CASCADING STYLE SHEETS

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS defines **how HTML elements are to be displayed**
- CSS saves a lot of work(The style definitions are normally saved in external .css files.)

- When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS.
- In HTML 4.0, all formatting could (and should!) be removed from the HTML document, and stored in a separate CSS file. With an external style sheet file, you can change the look of an entire Web site by changing just one file!
-

CSS Syntax

- A CSS rule set consists of a selector and a declaration block:

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

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: red;
    text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>

</body>
</html>
```

CSS Selectors

- The element Selector
- p {
  text-align: center;
  color: red;
  }
- The id Selector
- #para1 {
  text-align: center;
  color: red;
  }

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

class Selector

- selects elements with a specific class attribute.
- <!DOCTYPE html>
- <html>
- <head>
- <style>
- .center {
- text-align: center;
- color: red;
- }
- </style>
- </head>
- <body>

- <h1 class="center">Red and center-aligned heading</h1>
- Red and center-aligned paragraph.

- </body>
- </html>

- all <p> elements with class="center" will be center-aligned:
- p.center {
  text-align: center;
  color: red;
  }

- You can group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

- h1, h2, p {
  text-align: center;
  color: red;
  }

- When a browser reads a style sheet, it will format the document according to the information in the style sheet.

- There are three ways of inserting a style sheet:
- External style sheet
- Internal style sheet
- Inline style

EXTERNAL STYLE SHEET

- An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.
- Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:
- <head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  </head>

- An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file called "myStyle.css", is shown below:
- ```
  body {
      background-color: lightblue;
  }
  h1 {
      color: navy;
      margin-left: 20px;
  }
  ```
  ```

Internal Style Sheet

- An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: red;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>

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

</body>
</html>
```

Inline Styles

- An inline style loses many of the advantages of a style sheet
- To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a h1 element:
- <h1 style="color:blue;margin-left:30px;">This is a heading.</h1>

Multiple Style Sheets

- If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.
- For example, assume that an external style sheet has the following properties for the <h1> element:
- h1 {
  color: navy;
  margin-left: 20px;
  }

- then, assume that an internal style sheet also has the following property for the <h1> element:
- h1 {
  color: orange;
  }
- If the page with the internal style sheet also links to the external style sheet the properties for the <h1> element will be:
- color: orange;
  margin-left: 20px;
- The left margin is inherited from the external style sheet and the color is replaced by the internal style sheet.

Cascading order

- What style will be used when there is more than one style specified for an HTML element?
- Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:
- Browser default
- External style sheet
- Internal style sheet (in the head section)
- Inline style (inside an HTML element)

- So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).
- **Note:** If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: #b0c4de;
     background-image: url("paper.gif");
    background-repeat: no-repeat;
    background-position: right top;
}
</style>
</head>
<body>

<h1>My CSS web page!</h1>
<p>Hello world!.</p>

</body>
</html>
```

- If the image above is repeated only horizontally (background-repeat: repeat-x;)

- To repeat an image vertically, set background-repeat: repeat-y;
- The position of the image is specified by the background-position property

- body {
  background: #ffffff url("img_tree.png")
  no-repeat right top;
  }

TEXT FORMATTING

- Text Color
- Text Alignment: The text-align property is used to set the horizontal alignment of a text., is used to set the horizontal alignment of a text. left or right aligned, centered, or justified. When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):
- Text Decoration: The text-decoration property is used to set or remove decorations from text.

  none, overline,line-through,underline
- Text Transformation: The text-transform property uppercase,lowercase,capitalize
- Text Indentation: The text-indent property is used to specify the indentation of the first line of a text
- Text-indent:5px
- we are learning html and css as a part of web development in cdac noida.

- Font Family: font-family: “Times New Roman, ”Verdana”
- Font Style: font-style: Normal,italic,bold,oblique
- Font Size: font-size: 10 px,
- Set Font Size With Em:To allow users to resize the text (in the browser menu), many developers use em instead of pixels.
- 1 em=16 px.
- h1 {
  font-size: 2.5em;
  }

Styling Links

- The four links states are:
- 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

```
<style>
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
</style>
</head>
```

