

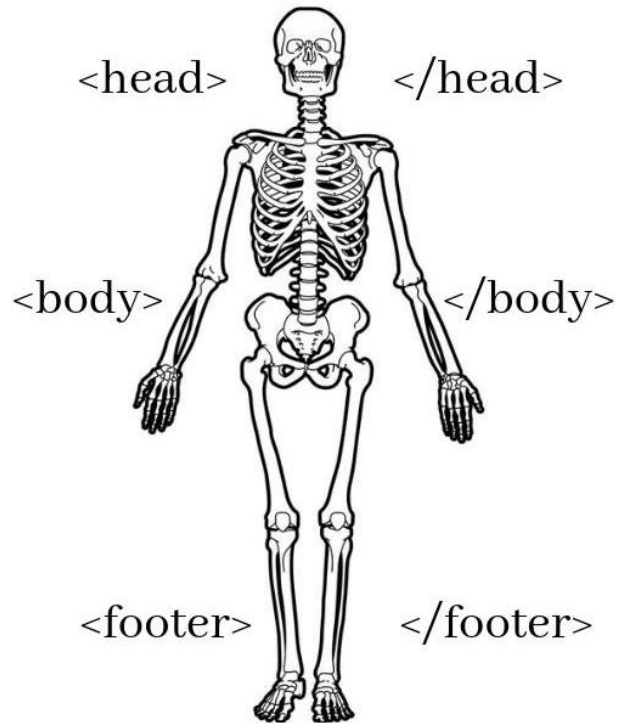
# **WEB TECHNOLOGY**

## **LECTURE – 03**

**CSS (CASCADING STYLE SHEETS)**

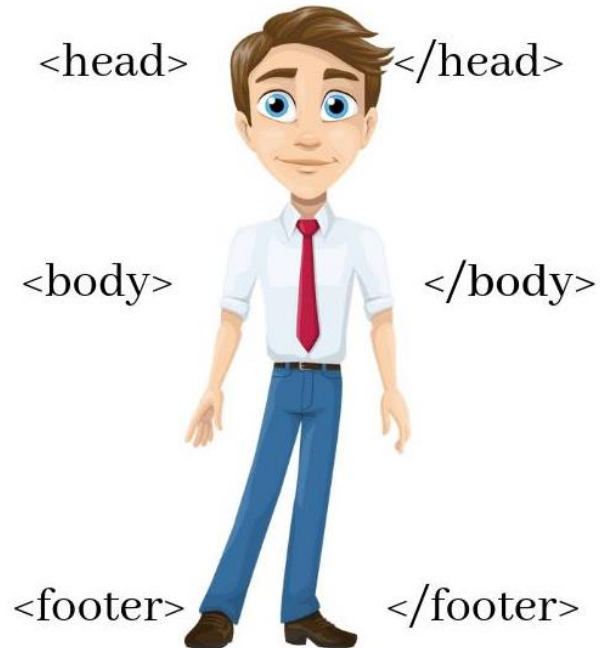
# HTML Vs CSS

## HTML



Structural Layer

## HTML + CSS



Presentational Layer



# HTML AND CSS

- **HTML** was **never meant** to be a presentation language. Some proprietary **vendors** have created **tags** to add **presentation** to structure. Example: `<font>`, `<b>`, `<i>`, `<u>`
- However, HTML presentation **tags are not sufficient** to all the desired designs and CSS has several other **advantages**.

