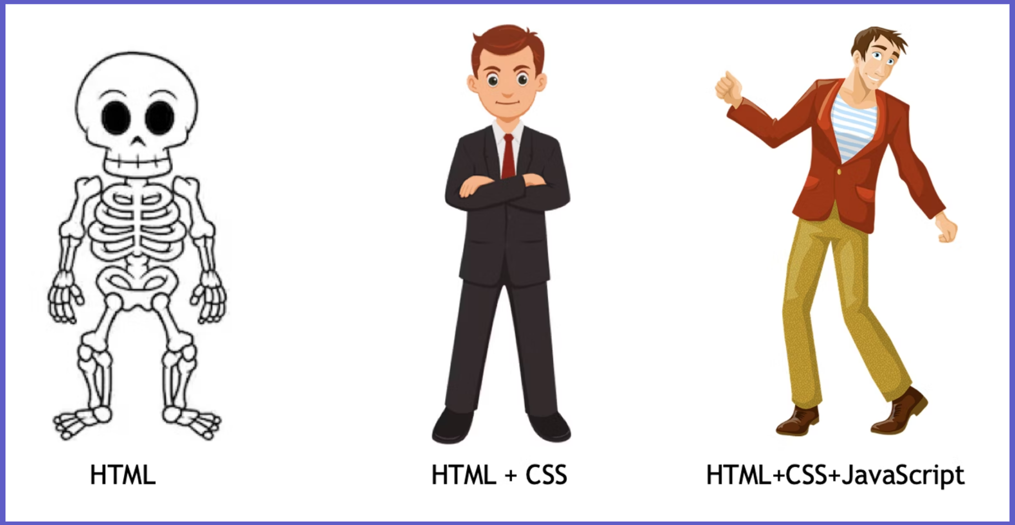
CSS Introduction

- CSS stands for Cascading Style Sheets.
- CSS is a standard style sheet language used for describing the presentation

(i.e. the layout and formatting) of the web pages.



What You Can Do with CSS?

- You can easily apply same style rules on multiple elements.
- You can control the presentation of multiple pages of a website with a single style sheet.
- You can present the same page differently on different devices.
- You can style dynamic states of elements such as hover, focus, etc. that isn't possible otherwise.
- You can change the position of an element on a web page without changing the markup.
- You can alter the display of existing HTML elements.
- You can transform elements like scale, rotate, skew, etc. in 2D or 3D space.
- You can create animations and transitions effects without using any JavaScript.
- You can create print friendly version of your web pages.

Including CSS in HTML Documents

CSS can either be attached as a separate document or embedded in the HTML document itself. There are three methods of including CSS in an HTML document:

- Inline styles — Using the style attribute in the HTML start tag.
- Embedded styles — Using the <style> element in the head section of a document.
- External style sheets — Using the <link> element, pointing to an external CSS file:

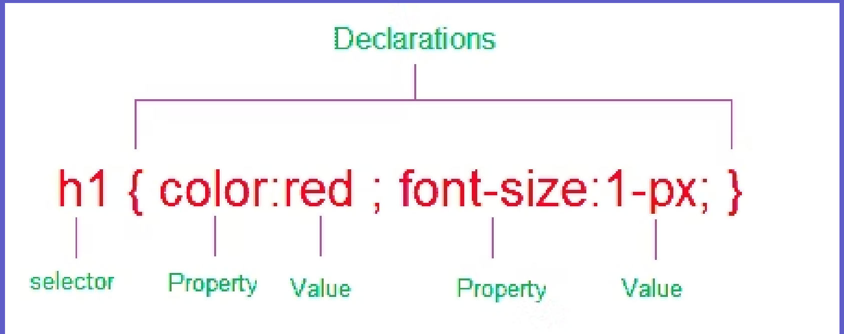
  ```
  <link rel="stylesheet" href="pagestyle.css">
  ```

Understanding CSS Syntax

- A CSS stylesheet consists of a set of rules that are interpreted by the web browser and then applied to the corresponding elements such as paragraphs, headings, etc. in the document.
- A CSS rule have two main parts

  - a selector

  - one or more declarations:



```
h1 { color:red ; font-size:1-px; }
```

What is Selector?

