

Cascading Style Sheets (CSS)

A series of horizontal lines in teal and light blue colors, with some lines having a slight 3D effect, positioned below the title.

Cascading Style Sheets (CSS)

- **“CASCADING STYLE SHEET ”**
- STYLING OR DESIGNING OF HTML DOCUMENT
- **USES OF CSS**
 - TIME SAVING
 - EASY MAINTENANCE

WHAT IS CSS?

- SYNTAX:

selector

{

property:value;

}

- Selector: HTML tag
- Property: attribute
- Value: value of attribute you want to change

Cascading Style Sheets (CSS)

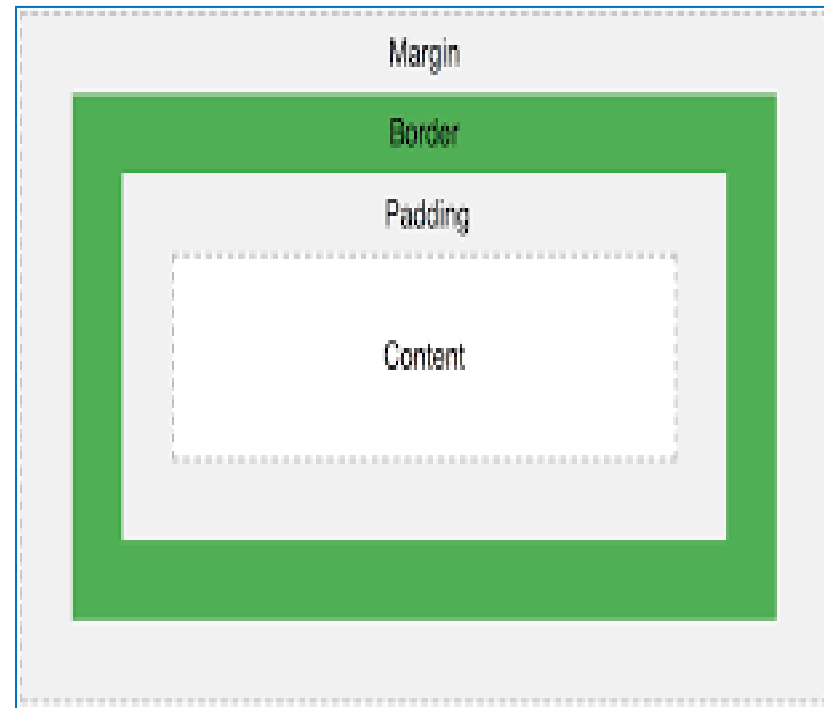
- Style sheets are used to separate presentational and structural information of the web page.
- Cascading style sheets use **selectors** (tags/id/class), followed by a list of properties

- Ex:

```
h1 {  
    color: red;  
    font: italic 1em Times, serif;  
    text-decoration: underline;  
    background: black;  
}
```

CSS BOX MODEL

- CSS Box model Contains:
- Content
 - Main Content like text, images, links etc.
- Padding
 - Goes around content and inside border
- Border
 - It is around padding and content
- Margin
 - Area between webpage boundary and border



Basic Syntax and Structure

- A CSS document consists of a list of rules containing a selector and declaration block.
- Rules
- Classes
- ID
- Pseudo-Class Selectors

Rules

- Syntax of CSS:
 - `Selector { property:value; property:value; .. }`
- The selector is the identifier of the element, followed by a list of paired property:value.
 - `body {color: yellow}`
 - `p {font-family: "sans serif"}` `// multiple words within quotes`
 - `p {` `// break up in different lines`
 `text-align: center;`
 `color: red;`
 `}`
- [Example\(File\)](#)

Rules

- Multiple selectors:
 - `h1, h2 {`
 `color: red;`
 `font: italic lem Times, serif;`
 `text-decoration: underline;`
 `background: black;`
 `}`
 - `h1, h2, h3, h4, h5, h6 {`
 `color: red;`
 `}`

Example(File)

Classes

- Need to make several styles for the HTML elements

- CSS code:

```
.left { text-align: left }
```

- Use it in HTML:

```
<h1 class="left">
```

This heading will be left aligned.

```
</h1>
```

```
<p class="left">
```

This paragraph will also be left aligned.

```
</p>
```

[Example1\(File\)](#)

[Example2\(File\)](#)

ID

- An ID selector will apply a specific style to an identified element.
 - CSS code:

```
#bluepara {  
    text-align: center;  
    color: red;  
}
```
 - Use it in HTML:

```
<p id="bluepara">  
    Some interesting thoughts....  
</p>
```

