

1.5 CASCADING STYLE SHEETS (CSS)

CSS helps to define the presentation of HTML elements as a separate file known as CSS file having .css extension. CSS helps to change formatting of any HTML element by just making changes at one place. All changes made would be reflected automatically to all of the web pages of the website in which that element appeared.

CSS Rules

CSS Rules are the styles that we have to create in order to create style sheets. These rules define appearance of associated HTML element.

Selector {property: value;}

- ❖ Selector is HTML element to which CSS rule is applied.
- ❖ Property specifies the attribute that the user wants to change corresponding to the selector.
- ❖ Property can take specified value. Property and Value are separated by a colon (:). Each declaration is separated by semi colon (;).

Examples: i) P { color : red;} ii) h1 (color : green; font-style : italic } iii) body { color : cyan; font-family : Arial; font- style : 16pt}

Embedding CSS into HTML

Following are the four ways to add CSS to HTML documents:

➤ Inline Style Sheets

Inline Style Sheets are included with HTML element i.e. they are placed inline with the element. To add inline CSS, we have to declare style attribute which can contain any CSS property.

Syntax: <Tagname STYLE = “ Declaration1 ; Declaration2 “> </Tagname>	Example: <p style="color: blue; text-align: left; font-size: 15pt">
--	--

➤ Embedded Style Sheets

Embedded Style Sheets are used to apply same appearance to all occurrence of a specific element. These are defined in element by using the <style> element. The

<style> element must include type attribute. The value of type attribute specifies what type of syntax it includes when rendered by the browser.

| Syntax: | Example: |
|---|---|
| <pre><head><title> </title> <style type = "text/css">CSS Rules/Styles....</head></pre> | <pre><style type="text/css"> p {color:green; text-align: left; font-size: 10pt} h1 { color: red; font-weight: bold} </style></pre> |

➤ External Style Sheets

External Style Sheets are the separate .css files that contain the CSS rules. These files can be linked to any HTML documents using <link> tag with rel attribute.

Syntax:

```
<head><link rel= "stylesheet" type="text/css" href= "url of css file"> </head>
```

In order to create external css and link it to HTML document, follow the following steps:

- ❖ Create a CSS file and define all CSS rules for several HTML elements. Let's name this file as external.css.

```
p { Color: orange; text-align: left; font-size: 10pt;}
h1 { Color: orange; font-weight: bold;}
```

- ❖ Now create HTML document and name it as externaldemo.html.

```
<html><head>      <title> External Style Sheets Demo </title>
<link rel="stylesheet" type="text/css" href="external.css"> </head>
<body>      <h1> External Style Sheets</h1>
<p>External Style Sheets are the separate .css files that contain the CSS
rules.</p> </body>      </html>
```

➤ Imported Style Sheets

Imported Style Sheets allow us to import style rules from other style sheets. To import CSS rules we have to use @import before all the rules in a style sheet.

Syntax:	Example:
<pre><head><title> Title Information </title> <style type="text/css"></pre>	<pre><html><head> <title> External Style Sheets Demo</pre>

<pre>@import URL (cssfilepath) ... CSS rules... </style> </head></style></pre>	<pre></title><style> @import url(external.css); </style> </head> <body> <h1> External Style Sheets</h1> <p>External Style Sheets.</p> </body> </html></pre>
--	---

CSS imported

```
<!DOCTYPE html><html><head><style>
p.ex1 { font: 15px arial, sans-serif;}
p.ex2 {font:italic bold 12px/30px Georgia, serif;}</style></head>
<body>
<p class="ex1">This is a paragraph. This is a paragraph. This is a paragraph. This is
a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a
paragraph.</p>
<p class="ex2">This is a paragraph. This is a paragraph. This is a paragraph. This is
a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a
paragraph.</p></body></html>
```

Output:

This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.

This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.

CSS - Pseudo Classes

CSS pseudo-classes are used to add special effects to some selectors.

Syntax:selector:pseudo-class {property: value}

CSS classes can also be used with pseudo-classes:

Syntax:selector.class:pseudo-class {property: value}

There are following most commonly used pseudo-classes:

Value	Description
:link	Use this class to add special style to an unvisited link.
:visited	Use this class to add special style to a visited link.
:hover	Use this class to add special style to an element when you mouse over it.
:active	Use this class to add special style to an active element.
focus	Use this class to add special style to an element while the element has focus.
:first-child	Use this class to add special style to an element that is the first child of some other element.
:lang	Use this class to specify a language to use in a specified element.