- A CSS selector is a pattern to match the elements on a web page.
    - The style rules associated with that selector will be applied to the elements that match the selector pattern.
- Selectors are one of the most important aspects of CSS as they allow you to target specific elements on your web page in various ways so that they can be styled.

So basically a CSS selector selects the HTML element(s) you want to style.

We can divide CSS selectors into 5 categories:

- **Simple selectors (select elements based on name, id, class)**
- **Combinator selectors(select elements based on a specific relationship between them)**
- **Pseudo-class selectors(select elements based on a certain state)**
- **Pseudo-elements selectors (select and style a part of an element)**
- **Attribute selectors (select elements based on an attribute or attribute value)**

Universal Selector

The universal selector, denoted by an asterisk (*), matches every single element on the page.

```css
* {
        margin: 0;
        padding: 0;
}
```

Element Selector

The element selector selects HTML elements based on the element name.

```
h1 {
        color: blue;
}
```

ID Selector

- Uses the id attribute to select an html element
- Unique within a page
- Selects one unique element with a hash (#) character

```
#uniqueIDAttributedElement {
        color: blue;
}
```

Class Selector

- Uses the class attribute to select html element(s)
- Selects one unique element with a period (.) character

```
.error {
 color: red;
}
p.error {
  color: darkred;
}
```

HTML elements can also refer to more than one class.

Grouping Selector (with a comma)

```
h1, h2, p {
 color: gray;
}
```

Combinator selectors

- descendant selector (space)
- child selector (>)
- adjacent(immediately following) sibling selector (+)
- general sibling selector (~)

Go through all kind of combinator selectors according to following code snippet:

```
  <p>Paragraph outside.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section>
<!– Descendant –>
   <p>Paragraph 3 in the div.</p>
  </section>
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
```

Pseudo-classes

A pseudo-class is used to define a special state of an element.

The syntax of pseudo-classes:

```
selector:pseudo-class {
  property: value;
}
```

- Link-Related pseudo-class selectors
- Input & link related pseudo-class selectors
- Position/Number-based pseudo-class selectors
- Relational pseudo class selectors
- Text-related pseudo class selectors

Link-Related pseudo-class selectors

:link - unvisited link
:visited - visited link

:hover - mouse over link

:active - selected link

Note: *`a:hover` MUST come after `a:link` and `a:visited` in the CSS definition in order to be effective! `a:active` MUST come after `a:hover` in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.*

Input & link related pseudo-class selectors

:focus

:target - The target pseudo class is used in conjunction with IDs, and match when the hash tag in the current URL matches that ID

:checked - Selects checkboxes that are, wait for it, checked.

:disabled - Selects inputs that have the disabled attribute. A lot of browsers will make the input a faded out gray, you can control that with this selector.

:enabled – Selects inputs that are in the default state of enabled and ready to be used.

:optional – Selects inputs that do not have the `required` attribute.

:required – Selects inputs with the `required` attribute.

:in-range

:out-of-range

:invalid

:valid

:read-only

:read-write

Position/Number-based pseudo-class selectors

:root - Selects the element that is at the root of the document(html element)

:first-child – Selects the first element within a parent.

:first-of-type -Selects the first element of this type within any parent.

:last-child - Selects the last element within a parent.

:last-of-type - Selects the last element of this type within any parent.

:nth-child(n)

:nth-last-child(n)

:nth-last-of-type(n)

:nth-of-type(n)

:only-of-type - Selects only if the element is the only one of its kind within the current parent.

:only-child

The Difference Between :nth-child and :nth-of-type:

Our `:nth-child` selector above, in "Plain English," means select an element *if*:

1. It is a paragraph element
2. It is the second child of a parent

Our `:nth-of-type` selector, in "Plain English," means:

1. Select the second paragraph child of a parent

Relational pseudo class selectors

`:not()` – Removes elements from an existing matched set that match the selector inside the parameter of :not(). So for example, all divs *except* those with a class of "music" = div:not(.music). The spec says that :not selectors cannot be nested, but they can be chained. Some browsers (Firefox) also support comma-separated selectors as the selector parameter, although chaining them would be a far safter bet. Also useful in conjunction with attribute selectors, e.g. input:not([disabled]).

