Cascading Style Sheets (CSS)

Prepared By Deepali Sonawane, DoCSA, MIT-WPU

Introduction to CSS

- CSS stands for **Cascading Style Sheets**.
- CSS is the language we use to style an **HTML document**.
- CSS describes how **HTML elements** should be displayed.
- CSS saves a lot of work. It can control the layout of **multiple web pages** all at once.
- Cascading style sheets used to describe the **look and formatting** of a document written in a **markup language**.
- Cascading style sheets used to specify particular styles for a **character**, a **word**, a **group of words**, a **page** or a **whole web site**.
- Style rule : Selector { property : value;}
- Ex : h1 {color: blue; font-size: 12px; text-align:center;}

Advantages of CSS

1. **CSS Saves Time:** We can write CSS once and then reuse same sheet in multiple HTML pages.
2. **Pages Load Faster:** If we are using CSS, we do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply to all the occurrences of that tag. Less code means faster download times.
3. **Easy maintenance:** To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
4. **Superior styles to HTML:** CSS has a much wider array of attributes than HTML so can give far better look to the HTML page in comparison of HTML attributes.
5. **Reduce the work**

Types of CSS

1. **Inline CSS**: For inline CSS every style content is in **HTML elements**. It is used for a **limited section**. Whenever the requirements are very small we can use inline CSS. It will affect **only single elements**.
2. **Internal CSS**: In internal CSS the style of CSS is specified in the **<head>** element. It affects all the elements in the **body** section. It is used in the condition when we want a style to be used in the **complete HTML body**.
3. **External CSS**: In external CSS we create a **.css** file and use it in a HTML page as per the requirements. Generally, external CSS are used whenever many of HTML attributes are used. Syntax to include **css file** in html file is <link rel="stylesheet" type="text/css" href="ExternalCSS.css">

Example of Inline CSS

```
<html>
<body>

<h1 style="color:blue; text-align:center;">This is a
heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Example of Internal CSS

```
<html>
<head>
<style>
body {
  background-color: yellow;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Example of External CSS

Code of mystyle.css

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Code of ExternalEg.html

```
<html>
<head>
<link rel="stylesheet" href="mystyle
.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

List of Properties and values

color : blue;

background : yellow;

background-color : cyan;

background-image : url(“c:/Logo.jpg”);

background-position: right top;

border : 1px solid black;

border-style: dotted/dashed/solid/double;

border-color: red;

margin-top: 100px;

margin-bottom: 100px;

margin-right: 150px;

margin-left: 80px;

height: 200px;

width: 50%;

List of Properties and values

text-align: center/left/right/justify;

vertical-align: top/bottom/middle;

text-decoration: underline/overline/line-through/none;

text-transform: uppercase/lowercase/capitalize;

letter-spacing: 3px;

word-spacing: 10px;

font-family: "Times New Roman“;

font-style: normal/italic;

font-size: 40px;

font-weight: normal/bold;

list-style-type: circle/square/upper-roman/lower-alpha;

list-style-image: url(“sqpurple.gif”);

padding: 15px;

Example of CSS id selector

```
<html>
<head>
<style>
#a {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<p id="a">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

Example of CSS class selector

```
<html>
<head>
<style>
.abc {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1 class=“abc”>Red and center-aligned heading</h1>
<p class=“abc”>Red and center-aligned paragraph.</p>
</body>
</html>
```

Example of CSS class selector

```
<html>
<head>
<style>
p.abc {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1 class=“abc”>This heading will not be affected</h1>
<p class=“abc”>This paragraph will be red and center-aligned.</p>
</body>
</html>
```

Example of CSS Universal selector

```
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<p>Every element on the page will be affected by the style.</p>
<p id="a">Me too!</p>
<p class="abc">And me!</p>
</body>
</html>
```

Example of CSS Grouping selector

```
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: RGB(255,0,0);
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<h2>Smaller heading!</h2>

<p>This is a paragraph.</p>
</body>
</html>
```

Example of CSS Borders

```
<html><head>
<style>
p.dotted {border-style: dotted; border-color: red;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid; border-width: 5px;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style></head>
```

```
<body>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>
</body>
</html>
```

CSS Border-Individual Sides

```
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

CSS Margins

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS Rounded Borders

```
p {
  border: 2px solid red;
  border-radius: 12px;
  padding: 5px;
}
```

CSS Box Model

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

CSS Dimension

- CSS dimension define the **size and space** occupied by elements on a webpage.
- The dimension properties like **height, width, max-height, max-width, line-height** and many more are used to define width, height of HTML elements in every screen sizes.
- Ex:

```
div {
  height: 200px;
  width: 50%;
  background-color: blue;
}
```

CSS Display Property

- The **display** property is the most important CSS property for **controlling layout**.
- The display property is used to specify how an element **is shown** on a web page.
- 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**.
- The display property is used to change the default **display behavior** of HTML elements.
- Ex:

```
<html>
<head>
<style>
li {
  display: inline;
}
</style>
</head>
<body>
<p>Display a list of links as a horizontal menu:</p>
<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>
</body>
</html>
```

Display a list of links as a horizontal menu:

HTML CSS JavaScript

CSS Display Property

```
<html>
<head><style>
li {
  display: block;
}
</style></head>
<body>
<p>Display a list of links as a horizontal menu:</p>
<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>
</body>
</html>
```

Display a list of links as a horizontal menu:

- HTML
- CSS
- JavaScript

CSS Position Property

- The position property specifies the type of **positioning method** used for an element.
- There are five different position values: **static, relative, fixed, absolute, sticky**

Ex. :

```
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: static;</h2>
<p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>
<div class="static">
This div element has position: static;
</div>
</body>
</html>
```

CSS Float Property

- The float property is used for **positioning and formatting content** e.g. let an image float left to the text in a container.
- In its simplest use, the float property can be used **to wrap text** around images.
- The float property can have one of the following values
  - left - The element floats to the left of its container
  - right - The element floats to the right of its container
  - none - The element does not float (will be displayed just where it occurs in the text). This is default
  - inherit - The element inherits the float value of its parent

Ex. <html><head><style>

```
img {
  float: right;
}
```

</style></head><body>

<img src="MIT-WPU.jpg" style="width:170px;height:170px;margin-left:15px;">In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image. </body></html>

CSS links/ CSS pseudo classes

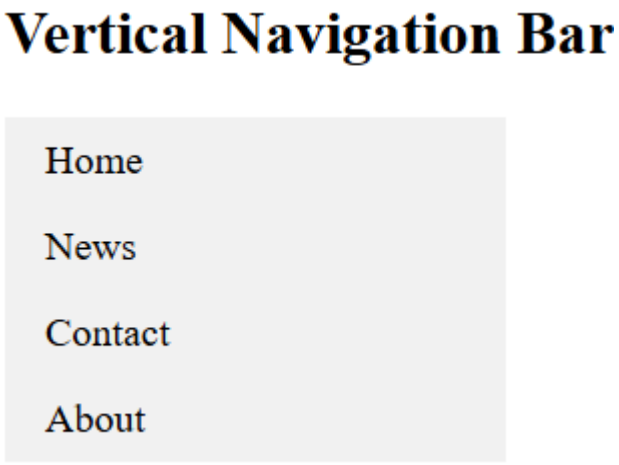
```
<html><head><style>
/* unvisited link */
a:link {
  color: red;
}
/* visited link */
a:visited {
  color: green;
}
/* mouse over link */
a:hover {
  color: pink;
}
/* selected link */
a:active {
  color: blue;
}
</style></head>
<body>
<h2>Styling a link depending on state</h2>
<p><b><a href="#" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>
</body></html>
```

CSS lists

```
<html><head>
<style>
ul.a {
  list-style-type: circle;
}
ul.b {
  list-style-type: square;
}
ol.c {
  list-style-type: upper-roman;
}
ol.d {
  list-style-type: lower-alpha;
}
</style>
</head>
<body>

<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li><li>Tea</li><li>Coca Cola</li>
</ul>
<ul class="b">
  <li>Coffee</li><li>Tea</li><li>Coca Cola</li>
</ul>
<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li><li>Tea</li><li>Coca Cola</li>
</ol>
<ol class="d">
  <li>Coffee</li><li>Tea</li><li>Coca Cola</li>
</ol>
</body></html>
```

CSS Vertical Navigation Bar

```
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
</style>
</head>
<body>
<h2>Vertical Navigation Bar</h2>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```



CSS Horizontal Navigation Bar


```
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}
li {
  float: left;
}
li a {
  display: inline;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
li a:hover {
  background-color: #111;
}
</style>
</head>
<body>
<ul>
  <li><a class="active"
href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

CSS Gradients

- CSS gradients display smooth transitions between two or more specified colors.
- CSS defines three types of gradients:
  - Linear Gradients (goes down/up/left/right/diagonally)
  - Radial Gradients (defined by their center)
  - Conic Gradients (rotated around a center point)

- **CSS Linear gradients:**
- Syntax : background-image: linear-gradient (direction, color-stop1, color-stop2, ...);

