tag is used
- Style tag is placed in the head tag of the html document
- **@import** is used to specify the location (internal or external) of the CSS styles
- @import can also be used in a CSS file to import other external styles

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My First HTML Page</title>
    <style>
      @import url("css/styles.css");
    </style>
  </head>
  <body>
    <h1>My First HTML Page</h1>
```

---

# VALUES AND UNITS

- Numeric values
    - Width: 80%, height:10em, border-width: 5px, animation-iteration-count: 5
    - Transform: rotate(180deg), animartion-duration: 750ms;
    - Length
      - Absolute: fixed size, not affected by values in related elements (eg. Px, cm, mm,..)
      - Relative: relational sizing (eg. Em, rem,)
  - Function values
    - Background-image: url('bg.jpg')
  - Initial value
-

---

# COLOR PROPERTY

- Basic color keywords (blue, red, green, yellow,..)
  - Extended color keywords (aliceblue, lightgray,..)
  - RGB (red, green, blue)
    - Hexadecimal: #rrggbb
      - Shorthand & longhand
    - Rgb() function : rgb(200,200,200), rgba(50%, 50%, 70%)
      - Alpha channel for opacity: rgba(134,156,145,0.5)
  - HSL(Hue, Saturation, Lightness)
  - Useful links:
    - <https://coolors.co/>
    - <https://color.adobe.com/>
-

---

# UNIVERSAL SELECTOR

- The universal selector (\*) selects all HTML elements on the page.
- It matches any type
- An asterisk can also be followed by a selector

```
*{  
  font-family:'Times New Roman', Times, serif  
}
```

---

# ID SELECTORS

- The id selector uses the id attribute of an HTML element to select a specific element
- The id of an element is unique within a page, so the id selector is used to select one unique element
- To select an element with a specific id, write a hash (#) character, followed by the id of the element

```
#second{  
    font-size: 30px;  
    color: ■ #b6b6f1;  
    border: 2px dotted ■ #000;  
}
```

```
<body>  
    <h2 id="second">This is the second page</h2>  
    <a href="..first.html">Go to First</a>  
</body>  
.. -
```

---

# CLASS SELECTORS

- selects HTML elements with a specific class attribute
- allow you to apply a style to a given *class* of an element.
- Class name has to be added to an element using **class** attribute to your html
- An element can have more than one class
- CSS Class rule is declared using a using a dot (.) followed by the class name

```
<p class="intro large"> this is my <strong>first</strong> paragraph  
  is simply dummy text of the printing and typesetting industry.  
  Lorem Ipsum has been the industry's standard dummy text ever  
  since the 1500s, when an unknown printer took a galley of type  
  and scrambled it to make a type specimen book. It has survived  
  not only five centuries, but also the leap into electronic typesetting,  
  remaining essentially unchanged.  
</p>  
<p class="large"> this is my second paragraph</p>
```

```
.intro {  
  background: #e2ebf3;  
  color: #000;  
  padding: 3px 3px;  
  line-height: 25px;  
}  
  
.large {  
  font-size: 18px;  
}
```

---

# PSEUDO-CLASSES

- A pseudo-class is a selector that selects elements that are in a specific state
- tend to act as if you had applied a class to some part of your document
- Pseudo-classes are keywords that start with a colon
  - *:pseudo-class-name*
- User action pseudo-classes (:hover, :focus)
- Link pseudo-classes: (:link, :hover, :active, :visited, :focus)
- For more: [shorturl.at/enKN9](https://shorturl.at/enKN9)

```
<section>
  <p> this is my <strong>first</strong> paragraph
    is simply dummy text of
    the printing and typesetting industry.
  </p>
  <p> this is my second paragraph</p>
</section>
```

