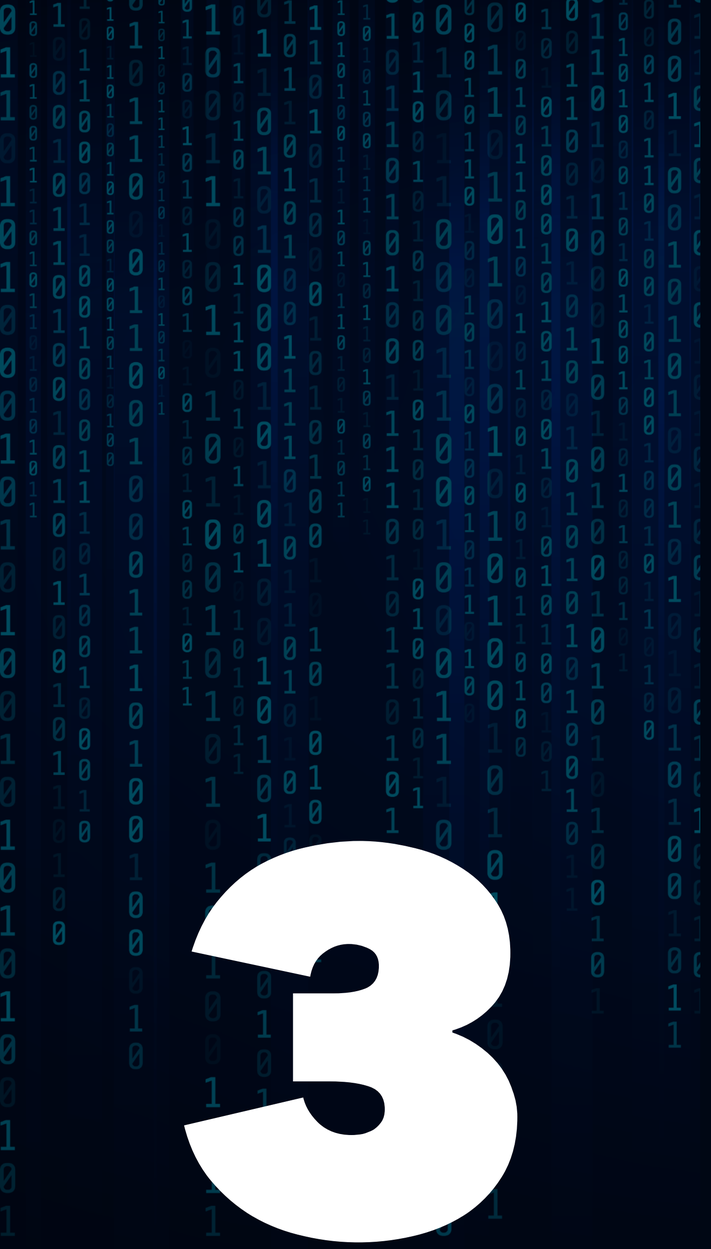
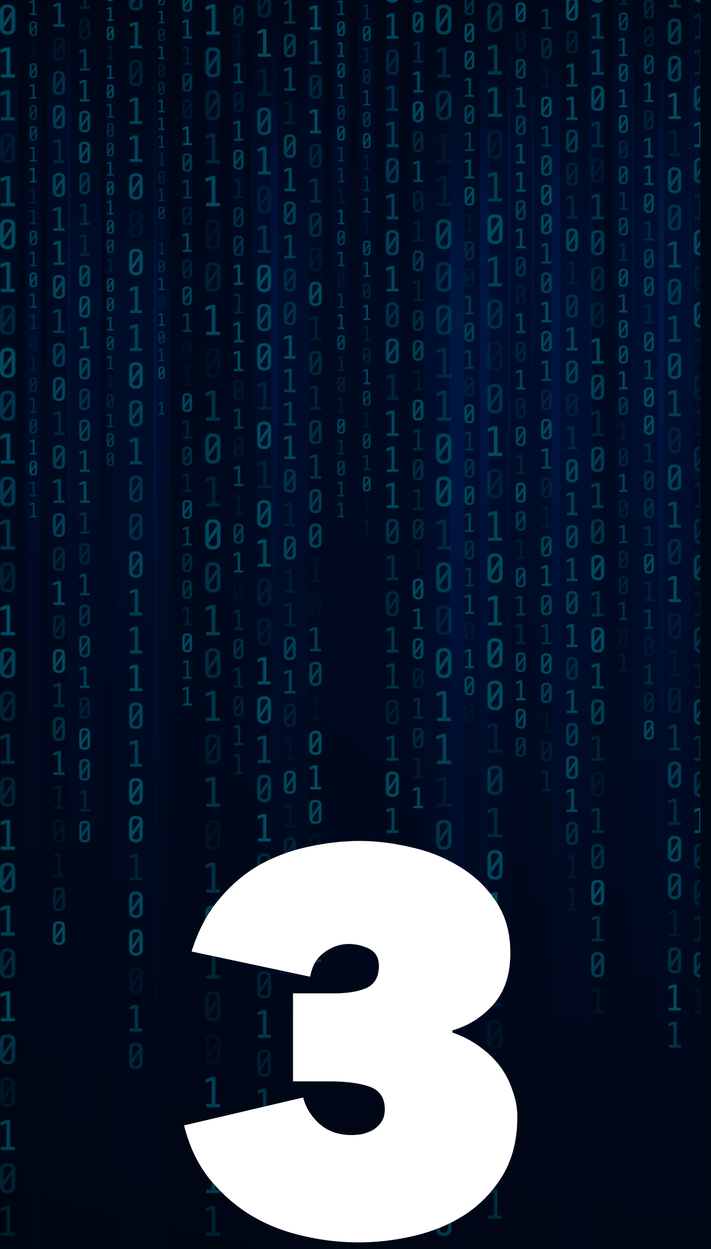