`:empty` – Selects elements which contain no text and no child elements. Like:

Text-related pseudo class selectors

`:lang` – This pseudo selector is in the CSS3 spec but is only implemented in IE 8+. Will match anything that either has or is a descendant of an element with a matching lang attribute. For example, `:lang(fr)` will match any paragraph, even without a lang attribute, if the parent body had `lang="fr"` as an attribute.

Pseudo-elements

Pseudo-element is used to style specified parts of an element.

The syntax of pseudo-classes:

```
selector::pseudo-element {
  property: value;
}
```

::after

::before

::first-letter

::first-line

::marker

::selection

Attribute Selectors

The `[attribute]` selector is used to select elements with a specified attribute.

[_attribute_] - `a[target] { background-color: red; }`

[attribute="value"] - exact checking

[_attribute_~="value"] - Selects all elements with the attribute that contains a space-separated list of words, one of which is “_value_”

[_attribute_|="value"] - Selects all elements with the attribute value starting with the "value"

[_attribute_^="value"] - Selects all elements with the attribute value starting with the "value"

[_attribute_$="value"] - Selects all elements with the attribute value ending with ""

[_attribute_*="value"] - Selects all elements with the attribute value containing the substring "value"

Color

The color property defines the text color of an element.

For instance, the color property specified in the body selector defines the default text color for the whole page.

Let's try out the following example to see how it works:

```
body {
    color: #ff5722;
}
```

Colors in CSS most often specified in the following formats:

a color name - like "red"

a HEX value - like "#ff0000"

an RGB value - like "rgb(255, 0, 0)"

CSS3 has introduced several other color formats such as HSL, HSLA and RGBA that also support alpha transparency. We'll learn about them in greater detail in CSS3 color chapter.

Styling Fonts

- Choosing the right font and style is very crucial for the readability of text on a page.
- CSS provide several properties for styling the font of the text, including changing their face, controlling their size and boldness, managing variant, and so on.
- The font properties are:
- `font-family`
- `font-style`
- `font-weight`
- `font-size`
- `font-variant`

Font Family

`font-family`: definies the font that is applied to the element

This property can hold several comma-separated font names as a fallback system, so that if the first font is not available on the user's system, browser tries to use the second one, and so on.

A typical font family declaration might look like this:

```
body {
        font-family: Arial, Helvetica, sans-serif;
}
```

Font Style

`font-style`: makes the text appear italicised or oblique.

- `normal`
- `italic`
- `oblique`

The default value is normal.

```
p.italic {
        font-style: italic;
}
```

Font Variant

- `font-variant`: changes target text to small caps.
  - `normal`
  - `small-caps`
  - `inherit`

Font Weight

- `font-weight`: sets the weight or the thickness of the font.
  - `normal`
  - `bold`
  - `bolder`
  - `lighter`
  - `100, 200, 300, 400, 500, 600, 700, 800, 900`
  - `inherit`

Font Size

- `font-size`: sets the height of the font.
  - `xx-small`
  - `x-small`
  - `small`
  - `medium`
  - `large`
  - `x-large`
  - `xx-large`
  - `smaller, larger`
  - `percentage`
  - `inherit`

Formatting Text

- CSS provides several properties that allows you to define various text styles such as color, alignment, spacing, decoration, transformation, etc. very easily and effectively.The commonly used text properties are: `text-align, text-decoration, text-transform, text-indent, line-height, letter-spacing, word-spacing`, and more. These properties give you precise control over the visual appearance of the characters, words, spaces, and so on

Text Alignment

- The text-align property is for setting the horizontal alignment of the text.
- Text can be aligned in four ways: to the left, right, centre or justified (straight left and right margins).

```
h1 {
        text-align: center;
}
```

Text Decoration

The **text-decoration** property is used to set or remove decorations from text.

