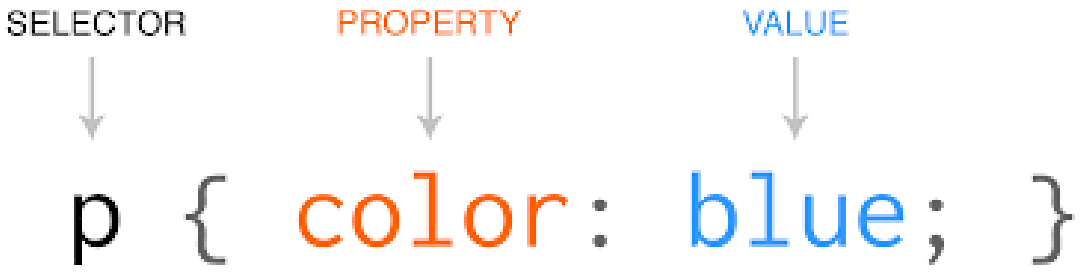
# An introduction to Cascading Style Sheets

- CSS is the acronym for: ‘Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen or in other media
- CSS is an extension to basic HTML that allows you to style your web pages.
- CSS work on  only **property: value**

# 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 set contains a selector and a declaration block.




- The selector specifies which element or elements in the HTML page the CSS rule applies to.
- Whereas, the declarations within the block determines how the elements are formatted on a webpage. Each declaration consists of a property and a value separated by a colon (:) and ending with a semicolon (;), and the declaration groups are surrounded by curly braces {}.

# Applying CSS

There are three ways to apply CSS to HTML

- In-line CSS
- Internal CSS
- External CSS

# In-line CSS

- In-line styles are plonked straight into the HTML tags using the style attribute.
- **Example**

```
<p style="color : red">This is my CSS Tag</p>
```

This will make that specific paragraph red

# Internal CSS

- Embedded, or internal styles are used for the whole page. Inside the head tags, the style tags surround all of the styles for the page.
- <style type="text/css">

```
    p {   color: red;     }
    a {   color: blue;    }
</style>
```

This will make all of the paragraphs in the page red and all of the links blue.

# External CSS

- External styles are used for the whole, multiple-page website. There is a separate CSS file.

- Make a CSS file Web.css.

```
p {  color: red; }
```

- Link the css file in your html document.
- <link href="web.css" rel="stylesheet" type="text/css" />

# CSS Selectors, Properties, and Values

- HTML has tags, CSS has 'selectors'. Selectors are the names given to styles in internal and external style sheets.
- Each selector there are 'properties' inside curly brackets, which simply take the form of words such as color, font-weight or background-color.
- A value is given to the property following a colon (NOT an 'equals' sign) and semi-colons separate the properties.

# CSS Selectors

- **The CSS element Selector :** The element selector selects HTML elements based on the element name.
- **Example :**

```
p {
    text-align: center;
    color: red;
}
```

- **The CSS id Selector :** 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.

# CSS Selectors

- **The CSS id Selector :**
- **Example :**

```
#para1 {
        text-align: center;
  color: red;
 }
```

- **The CSS class Selector :** The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
- **Example :**

```
.center {
    text-align: center;
color: red;
    }
```

**Note :** HTML elements can also refer to more than one class.

# CSS Selectors

- **The CSS Universal Selector :** The universal selector (*) selects all HTML elements on the page.
- **Example :**

```
* {
    text-align: center;
    color: blue;
}
```

- **The CSS child Selector :** This rule will render all the paragraphs in red if they are direct child of <body> element. Other paragraphs put inside other elements like  or <td> would not have any effect of this rule.
- **Example :**

```
body > p {
    color: red;
}
```

# CSS Selectors

- **The CSS  Descendant Selector :** It is used to select those elements which are inside of a given parent.
- **Example :**

```
div p {
color: blue;
 }
```

It will select each and every <p> element which is inside of the . It means that these paragraphs either can be the direct child of the div or can be the nested child of the div element.

# CSS Selectors

- **Adjacent and general sibling selectors:** Adjacent means "immediately following". This selector is used when you want to select the elements that immediately follow the specified element (adjacent siblings). In other words it selects the element which is right next to another element at the same level of the hierarchy.
- **Example :**

```
div + p {
color: blue;
 }
```

- It selects p element that are directly following a 'div' element

# CSS Selectors

- **The CSS Attribute Selectors :** You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text
- **Example :**

```
input[type='text'] {
                    color: blue;
                        }
```

- **Grouping Selectors:** You can apply a style to many selectors if you like. Just separate the selectors with a comma.
- **Example :**

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

# CSS Selectors

- **Pseudo-classes and Pseudo element :** It styles an element based on the state of a specified element, CSS uses pseudo classes (:)
- **Example :** button:hover

```
{
    color: green;
}
```

# CSS Selectors

- **The :first-child Pseudo-class :** The :first-child pseudo-class matches a specified element that is the first child of another element.
- **Example :** `p:first-child`

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

```
p i:first-child

{
  color: blue;
}
```

- The selector matches the first <i> element in all <p> elements:

# CSS Selectors

- **The :last-child Pseudo-class :** The :first-child pseudo-class matches a specified element that is the first child of another element.
- **Example :** `p:last-child`

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

- Selects every <p> elements that is the last child of its parent

# CSS Selectors

- **The : nth-child Pseudo-class :** The :nth-child(n) selector matches every element that is the nth child of its parent.
- **n** can be a number, a keyword (odd or even), or a formula (like an + b).