INM120

Introduction to Adaptive Web Design



3

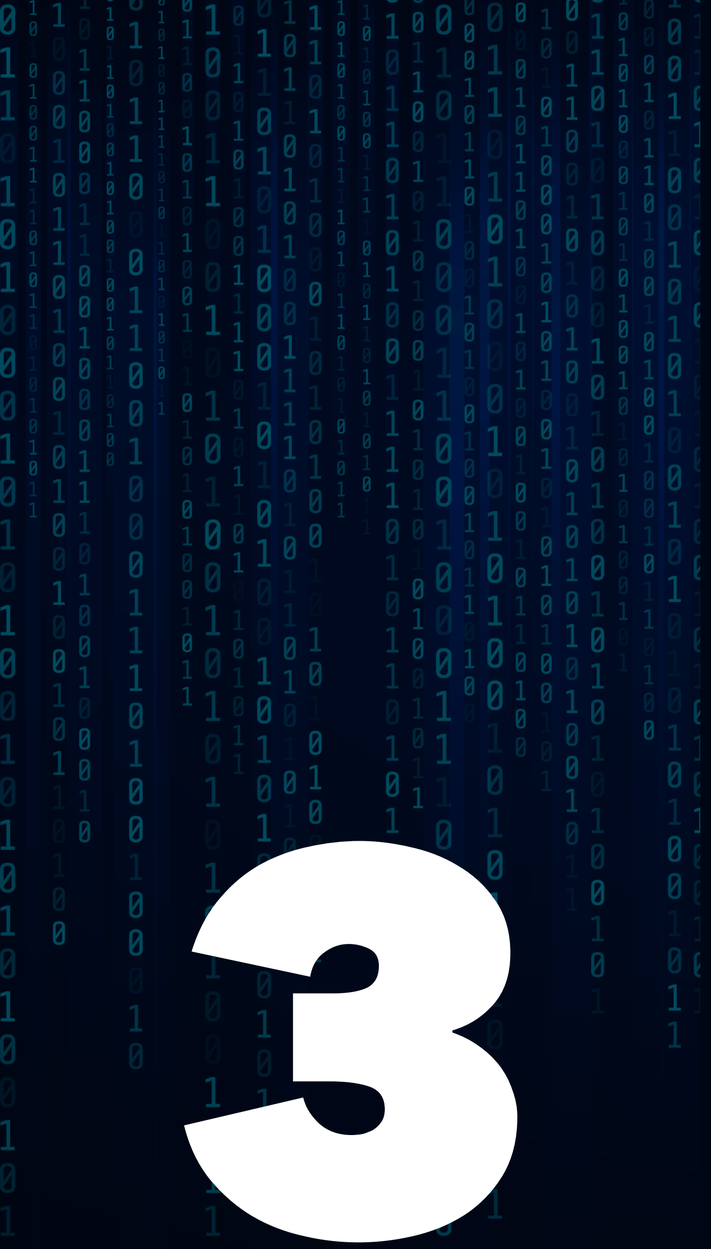
CASCADING STYLE SHEET

CSS stands for Cascading Style Sheets. It's a programming language used in web development to control the layout, formatting, and presentation of HTML documents.

CSS allows web developers to apply styles, such as colours, fonts, spacing, and positioning, to HTML elements. By separating content (HTML) from its presentation (CSS), web designers can create visually appealing and consistent websites across different devices and screen sizes.

There are three primary ways to add CSS (Cascading Style Sheets) to an HTML document:

- Inline CSS
- Internal CSS
- External CSS



CASCADING STYLE SHEET

Inline CSS:

You can apply CSS directly to individual HTML elements using the "style" attribute.

```
<p style="color: blue;">This is a blue and larger text.</p>
```

Inline CSS is useful for making quick, specific style changes to a single element.

Internal CSS:

You can embed CSS within the HTML document using the <style> element in the document's <head>.

```
<html>
  <head>
    <style>
      p {
        color: green;
      }
    </style>
  </head>
  <body>
    <p>This is green and larger text.</p>
  </body>
</html>
```

Internal CSS is useful when you want to apply styles to multiple elements on a single page.

3

CASCADING STYLE SHEET

External CSS:

You can create a separate CSS file (with a **.css extension**) and link it to your HTML document using the **<link>** element.