While defining pseudo-classes in a <style>...</style> block, following points should be noted:

- 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.
- Pseudo-class are different from CSS classes but they can be combined.

The :link pseudo-class

```
<style type="text/css">
a:link {color:#000000}
</style>
<a href="/html/index.htm">Black Link</a>
```

The :visited pseudo-class

```
<style type="text/css">
a:visited {color: #006600}
</style>
<a href="/html/index.htm">Click this link</a>
```

When the link is clicked, it will change its color to green.

The :hover pseudo-class

```
<style type="text/css">
a:hover {color: #FFCC00}
</style>
<a href="/html/index.htm">Bring Mouse Here</a>
```

When the mouse is moved over this link and it changes its color to yellow.

The :active pseudo-class

```
<style type="text/css">
a:active {color: #FF00CC}
</style>
<a href="/html/index.htm">Click This Link</a>
```

When this link is clicked the color will be changed to pink.

Inheritance in CSS

In CSS, inheritance controls what happens when no value is specified for a property on an element. The inherit keyword allows authors to explicitly specify inheritance. It works on both inherited and non-inherited properties.

Inherited property: When no value for an inherited property has been specified on an element, the element gets the computed value of that property on its parent element. Only the root element of the document gets the initial value given in the property's summary.

Example: color

Non inherited property: When no value for a non-inherited property has been specified on an element, the element gets the initial value of that property

Example: border

The following are the property values:

- **Inherit:** Sets the property value applied to a selected element to be the same as that of its parent element.
- **Initial:** Sets the property value applied to a selected element to be the same as the value set for that element in the browser's default style sheet. If no value is set by the browser's default style sheet and the property is naturally inherited, then the property value is set to inherit instead.

- **Unset:** Resets the property to its natural value, which means that if the property is naturally inherited it acts like inherit, otherwise it acts like initial.
- **Revert:** Reverts the property to the value it would have had if the current origin had not applied any styles to it. In other words, the property's value is set to the user stylesheet's value for the property (if one is set), otherwise, the property's value is taken from the user agent's default stylesheet.

<pre> Default link color <li class="my-class-1">Inherit the link color <li class="my-class-2">Reset the link color <li class="my-class-3">Unset the link color body { color: green;} .my-class-1 a { color: inherit;} .my-class-2 a { color: initial;} .my-class-3 a { color: unset; } </pre>	<p>Result:</p> <ul style="list-style-type: none"> • Default link color • Inherit the link color • Reset the link color • Unset the link color
--	---

- Set the color of the <body> to green.
- As the color property is naturally inherited, all child elements of body will have the same green color. It's worth noting that browsers set the color of links to blue by default instead of allowing the natural inheritance of the color property, so the first link in our list is blue.
- The second rule sets links within an element with the class my-class-1 to inherit its color from its parent. In this case, it means that the link inherits its color from its parent, which, by default inherits its color from its own parent, which ultimately inherits its color from the <body> element, which had its color set to green by the first rule.
- The third rule selects any links within an element with the class my-class-2 and sets their color to initial. Usually, the initial value set by browsers for the text color is black, so this link is set to black.
- The last rule selects all links within an element with the class my-class-3 and sets their color to unset — we unset the value. Because the color property is a naturally inherited property it acts exactly like setting the value to inherit. As a consequence, this link is set to the same color as the body — green.

CSS Background:

Defines how the background image will behave when scrolling the page. The background image will scroll with the page. It will also position and resize itself according to

the element it's applied to. The background property is specified as one or more background layers, separated by commas. Each layer may include zero or one occurrences of any of the following values: <attachment>, <bg-image>, <position>, <bg-size> and <repeat-style>


The <bg-size> value may only be included immediately after <position>, separated with the '/' character, like this: "center/80%".

The <box> value may be included zero, one, or two times. If included once, it sets both background-origin and background-clip. If it is included twice, the first occurrence sets background-origin, and the second sets background-clip. The <background-color> value may only be included in the last layer specified.