`<p style="color: blue; font-size: 12px;">Hello World</p>`

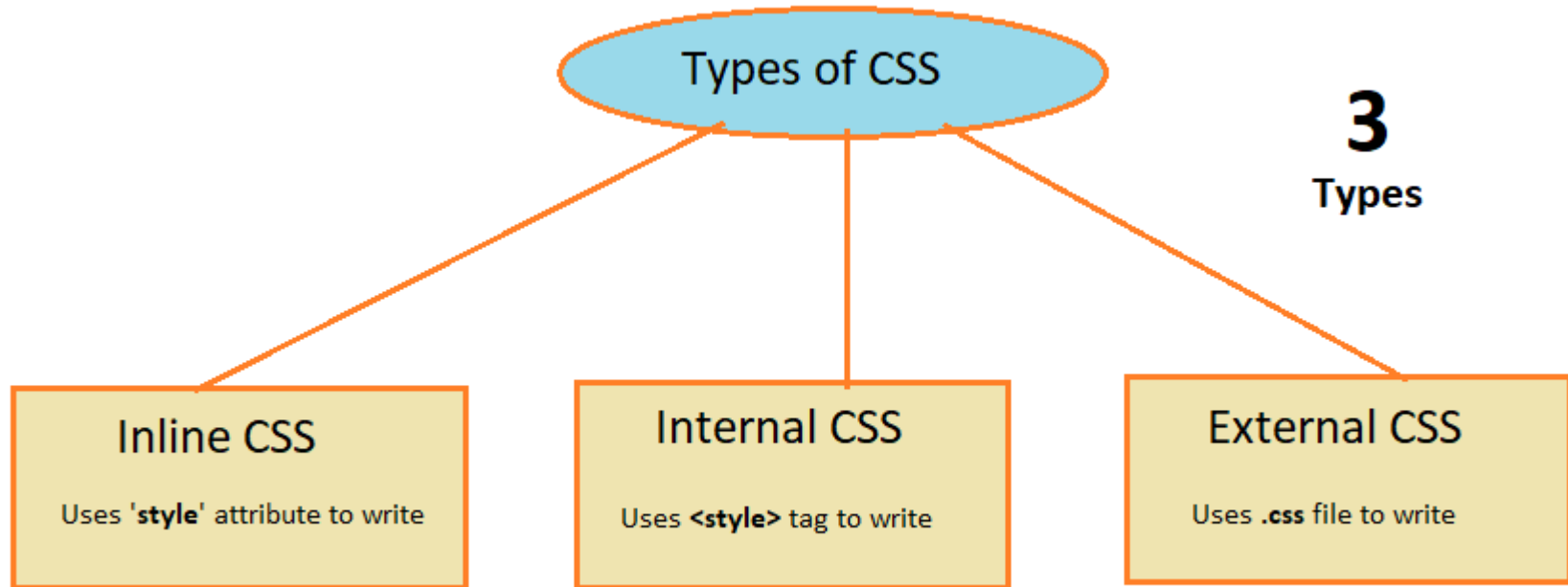
Style  
Attribute

Value as CSS

P Tag

- This is known as inline CSS

# CSS TYPES



External CSS supports writing embedded (or imported) CSS where one css file import another css file.



# INTERNAL CSS

```
<head>
  <style>
    h1 {
      color: blue; font-
size: 12px;
    }
  </style>
</head>
<body >
  <h1> This is internal
  CSS </h1>
</body>
```

# EXTERNAL CSS

```
<head>
<link rel="stylesheet"
  type="text/css"
  href="style.css" />
</head>
<body >
  <h1> This is
  external CSS </h1>
</body>
```

style.css

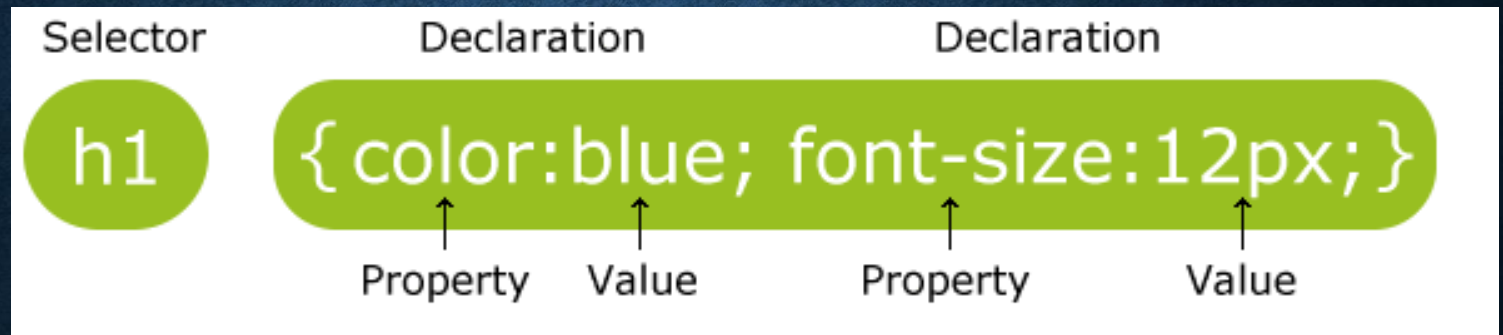
```
h1 {
  color: blue;
font-size: 12px;
}
```

# INTERNAL AND EXTERNAL CSS SYNTAX

Internal and External CSS require CSS selectors.

Syntax: selector {  
    property: value;  
}

Example:





# TAG SELECTOR

- Specify the style for a single HTML element.

```
P {  
  color: blue;  
  border-top: 1px solid #ff0;  
}
```

- “Comma” is used to group and specify style multiple elements.

```
h1, h2, h3, h4, h5, h6 {  
  font-family: “Trebuchet MS”, sans-serif;  
}
```

# CLASS AND ID SELECTOR

Class is used to group elements without using comma. It starts with "."

- Example

```
<p class="intro">This is my introduction with p</p>  
<div class="intro">This is my another intro with div</div>
```

```
.intro {  
    font: 12px verdana, sans-serif;  
    margin: 10px;  
}
```

ID is used to apply style for a single, unique element. It starts with "#"

- Example

```
<p id="pID"> This is my introduction text</p>
```

```
#pID {  
    border-bottom: 2px dashed #fff;  
}
```



# ADVANCED CSS SELECTORS

- Descendant Selector(space)

`div h1 { ... }`

`#navigation p { ... }`

- Child Selectors

`div ol > p { ... }`

- Universal Selector

`* { ... }`

- Attribute Selectors

`a[href="http://home.org"]`

- Pseudo-Class Selectors

`a:active { ... }`

`#nav:hover { ... }`

- Tag.Class: tag having that class

`P.first {....}`

# SPECIFICITY IN CSS

When multiple styles target the same element, **CSS Specificity** determines which rule applies.