[Example \(File\)](#)

Class Vs. ID

- Class may apply to several parts/elements of a page whereas id applies only to one element and it should be unique.
- For example, it would be **wrong** to do this with the same id:
 - `<h2 id="alert"> This is an alert! </h2>`
 - `<p id="alert"> This is an alert! </p>`
 - `<h3 id="alert"> This is an alert! </h3>`

Using CSS

- To use a CSS, we can either embed the rules in the HTML document or have a totally separate file.
- External Style Sheets
- Embedded Style Sheets
- Inline style sheets

External CSS

- A separate external CSS file
- Extension .css
- We can change the look of the entire site by altering only one file.
- Each page that uses the style sheet needs to have a link to associate it with the style document

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css" />

</head>

- [Example](#) ([CSS File](#))

INLINE STYLESHEET

- WRITTEN DIRECTLY IN TAG

- Example:

```
<p style="color:blue">  
    text here  
</p>
```

Embedded CSS

- Internal style sheet
- Style information is attached to only one document.

```
<head>
```

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

```
    hr { color: sienna }
```

```
    p { margin-left: 20px }
```

```
    body { background-image: url(picture1.jpg) }
```

```
  </style>
```

```
</head>
```

- [Example](#)

CSS TEXT & PROPERTIES

- **Color**
- **Text-align**
- **Text-decoration**
- **Text-transform**
- **Text-indent**
- **Text-shadow**
- **Letter-spacing**
- **Line-height**

```
P
{
Color:blue;
Text-align:right;
Text-decoration:underline;
Text-transform:lowercase;
Text-indent:10px;
Text-shadow:2px 2px 2px
#ff1234;
Letter-spacing:2px;
Line-height:60%
}
```


CSS FONTS

- **Font**
- **Font-family**
- **Font-size**
- **Font-style**
- **Font-variant**
- **Font-weight**

```
P
{
  Font:italic bold 12px serif;
  Font-family:"Times New
  Roman","serif";
  Font-size:small;
  Font-style:normal;
  Font-variant:normal;
  Font-weight:bold;
}
```

CSS LINKS

- **Unvisited Link**

a:link {color: #123456;}

- **Visited Link**

a:visited {color:#FF0500;}

- **Mouse Over Link**

a:hover { color:#1234FF;}

- **Selected Link**

a:active { color:#FF10FF;}

Pseudo-Class Selectors

- Some selectors can be considered different because of the way the element they belong to works.
- For example, the anchor that creates a link can have pseudo classes attached to it.
- It could be visited, not visited or in process of being selected.
- To catch these states, pseudo-class selectors can be used:
 - `a:link {color: red }`
 - `a:active {color: yellow }`
 - `a:visited {color: green }`
- A link would be initially red, when visited it would be green, in the process of being clicked it would be yellow.
- To make a link bold when the cursor is above it,
 - `a:hover {font-weight: bold }`
 - `a:link:hover {font-weight: bold }` // specific to links only

CSS BACKGROUND

- **Background:** write color or url of image
- **Background-attachment:** attachment is fixed or not
- **Background-color:** color of background
- **Background-image:** url of image
- **Background-position:** center, left, right
- **Background-repeat:** repeat or not repeat

CSS BORDERSTYLE

none: Defines no border

dotted: Defines a dotted border

dashed: Defines a dashed border

solid: Defines a solid border

double: Defines two borders. The width of the two borders are the same as the border-width value

groove: Defines a 3D grooved border. The effect depends on the border-color value

ridge: Defines a 3D ridged border. The effect depends on the border-color value

inset: Defines a 3D inset border. The effect depends on the border-color value

outset: Defines a 3D outset border. The effect depends on the border-color value

Background Color

- Sets the background color of the element.

```
body {  
    background: #F0F8FF;  
}
```

Or

```
body {  
    background: yellow;  
}
```

- CSS standard supports 16 color names: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white and yellow.

Background Images

- [Example](#) ([CSS file](#))

```
body {  
    background: #F4a460;  
    background-image: url("Picture1.jpg");  
    background-repeat: no-repeat;  
}
```

Background Images

Property	Description	Values
Background	Sets all background properties in one declaration	As RGB
Background-attachment	Sets whether image moves with page when scrolled	Scroll or fixed
background-color	Sets the background color of an element	RGB, hex, name or transparent
background-image	Sets an image in the background	URL or none
Background-position	Sets the starting position of an image in background	Top left/top center/top right/center left/center center/center right/bottom left/bottom center/bottom right/ x-% y-% / x-pos y-pos
background-repeat	Sets the repetition of an image used in the background	Repeat Repeat-x Repeat-y No-repeat