<pre> <p class="topbanner"> Starry sky
 Twinkle twinkle
 Starry sky</p> <p class="warning">Here is a paragraph</p> .warning { background: pink; } .topbanner { background: url("https://mdn.mozillademos.org/files/11983/starsolid.gif") #99f repeat-y fixed; }</pre>	
--	---

Border:


The border shorthand CSS property sets an element's border. It sets the values of border-width, border-style, and border-color.

<pre> <div>I have a border, an outline, AND a box shadow! Amazing, isn't it?</div>★ div { border: 0.5rem outset pink; outline: 0.5rem solid khaki; box-shadow: 0 0 0 2rem skyblue; border-radius: 12px; font: bold 1rem sans-serif; margin: 2rem; padding: 1rem; outline-offset: 0.5rem;}</pre>	
--	---

CSS Colors:

CSS Color is a CSS module that deals with colors, color types, color blending, opacity, and how you can apply these colors and effects to HTML content. Not all CSS properties that take a <color> as a value are part of this module, but they do depend upon it. The <color> CSS data type represents a color in the RGB color space. A <color> may also include an alpha-channel transparency value, indicating how the color should composite with

its background. A `<color>` can be defined in any of the following ways: Using a keyword, Using the RGB cubic-coordinate system and Using the HSL cylindrical-coordinate system

<pre><div style="color:blue; border: 1px dashed currentColor;"> The color of this text is blue. <div style="background:currentColor; height:9px;"></div> This block is surrounded by a blue border. </div></pre>	
--	--

CSS Shadows:

CSS can add shadow to text and to elements.

CSS Text Shadow: The CSS text-shadow property applies shadow to text.

```
h1 { text-shadow: 2px 2px;}
```

To add more than one shadow to the text, add a comma-separated list of shadows.

```
h1 { text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;}
```

Box Shadow: The CSS box-shadow property applies shadow to elements.

```
div { box-shadow: 10px 10px grey;}
```

CSS Text

Text Color: The color property is used to set the color of the text. The color is specified by: a color name - like "red", a HEX value - like "#ff0000" or an RGB value - like "rgb(255,0,0)"

- direction-Specifies the text direction/writing direction
- letter-spacing- Increases or decreases the space between characters in a text
- line-height-Sets the line height
- text-align-Specifies the horizontal alignment of text
- text-decoration-Specifies the decoration added to text
- text-indent-Specifies the indentation of the first line in a text-block
- text-shadow- Specifies the shadow effect added to text

-
- text-transform -Controls the capitalization of text
 - text-overflow- Specifies how overflowed content that is not displayed should be signaled to the user
 - vertical-align- Sets the vertical alignment of an element
 - white-space- Specifies how white-space inside an element is handled
 - word-spacing- Increases or decreases the space between words in a text

```
html { font-size: 10px;}
h1 { font-size: 2.6rem; text-transform: capitalize;}
h1 + p { font-weight: bold;}
p { font-size: 1.4rem; color: red; font-family: Helvetica, Arial, sans-serif;}
```

CSS Transformation

The transform CSS property allows to rotate, scale, skew, or translate an element. It modifies the coordinate space of the CSS visual formatting model. Only transformable elements can be transformed. That is, all elements whose layout is governed by the CSS box model except for: non-replaced inline boxes, table-column boxes, and table-column-group boxes.

Properties:

- none -Defines that there should be no transformation
- matrix(n,n,n,n,n,n)-Defines a 2D transformation, using a matrix of six values
- matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)-Defines a 3D transformation, using a 4x4 matrix of 16 values
- translate(x,y)-Defines a 2D translation
- translate3d(x,y,z)-Defines a 3D translation
- translateX(x)- Defines a translation, using only the value for the X-axis
- translateY(y)- Defines a translation, using only the value for the Y-axis
- translateZ(z)- Defines a 3D translation, using only the value for the Z-axis
- scale(x,y)- Defines a 2D scale transformation
- scale3d(x,y,z)-Defines a 3D scale transformation
- scaleX(x)-Defines a scale transformation by giving a value for the X-axis