Selector Type	Specificity Value
Inline styles ( <code>style=""</code> )	1000
ID Selector ( <code>#id</code> )	100
Class, Pseudo-class ( <code>.class</code> , <code>:hover</code> )	10
Element Selector ( <code>div</code> , <code>p</code> )	1
Universal Selector ( <code>*</code> )	0

Note: If you want to force a style, use `!important`, Even if an inline style exists,

```
p { color: yellow !important; }
```



# CSS ORDER (CASCADE RULES)

Rule: The closer the style is to the element, the higher its priority.

## 1. Inline CSS (Highest Priority)

- It stays closest to the element
- Override everything except !important

## 2. Internal & External CSS (Depends on Specificity & Order)

- In top down approach the last defined one overrides the previous one.
- For example, if two styles are applied on the same paragraph the second one will get more priority.

## 3. Browser Default Styles (Lowest Priority)



# CSS BACKGROUND

Property	Values
background-color	color-rgb/ color-hex/ color-name/ transparent/ inherit
background-image	url(URL)/ none/ inherit
background-repeat	repeat/repeat-x/repeat-y/no-repeat/inherit
background-position	center/ top left/ top right/ bottom left/ bottom right / x% y% / xpx ypx/ inherit
background-attachment	Scroll/ fixed/ inherit



# EXAMPLE

- Specifies the background color of an element

```
<html>
```

```
<body style="background-color:#b0c4de;" >
```

```
<h1>My CSS web page!</h1>
```

```
<p>Hello world! This is a W3Schools.com  
example.</p>
```

```
</body>
```

```
</html>
```

\* Try *background-image:url(abc.gif), background-position and etc*

# BACKGROUND - SHORTHAND PROPERTY

- It is also possible to specify all the properties in one single property. This is called a shorthand property.
- The shorthand property for background is  
*background:#ffffff url('img\_tree.png') no-repeat right top;*
- order of the property values are:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
- Comment is written using // or "/\*" like C.



# CSS Font

# CSS FONT

- CSS font properties define the font **family, boldness, size, and the style** of a text

Generic Family	Font Family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width



# FONT FAMILY

- The font-family property should **hold several font names** as a **"fallback"** system.
- If the browser **does not support the first font**, it tries the next **font**.
- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.
- If the name of a font family is more than one word, it **must be in quotation marks**, like font-family: "Times New Roman".

# EXAMPLE

```
<html>
```

```
<head>
```

```
  <style type="text/css">
```

```
    p.serif{font-family:"Times New Roman",Times,serif;}
```

```
    p.sansserif{font-family:Arial,Helvetica,sans-serif;}
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>CSS font-family</h1>
```

```
  <p class="serif">This is a paragraph, shown in the Times  
    New Roman font.</p>
```

```
  <p class="sansserif">This is a paragraph, shown in the  
    Arial font.</p>
```

```
</body>
```

```
</html>
```

## CSS font-family

This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.



# FONT STYLE, WEIGHT AND SIZE

- Font-Style: normal | italic | oblique (oblique is very similar to italic, but less supported)
- font-weight: **normal** | **lighter** | **bold** | (*value bet. 100-900*)
- font-size: **relative** | **absolute value**;
- **value** can be defined in different units, including pixels (**px**), em (**em**), percentages (%), and **relative** units.
- If you do not specify a font size, the default size for normal text, like paragraphs, is **16px** (16px=1em).

# RELATIVE & ABSOLUTE FONT SIZE

- Absolute: Does not change with respect to other's change.

Unit	Description
px	Pixels (Most common for precise control)
pt	Points (Used in print, 1pt = 1/72 inch)
cm, mm, in	Centimeters, millimeters, inches (Rarely used)

- Relative: **adjust** changes based on parent elements, screen size, or user settings.

Unit	Description
em	Relative to parent element's font size (1em = parent size)
rem	Relative to the root element ( <code>html</code> tag)
%	Percentage of the parent font size
vw, vh	Viewport width & height (responsive text)

Example: `p { font-size: 1.2em; }`



# BEST PRACTICE

- ✓ Set the base font size in html or body for better scalability.
- ✓ Less use absolute, more use relative units.
- ✓ Use media queries to adjust font sizes for different screen sizes.

- Example:

```
html {  
    font-size: 16px; /* Base size */  
}
```

```
p {  
    font-size: 1rem; /* 16px */  
}  
  
@media (max-width: 600px) {  
    html {  
        font-size: 14px; /* Smaller font on  
small screens */  
    }  
}
```

# CSS TEXT

- **COLOR:** name | rgb(r, g, b) | #rrggbb; To apply text color
- *text-decoration: overline | line-through | underline;* To add or remove line
- *text-indent: 20px;* To apply indentation
- *text-align: center | right | left | justify;*
- *text-transform: uppercase | lowercase | capitalize;*
- *word-spacing: 10px;*
- *letter-spacing: 3px;*
- *white-space: normal | nowrap | pre; // Pre is like the <pre> tag*

**CSS overline**

**~~CSS line-through~~**

**CSS underline**



**END**