Manipulating Text

- Set the color of the text

```
h1 { color: #00ff00 }
```

```
h2 { color: yellow }
```

```
h3 { color: rgb(255,20,40) }
```

```
P { color: rgb(0,0,255) }
```

- The background color of the text can also be changed:

```
h3 { color: rgb(255,20,40) ; background-color: yellow }
```

- Letter spacing in the Text is done by:

```
h1 { color: #00ff00; letter-spacing: -3px }    (decreased spacing)
```

```
h2 { color: yellow; letter-spacing: 0.6px }    (increased spacing)
```

Manipulating Text

- Text Alignment:

h1 {text-align: center}

h2 {text-align: left}

h3 {text-align: right}

- Text decoration:

h1 {text-decoration: overline }

h2 {text-decoration: line-through}

h3 {text-decoration: underline }

a {text-decoration: none }

Manipulating Text

- Text Indentation:

`p {text-indent: 1cm}`

- Text Case:

`p.uppercase {text-transform: uppercase }`

`p.lowercase {text-transform: lowercase }`

`p.capitalize {text-transform: capitalize}`

- Example

Fonts

- Font features: [Example](#)
 - `h3 {`
 `font-family: times;`
 `font-size: 150%;`
 `font-style: normal`
 `}`
 - `p {`
 `font-family: courier;`
 `font-size: 100%;`
 `font-style: oblique`
 `}`
 - `p.sansserif {`
 `font-family: sans-serif;`
 `font-style: italic`
 `}`

Borders and Boxes

- The border properties allow many styles of border to be placed around areas.

Property	Description	Value
Border	Sets all properties for four borders in one declaration	Border-width Border-style Border-color
Border-bottom	Sets all properties for bottom border in one declaration	Border-bottom-width Border-style Border-color
Border-bottom-color	Sets bottom border color	Border-color
Border-bottom-style	Sets bottom border style	None/hidden/dotted/dashed/solid/double/groove/ridge/inset/outset
Border-color	Sets the color of borders	Color

Borders and Boxes

Property	Description	Value
Border-style	Sets style for the four borders	None/hidden/dotted/dashed/solid/double/groove/ridge/inset/outset
Border-left	Sets all properties for left side border	Border-left-width Border-style Border-color
Border-right	Sets all properties for right side border	Border-right-width Border-style Border-color
Border-top	Sets all properties for top side border	Border-top-width Border-style Border-color
Border-width	Sets width for all borders	Thin/medium/thick/length

Margins

- Margin is the space between one element and another.

```
p.margin { margin-top: 5cm }
```

- margins can be set for top, bottom, left and right.

```
p.margin { margin: 2cm 4cm 3cm 4cm }
```

Padding

- Padding allows us to insert extra space around the contents of the element but inside the border.
- We can change only the padding zone's thickness, but not its border.
 - `td {padding-bottom: 2cm; padding-top:2cm }`
- [Example](#)

Lists

- It is possible to alter the list item marker and where the marker is placed. ([Example](#))
 - `ul.disk { list-style-type: disc }`
 - `ul.circle { list-style-type: circle }`
 - `ul.square { list-style-type: square }`
 - `ul.none { list-style-type: none }`

Property	Description	Values
List-style-image	Use an image as a marker	None, url
List-style-position	Positioning of marker	Inside, outside
List-style-type	Type of marker	None, disc, circle, square, decimal, ..
List-style	Set all properties	All above properties

Positioning using CSS

- Unlike HTML, CSS allows good control over the positioning of elements.
- CSS controls exactly where elements will appear.
- Absolute positioning
 - The positioning sets the top left of the element
 - [Example](#)
- Relative positioning
 - Elements can be placed relatively; in relation to its normal position.
 - position: relative;
left/top/right/bottom: n
 - [Example](#)

Positioning using CSS

- Z-Index positioning

- It is possible not only to position elements in x and y coordinates, but in terms of depth of the screen too so things can appear to be behind or in front of the others.
- A Z-Index property has a number that describes where the object is in the stack of visible elements.
- Example
- In example, the heading is at depth 2 so is most visible, followed by the image at 1 and finally the furthest back is the paragraph.

Positioning using CSS

- Shaping an Element

- An element, such as an image, can also be shaped with the clip property.
- `img {
 position: absolute;
 clip: rect(0px 50px 200px 0px)
}`

Positioning using CSS

- Layout and Structure

- It is important to maintain structure within the document that will be styled by CSS.
- Break the page into logical sections with div (division) elements.
- [Example](#)