**Example :**

```
/* Selects the second element of div siblings */
div:nth-child(2) {
      background: red;
}

/* Selects the second li element in a list */
li:nth-child(2) {
      background: lightgreen;
}

/* Selects every third element among any group of siblings */
:nth-child(3) {
      background: yellow;

}
```

# CSS Backgrounds

| Property | Description | Values |
|---|---|---|
| background-color | specifies the background color of an element. | RGB, hex, keyword |
| background-image | Sets the background image for an element | url |
| background-position | Sets the starting position of a background image | left, right, center |
| background-repeat | Sets how a background image will be repeated | repeat, repeat-x, repeat-y |
| background-attachment | Sets whether a background image is fixed or scrolls with the rest of the page | Scroll, fixed |
| | | |

# Text Properties

| Property | Description | Values |
|---|---|---|
| **color** | Sets the color of a text | RGB, hex, keyword |
| **line-height** | Sets the distance between lines | normal, *number*, *length*, *%* |
| **letter-spacing** | Increase or decrease the space between characters | normal, *length* |
| **text-align** | Aligns the text in an element | left, right, center, justify |
| **text-decoration** | Adds decoration to text | none, underline, overline, line-through |
| **text-indent** | Indents the first line of text in an element | *length*, *%* |
| **text-transform** | Controls the letters in an element | none, capitalize, uppercase, lowercase |
| **text-shadow** | The text-shadow property adds shadow to text. | Number, color name |

# CSS Fonts

| Property | Description | Values |
|---|---|---|
| **font-family** | this property is to specify the font of a text. | Font Names |
| **font Style** | This property is mostly used to specify italic text. | normal , italic |
| **font-weight** | This property specifies the weight of a font | Normal, lighter, bold |
| **font Variant** | property specifies whether or not a text should be displayed in a small-caps font. | Normal, small-caps |
| **font-size** | This property sets the size of the text. | int |
| | | |

# Border Properties

| Property | Description | Values |
|---|---|---|
| **border** | Sets all the border properties in one declaration | *border-width*, *border-style*, *border-color* |
| **border-bottom** | Sets all the bottom border properties in one declaration | *border-bottom-width*, *border-bottom-style*, *border-bottom-color* |
| **border-bottom-color** | Sets the color of the bottom border | *border-color* |
| **border-bottom-style** | Sets the style of the bottom border | *border-style* |
| **border-bottom-width** | Sets the width of the bottom border | *border-width* |
| **border-color** | Sets the color of the four borders | *color_name*, *hex_number*, *rgb_number*, transparent, inherit |
| **border-left** | Sets all the left border properties in one declaration | *border-left-width*, *border-left-style*, *border-left-color* |
| **border-left-color** | Sets the color of the left border | *border-color* |
| **border-left-style** | Sets the style of the left border | *border-style* |
| **border-left-width** | Sets the width of the left | *border-width* |

# CSS Lists

| Property | Description | Values |
|---|---|---|
| list-style-type | property specifies the type of list item marker | Circle, square, upper-roman, lower-alpha,none |
| list-style-image | property specifies an image as the list item marker | url(‘ ‘ ) |
| list-style-position | list-style-position property specifies the position of the list-item markers (bullet points). | Outside,inside |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# CSS Links

| Property | Description | |
|---|---|---|
| a:link | a normal, unvisited link | |
| a:visited | a link the user has visited | |
| a:hover | a link when the user mouses over it | |
| a:active | a link the moment it is clicked | |

**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.

# CSS Table

| Property | Description | Values |
|---|---|---|
| border | specify table borders in CSS | |
| width | Specifies width of table | |
| border-collapse | property sets whether the table borders should be collapsed into a single border | collapse |
| nth-child() | For zebra-striped tables | Even, odd |
| border-spacing | Specifies the distance between the borders of adjacent cells | |
| | | |
| | | |
| | | |
| | | |
| | | |

The display property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

**Block-level Elements :** A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

**Example :**  , <h1> - <h6> , <p>

**Inline Elements :** An inline element does not start on a new line and only takes up as much width as necessary.

**Example :** <a> , <img>

**display** property specifies the display behavior (the type of rendering box) of an element.

**Syntax** :

display: value;

**Property Values :**

| Value | Description |
|---|---|
| inline | Displays an element as an inline element (like ). Any height and width properties will have no effect |
| block | Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width |
| contents | Makes the container disappear, making the child elements children of the element the next level up in the DOM |
| grid | Displays an element as a block-level grid container |
| inline-block | Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values |
| list-item | Let the element behave like a <li> element |
| none | The element is completely removed |

- CSS transforms allow you to move, rotate, scale, and skew elements.
- CSS 2D Transforms Methods
- With the CSS transform property you can use the following 2D transformation methods:

| | |
|---|---|
| **translate()** | **rotate()** |
| **scaleX()** | **scaleY()** |
| **scale()** | **skewX()** |
| **skewY()** | **skew()** |

**translate() :** The translate() method moves an element from its current position.

Below example moves the  element 50 pixels to the right, and 100 pixels down from its current position:

```
div {
  transform: translate(50px, 100px);
}
```

**rotate() Method :** The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

The below example rotates the  element clockwise with 20 degrees.

```
div {
  transform: rotate(20deg);
}
```

Using negative values will rotate the element counter-clockwise.

- **scale() Method:** The scale() method increases or decreases the size of an element.
- Below example increases the  element to be two times of its original width, and three times of its original height:

```
div {
  transform: scale(2, 3);
}
```

- **skew() Method :** The skew() method skews an element along the X and Y-axis by the given angles.

```
div {
  transform: skew(20deg, 10deg);
}
```

# Thank you