Using external CSS files is the most common and recommended approach for larger websites because it allows you to keep your styles separate from your HTML, making it easier to manage and maintain consistent styles across multiple web pages.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css"
    href="styles.css">
  </head>
  <body>
    <p class="highlight">This text has a special
    highlight.</p>
  </body>
</html>
```

rel Attribute: The rel attribute stands for "relationship," and it specifies the relationship between the current document and the linked resource.

type Attribute: The type attribute specifies the MIME type of the linked resource. MIME types describe the nature of the resource, whether it's HTML, CSS, JavaScript, images, or other data.

href Attribute: The href attribute specifies the URL of the linked resource. It tells the browser where to find the external file or resource.

3

CSS SYNTAX

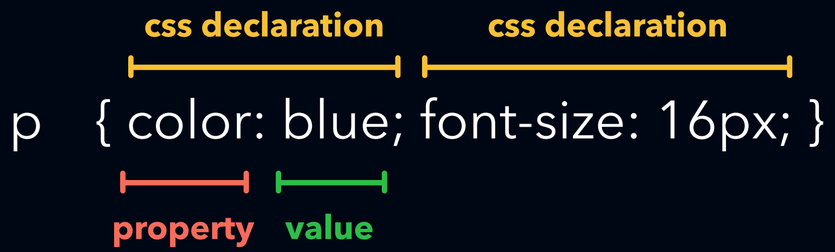
CSS syntax is a set of rules and conventions that define how CSS code should be written to apply styles to HTML elements effectively. Here are some key components of CSS syntax:

Selectors: Selectors are used to target HTML elements to which you want to apply styles. Selectors can be element **names** (e.g., p for paragraphs), **class names** (e.g., .highlight for elements with a "highlight" class), **IDs** (e.g., #header for an element with the "header" ID), or more **complex selectors**.

Declaration Blocks: After selecting an element, you define the styles you want to apply within a declaration block. A declaration block is enclosed in **curly braces {}**. Inside the block, you list one or more property-value pairs.

3

CSS SYNTAX



CSS declaration: A CSS declaration is a fundamental part of CSS syntax. It consists of two main components: a **CSS property** and its associated **value**. Declarations are used to define how specific HTML elements should be styled or formatted on a web page.

CSS Property: This is the attribute or characteristic of an HTML element that you want to style. Properties specify **what aspect** of the element you **want to change**, such as its **color**, **font size**, **margin**, **padding**, **border**, and more.