This property typically accepts one of the following values:

- underline
- overline
- line-through
- none.

You should avoid underline text that is not a link, as it might confuse the visitor.

```
h2 {
        text-decoration: line-through;
}
```

Text Transformation

The text-transform property is for setting the cases for a text.

Text are often written in mixed case. However, in certain situations you may want to display your text in entirely different case. Using this property you can change an element's text content into

- `uppercase`
- `lowercase`
- `capitalize the first letter of each word`

without modifying the original text.

```
        h3 {
        text-transform: lowercase;
        }
```

Letter Spacing

The letter-spacing property is used to set extra spacing between the characters of text.

This property can take a length value in pixels, ems, etc.

It may also accept negative values.

When setting letter spacing, a length value indicates spacing in addition to the default inter-character space.

```
   p {
        letter-spacing: 10px;
     }
```

Word Spacing

The word-spacing property is for specifying additional spacing between the words.

This property can accept a length value in pixels, ems, etc. Negative values are also allowed.

```
p.normal {
        word-spacing: 20px;
}
```

Line Height

The line-height property is used to set the height of the text line.

It is also called leading and commonly used to set the distance between lines of text.

The value of this property can be a number, a percentage (%), or a length in pixels, ems, etc.

```
p {
        line-height: 1.2;
}
```

Background

Background plays an important role in the visual presentation of a web page.

CSS provide several properties for styling the background of an element, including coloring the background, placing images in the background and managing their positioning, etc.

The background properties are :

- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`
- `background-color`

Background Color

The background-color property in CSS applies solid colors as background on an element. Here's an example:

```
body {
        background-color: #f0e68c;
}
```

Background Image

The `background-image` property in CSS applies a graphic (e.g. PNG, SVG, JPG, GIF, WEBP) or gradient to the background of an element.

```
body {  background-image: url(sweettexture.jpg);}
```

Example for gradient:

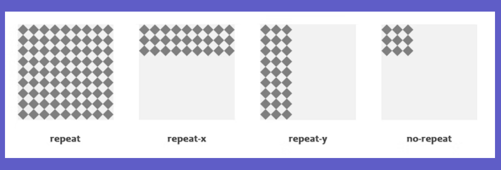
```
body {
 background-image: linear-gradient(#e66465, #9198e5);
}
```

Background Repeat

If a `background-image` property is specified, the `background-repeat` property in CSS defines if (and how) it will repeat. Here's an example:

```
body {
   background-image: url(imageUrl.jpg);
   background-repeat: repeat-x;
}
```

These are the possible values for this property (besides the usual stuff like `inherit`):



- `repeat`: tile the image in both directions. This is the default value.
- `repeat-x`: tile the image horizontally
- `repeat-y`: tile the image vertically
- `no-repeat`: don't tile, just show the image once
- `space`: tile the image in both directions. Never crop the image unless a single image is too large to fit. If multiple images can fit, space them out evenly images always touching the edges.
- `round`: tile the image in both directions. Never crop the image unless a single image is too large to fit. If multiple images can fit with leftover space, squish them or stretch them to fill the space. If it's less than half one image width left, stretch, if it's more, stretch.

Background Position

The `background-position` property in CSS allows you to move a background image (or gradient) around within its container.

```
html {  background-position: 100px 5px; }
```

It has three different types of values:

- Length values (e.g. `100px 5px`)
- Percentages (e.g. `100% 5%`)
- Keywords (e.g. `top right`)

Background Position

The background-size CSS property specifies the size of the background images.

The following table summarizes the usages context and the version history of this property.

```
background-size: length | percentage | auto | cover | contain
```

Shorthand

As you can see in the examples above, there are many properties to consider when dealing with the backgrounds. However, it is also possible to specify all these properties in one single property to shorten the code or avoid extra typing. This is called a shorthand property.

```
body {
        background: #f0e68c url("imgUrl.png") no-repeat fixed
250px 25px;
}
```