```
#grad {
  background-image: linear-gradient(red, yellow);
}

#grad {
  background-image: linear-gradient(to right, red , yellow);
}

#grad {
  background-image: linear-gradient(to bottom right, red, yellow);
}

#grad {
    background-image: linear-gradient(180deg, red, yellow);
  }
```

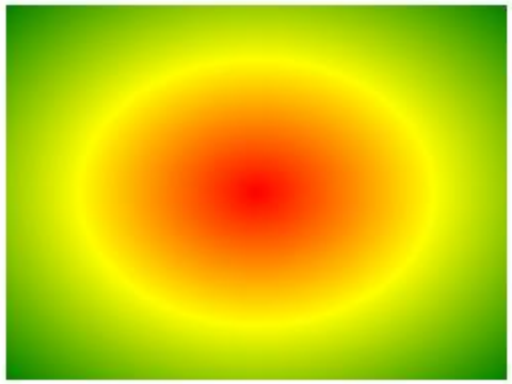
CSS Gradients Example

```
<html>
<head><style>
#grad1 {
  height: 200px;
  background-image: linear-gradient(red, yellow);
}
</style></head> <body>

<h1>Linear Gradient - Top to Bottom</h1>
<p>This linear gradient starts red at the top, transitioning to yellow at the bottom:</p>

<div id="grad1"></div>

</body>
</html>
```

CSS Radial Gradients

- A radial gradient is defined by its **center**.
- To create a radial gradient we must also define at least two colors stops.
- Syntax : background-image: radial-gradient(shape size at position, start-color, ..., last-color);

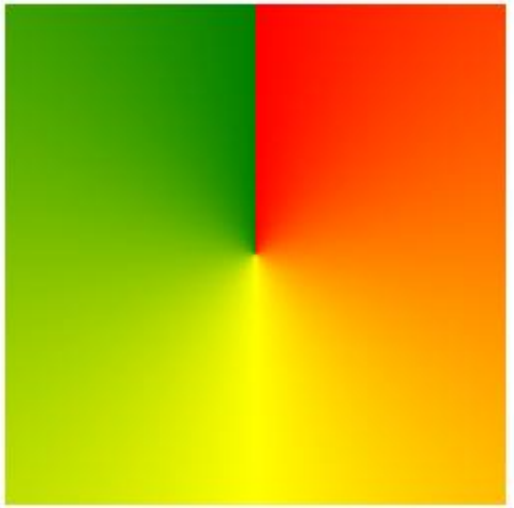
```
#grad {
  background-image: radial-gradient(red, yellow, green);
}

#grad {
  background-image: radial-gradient(red 5%, yellow 15%, green 60%);
}

#grad {
  background-image: radial-gradient(circle, red, yellow, green);
}
```



CSS Conic Gradients

- A conic gradient is a gradient with color transitions **rotated around a center point.**
- To create a conic gradient you must define at least two colors.
- Syntax: `background-image: conic-gradient([from angle] [at position,] color [degree], color [degree], ...);`
- Ex.:

```
#grad {
  background-image: conic-gradient(red, yellow, green);
}

#grad {
  background-image: conic-gradient(red 45deg, yellow 90deg, green 210deg);
}

#grad {
  background-image: conic-gradient(red, yellow, green, blue, black);
}

#grad {
  background-image: repeating-conic-gradient(red 10%, yellow 20%);
  border-radius: 50%;
}
```



CSS Animation

- CSS allows animation of HTML elements **without using JavaScript**.
- Animation is an element gradually change **from one style to another**.
- To use CSS animation, you must first specify some **keyframes** for the animation.
- Keyframes hold what styles the element will have at certain times.

Ex.: <html><head><style>

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style></head><body>
<h1>CSS Animation</h1><div></div></body></html>
```

CSS Animation

Example 2:

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
```

Example 3:

```
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
```

Overview of CSS2

- **Focus**: Basic styling and layout of web pages.
- **Structure**: CSS2 was a single document with all the Cascading Style Sheets information within it.
- **Features**: Provided fundamental styling properties for text, colors, fonts, and basic layout.
- **Example**: Basic text formatting, simple box model, and limited animation capabilities.

Overview of CSS3

- **Focus**: More advanced styling, layout, and visual effects for web pages.
- **Structure**: CSS3 is broken down into modules, each addressing a specific area of styling.
- **Features**:
  - Advanced Layout: Flexible box layout, grid layout.
  - Visual Effects: Transitions, animations, transformations, gradients, shadows, rounded corners.
  - New Selectors: More precise ways to target elements.
  - Expanded Color Options: gradient colors.
  - New Properties: Multiple backgrounds, multi-column layouts, and more.
- **Example**: Smooth transitions, complex animations, 3D transformations, and multi-column layouts.

Thank you