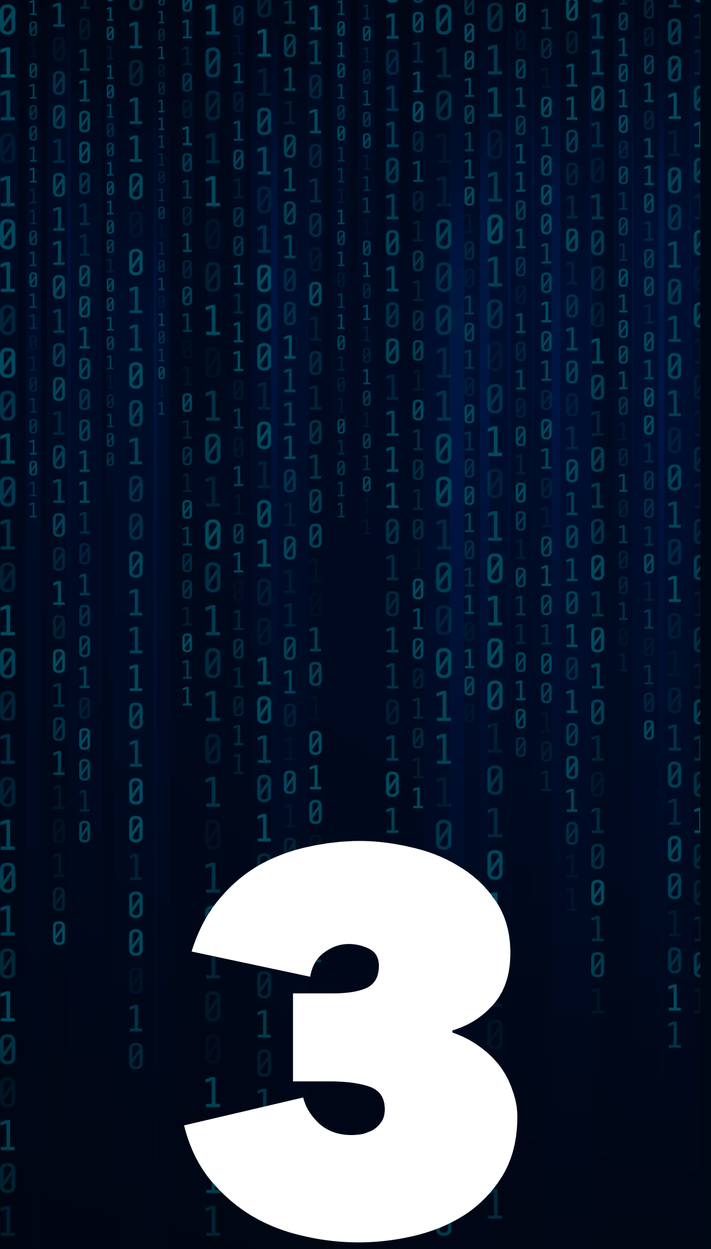
Property Value: This is the setting or value assigned to the CSS property. It determines how the property should affect the selected HTML element. Property values can be specific values like colors (e.g., **red**, **#00FF00**), sizes (e.g., **16px**, **2em**), positions (e.g., **center**, **top left**), or even references to external resources like image URLs (e.g., **url('image.jpg')**). The value depends on the property being used.

CSS SYNTAX

Semicolons: Each property-value pair within a declaration block is separated by a **semicolon (;)**. The semicolon tells the browser that one **property-value pair has ended**, and the next one is beginning.

NOTE: If you miss a semicolon at the end of a property-value pair, it can have several consequences:

- **Parsing Error:** It confuses the browser, causing it to misinterpret the CSS. This can lead to incorrect or inconsistent styles on your webpage.
- **Other Rules Affected:** It can affect not only the current rule but also subsequent rules in the same section. This might result in unexpected and undesirable webpage styles.
- **Unpredictable Results:** Sometimes, the browser tries to guess where the missing semicolon should go. This can lead to unpredictable webpage appearance, not matching what you intended.



CSS SYNTAX

CSS Comments: CSS comments are explanatory notes or remarks that you can add to your CSS code to provide information, explanations, or reminders for yourself or other developers working on the code. CSS comments are **not visible** on the web page and are ignored by web browsers when rendering the page. They are purely for human understanding and documentation purposes.

Purposes:

- CSS comments are helpful for documenting your code, **providing context**, **explaining** complex or non-obvious styling choices.
- You can also use CSS comments to "**comment out**" or **temporarily disabling** specific styles for **debugging** or **testing purposes**.

short key for comments command + ?