```
p:first-child{
  background-color: yellow;
  border: 1px solid rgb(104, 104, 131);
}
```

---

# COMBINATORS

- Provides a way to combine many selectors to give more specific styles to elements
- There are many types:
  - **Descendent:** `ul li a` (targets a link that's inside of a li and inside of an ul)
  - **Child combinator:** `section > p` (target a paragraph that are direct children of section)
  - **Adjacent sibling combinator:** `p + img` (targets images that are immediately preceded by a paragraph)
  - **General sibling combinator:** `p ~ img` (targets images that come anywhere after a paragraph)



---

# SPECIFICITY

- Determines how browsers will decide which CSS rule takes precedence
  - Precedence order:
    1. Universal
    2. Element
    3. Class
    4. Id
-

# CSS BOX MODEL

Everything in CSS has an invisible box around it. These boxes are key to CSS layout.

---

# BLOCK AND INLINE BOXES

- **Block boxes:**

- break on a new line
- Occupy the entire width of its container
- Width and height are respected
- Eg: h1, div, p

- **Inline Boxes:**

- The box will not break onto a new line
  - The width and height properties will not apply
  - Eg: a, span, ..
-

---

# CSS BOX MODEL

- **Content box:** sized with *Height* and *Width* properties
- **Padding box:** sized with *padding* property
- **Border box:** sized with *border* property
- **Margin Box:**



---

# CODING DEMO: BOX MODEL

```
mirror_mod = modifier_ob.  
Get mirror object to mirror.  
mirror_mod.mirror_object =
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

-- OPERATOR CLASSES --

```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

```
context):  
context.active_object is not
```

# TYPOGRAPHY

Arranging type for  
readability and to engage  
and communicate with  
reader

---

# TYPE FACE

- Font face/Typeface: a set of fonts designed with common characteristics , composed with glyphs
  - Font families:
    - **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
    - **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
    - **Monospace** fonts - here all the letters have the same fixed width. They create a mechanical look.
    - **Cursive** fonts imitate human handwriting.
    - **Fantasy** fonts are decorative/playful fonts
  - Adding fonts: font-family: sans-serif, roboto, helvetica;
  - You can use online fonts. (google fonts, adobe fonts,...)
-

---

# FONTS

- Web Safe Fonts: fonts that are universally installed across all browsers and devices
    - Arial, Verdana, times new roman,....
  - Fallbacks Fonts: alternative fonts to be applied when the selected one isn't available
  - Font properties:
    - Font family: verdana, sans-serif
    - Font size: 16px;
    - Font weight: 400;
-



# CSS LAYOUT

controlling where web  
page elements are  
positioned relative to their  
default position in normal  
layout flow

---

# CSS LAYOUT

- There are many way to layout elements:
    - **Normal flow:** normal way of how elements are added to the web page
    - **The display property:** values such as block, inline or inline-block can change how elements behave in normal flow
    - **Flexbox:**
    - **Grid:**
    - **Floats:** cause block level elements to wrap alongside one side of an element
    - **Positioning:** The position property — Allows you to precisely control the placement of boxes inside other boxes.
    - **Table layout:** can be used on non-table elements using display: table and associated properties
    - **Multiple-column layout:** The Multi-column layout properties can cause the content of a block to layout in columns, as you might see in a newspaper
-

---

# CSS LAYOUT WITH FLEXBOX

- **Flexbox:** designed to make it easy for us to lay things out in one dimension — either as a row or as a column
    - **Flex container:**
      - Display: flex
      - Flex-direction:
      - Flex-wrap
    - **Flex item:** (short hand: flex: grow shrink basis, **eg:** flex: 0 1 100px;
      - Flex-basis: set initial size of the flex items
      - Flex-grow: how items will expand in case of an extra space in the container
      - Flex-shrink: how items will shrink if there's not enough room in the container
-

---

# CSS LAYOUT WITH GRID

- offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning
  - Grid container
    - Grid-template-column
    - Grid-template-row
    - Justify-content
    - Align content
  - Grid Items
    - Grid-column
    - Grid-row
-

---

# REFERENCES

- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://www.w3schools.com/css/default.asp>