- `scaleY(y)`-Defines a scale transformation by giving a value for the Y-axis
- `scaleZ(z)`-Defines a 3D scale transformation by giving a value for the Z-axis
- `rotate(angle)`-Defines a 2D rotation, the angle is specified in the parameter
- `rotate3d(x,y,z,angle)`-Defines a 3D rotation
- `rotateX(angle)`-Defines a 3D rotation along the X-axis
- `rotateY(angle)`-Defines a 3D rotation along the Y-axis
- `rotateZ(angle)`-Defines a 3D rotation along the Z-axis
- `skew(x-angle,y-angle)`-Defines a 2D skew transformation along the X- and the Y-axis
- `skewX(angle)` Defines a 2D skew transformation along the X-axis
- `skewY(angle)` Defines a 2D skew transformation along the Y-axis
- `perspective(n)` Defines a perspective view for a 3D transformed element
- `initial` Sets this property to its default value.
- `inherit` Inherits this property from its parent element.

```
<div>Transformed element</div>
```

```
div { border: solid red;
      transform: translate(30px, 20px) rotate(20deg);
      width: 140px; height: 60px; }
```



CSS Transitions:

CSS transitions provide a way to control animation speed when changing CSS properties. Instead of having property changes take effect immediately, the transition cause the changes in a property to take place over a period of time. To create a transition effect, you must specify two things:

1. the CSS property you want to add an effect to
2. the duration of the effect. If the duration part is not specified, the transition will have no effect, because the default value is 0.

transition-property: Specifies the name or names of the CSS properties to which transitions should be applied. Only properties listed here are animated during transitions; changes to all other properties occur instantaneously as usual.

transition-duration: Specifies the duration over which transitions should occur. You can specify a single duration that applies to all properties during the transition, or multiple values to allow each property to transition over a different period of time.

Example:transition-duration: 1s

transition-timing-function: Specifies a function to define how intermediate values for properties are computed. Timing functions determine how intermediate values of the transition are calculated. Most timing functions can be specified by providing the graph of the corresponding function, as defined by four points defining a cubic bezier. Some of the functions are:

- ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
- linear - specifies a transition effect with the same speed from start to end
- ease-in - specifies a transition effect with a slow start
- ease-out - specifies a transition effect with a slow end
- ease-in-out - specifies a transition effect with a slow start and end
- cubic-bezier(n,n,n,n) - lets to define our own values in a cubic-bezier function

transition-delay: Defines how long to wait between the time a property is changed and the transition actually begins.

CSS Animations

CSS animations make it possible to animate transitions from one CSS style configuration to another. Animations consist of two components, a style describing the CSS animation and a set of keyframes that indicate the start and end states of the animation's style, as well as possible intermediate waypoints.

Advantages of animations over transitions:

- They're easy to use for simple animations; you can create them without even having to know JavaScript.
- The animations run well, even under moderate system load.
- Letting the browser control the animation sequence lets the browser optimize performance and efficiency.

Keyframes are used to create animations. The @keyframes CSS at-rule controls the intermediate steps in a CSS animation sequence by defining styles for keyframes along the animation sequence. This gives more control over the intermediate steps of the animation sequence than transitions. The sub-properties of the animation property are:

- animation-name- Specifies the name of the @keyframes at-rule describing the animation's keyframes.

- animation-duration- Configures the length of time that an animation should take to complete one cycle.
- animation-timing-function- Configures the timing of the animation; that is, how the animation transitions through keyframes, by establishing acceleration curves.
- animation-delay- Configures the delay between the time the element is loaded and the beginning of the animation sequence.
- animation-iteration-count- Configures the number of times the animation should repeat.
- animation-direction- Configures whether or not the animation should alternate direction on each run through the sequence or reset to the start point and repeat itself.
- animation-fill-mode- Configures what values are applied by the animation before and after it is executing.
- animation-play-state- Lets to pause and resume the animation sequence.

```
p { animation-duration: 3s; animation-name: slidein;}
@keyframes slidein { from { margin-left: 100%; width: 300%; }
to { margin-left: 0%; width: 100%; }}
```

In this example the style for the <p> element specifies that the animation should take 3 seconds to execute from start to finish, using the animation-duration property, and that the name of the @keyframes at-rule defining the keyframes for the animation sequence is named "slidein".