```
* This is a single-line CSS comment. */
/* This is a multiline CSS comment.
   It can span multiple lines for detailed explanations. */

/*  p {
    color: red; /* This style is temporarily disabled */
}  */
```

single & multiline comment

comment out code

3

CSS SPECIFICITY OR CASCADE

In CSS, there is **a priority hierarchy** for determining which styles take precedence when multiple conflicting styles are applied to the same HTML element. This hierarchy is commonly referred to as the "**CSS Specificity**" or "**Cascade**."

The order of precedence, from highest to lowest, is as follows:

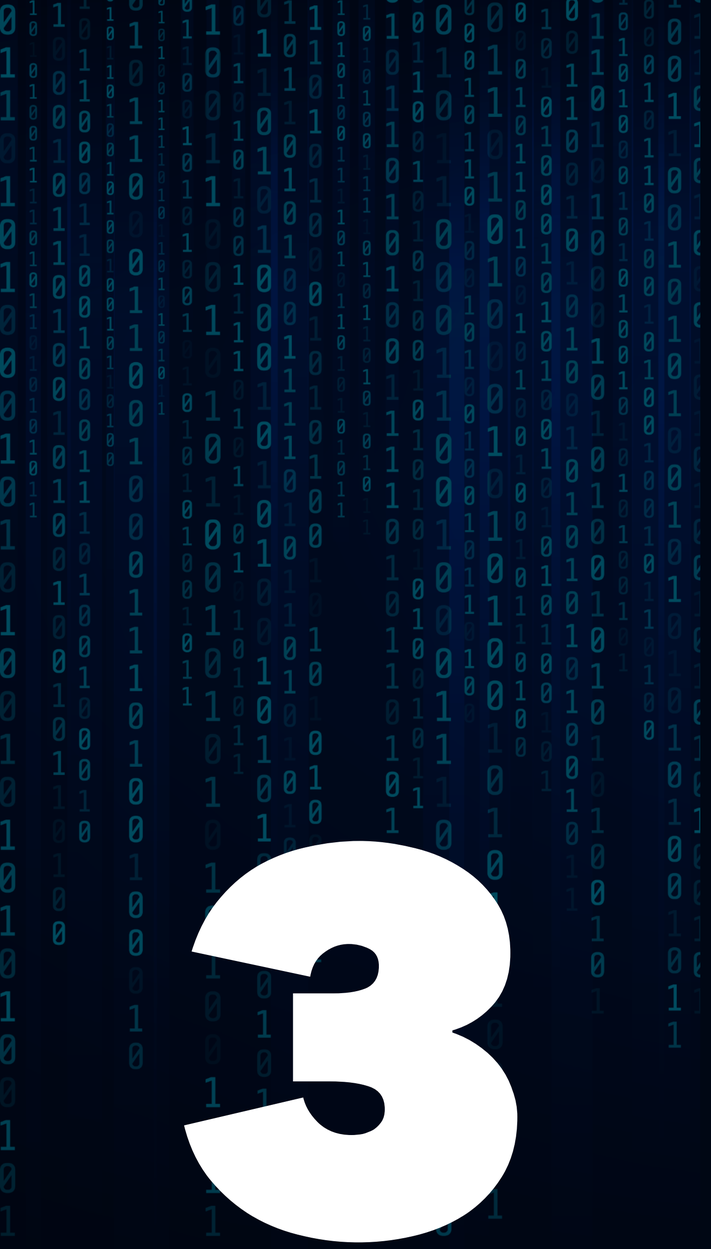
1. **Inline Styles**
2. **Internal (Embedded) Styles**
3. **External Stylesheets**

WHAT IS CSS RESET?

A CSS reset is **a set of CSS rules** or styles that aim to "**reset**" or standardize **the default browser styling** of HTML elements. The purpose of a CSS reset is to create a consistent baseline for styling across different web browsers and ensure that web developers have more control over the appearance of their web pages.

ADD THIS TO YOUR TEMPLATE:

```
<link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/reset-css@3.0.0/reset.min.css"/>
```



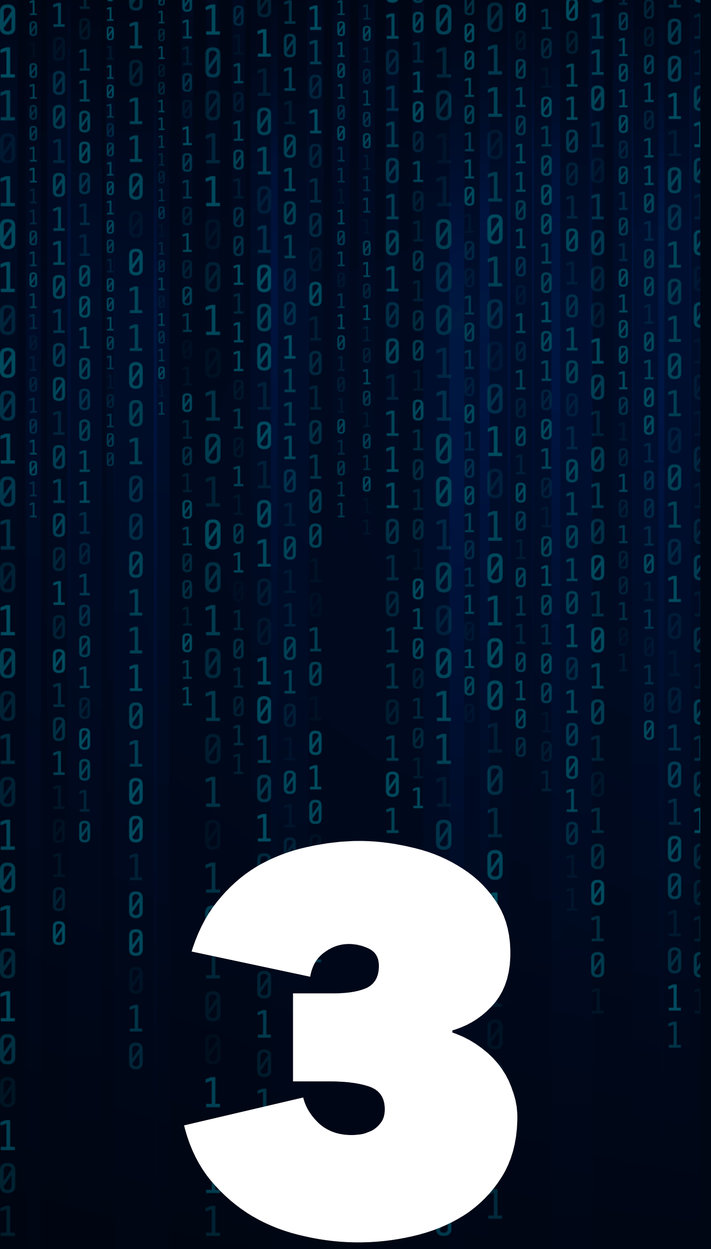
CSS SELECTORS

Selectors in CSS are patterns that define which HTML elements should be styled by a given set of CSS rules. There are several types of selectors in CSS, each with its own way of targeting elements:

- **Type Selector (Element Selector):**
  Target elements based on their HTML tag name.
  Example:
  p { color: blue; }, **p** targets all **<p>** elements.

- **Class Selector:**
  Target elements with a specific class attribute.
  Example:
  .highlight targets elements with **class="highlight"**.

- **ID Selector:**
  Target a single element with a specific ID attribute.
  Example:
  #header targets the element with **id="header"**.



CSS SELECTORS

- **Descendant Selector:**
  Target elements that are **descendants** of a specified element.
  Example:
  ul li targets all **<li>** elements **within a **.

  space ( )

- **Child Selector:**
  Target elements that are **direct children** of a specified parent element.
  Example:
  ul > li targets **only ** elements that are **direct children** of a **<ul>**.

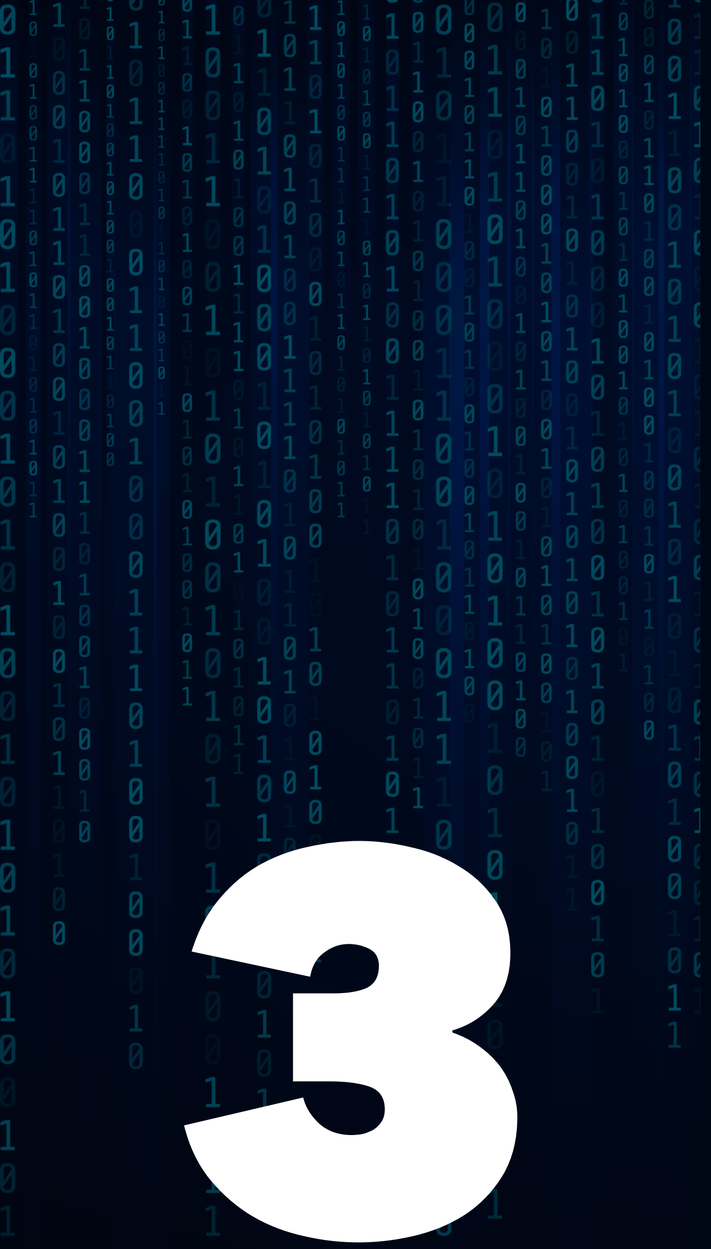
  close angle bracket (>)

- **Adjacent Sibling Selector:**
  Target an element that is **immediately preceded** by another element with the same parent. Example:
  h2 + p targets the **<p>** **immediately following an <h2>**.

  plus (+)

- **General Sibling Selector:**
  Target **elements that are siblings** of a specified element and share the same parent. Example:
  h2 ~ p targets all **<p>** elements that are **siblings of an <h2>**.

  tilda (~)



CSS SELECTORS

- **Attribute Selector:**
  Target elements with a specific attribute or attribute value.
  Example:
  input [type="text"] targets **input** elements **with type="text"**.

  square brakets [ ]

- **Pseudo-Class Selector:**
  Target elements based on their **state** or **position**, such as **:hover**, **:active**, **:first-child**, **:nth-child()**, and more. Example:
  a:hover targets **links when** they are **hovered over**.

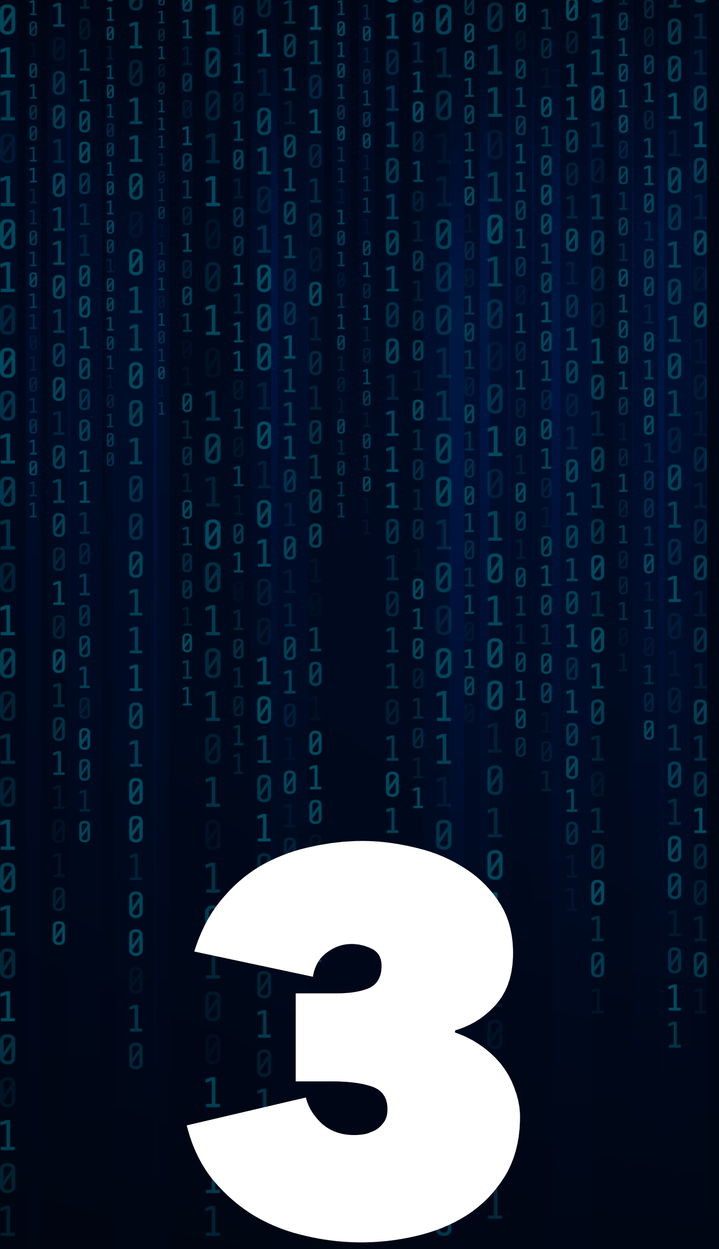
  colon (:)

- **Pseudo-Element Selector:**
  Target **parts of an element** or **generate content** that is not part of the HTML structure, such as **::before** and **::after**. Example:
  p::before inserts content before all **<p>** elements.

  double colons (::)

- **Universal Selector:**
  Target all elements on the page.
  Example:
  * targets **all elements**.

  tilda (*)



CSS SELECTORS PRIORITY

In CSS, selectors have a specific priority or specificity when determining which styles should be applied to an element if multiple conflicting styles are defined. The priority is determined by the specificity of the selector.

1. ID Selectors
2. Class Selectors and Attribute Selectors:
3. Type (Element) Selectors and Pseudo-Elements:
4. Universal Selectors:
5. Combinations and Specificity Calculations:

!important:

!important declaration is used to give a specific style rule **the highest priority** or specificity, overriding any other conflicting styles that may apply to an element. When a style rule is marked as !important, it tells the browser to **apply that rule's styles no matter what**, even if other rules have higher specificity or are defined later in the stylesheet.