- <style>
- ul {
- list-style-image: url('sqpurple.gif');
- }
- </style>

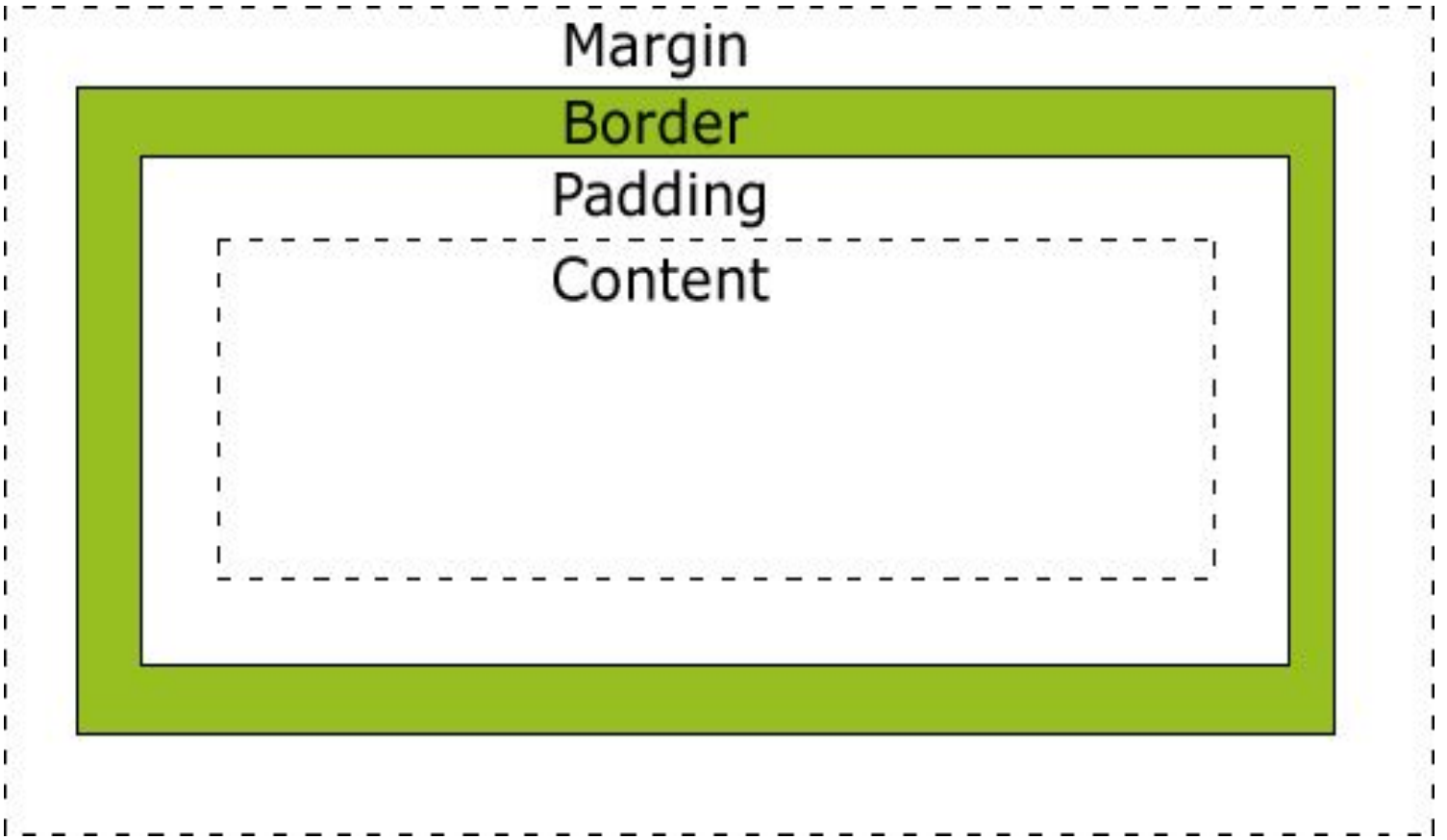
- Table Borders
- table, th, td {
  border: 1px solid black;
  }
- Notice that the table in the example above has double borders. This is because both the table and the <th>/<td> elements have separate borders.

- To display a single border for the table, use the border-collapse property.
-

```
table {
    border-collapse: collapse;
}
```

The CSS Box Model

- All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.
- The box model allows us to add a border around elements, and to define space between elements.
- The image below illustrates the box model:


Margin
Border
Padding
Content

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: lightgrey;
    width: 300px;
    padding: 25px;
    border: 25px solid navy;
    margin: 25px;
}
</style>
</head>
<body>

<div>cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdac cdacdiv>

</body>
</html>
```

- Width and Height of an Element
- In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.
- **Important:** When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the full size of an element, you must also add padding, borders and margins.

- The total width of an element should be calculated like this:
- Total element width = width + left padding + right padding + left border + right border + left margin + right margin
- The total height of an element should be calculated like this:
- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Border Properties

- Border Style: border-style:dotted
- 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, ridge, inset,outset:Defines a 3D ridged border. The effect depends on the border-color value

- **Border Width**
- The width is set in pixels, or
- three pre-defined values: thin, medium, or thick.
- **Border color**
- border-style: solid;
  border-color: red;

- **Border - Individual sides**
- border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-width: 5px;
    border-color: red;
}
p.two {
    border-style: solid;
    border-width: medium;
}
p.three {
    border-style: solid;
    border-width: 1px;
}
</style>
```

- </head>
- <body>

- Some text.
- Some text.
- Some text.
- Note: The "border-width" property does not work if it is used alone. You must add the "border-style" property to set the borders first.

- </body>
- </html>

- The border-style property can have from one to four values.
- **border-style: dotted solid double dashed;**
  - top border is dotted
  - right border is solid
  - bottom border is double
  - left border is dashed

- **border-style: dotted solid double;**
  - top border is dotted
  - right and left borders are solid
  - bottom border is double

- **border-style: dotted solid;**
  - top and bottom borders are dotted
  - right and left borders are solid
-

- **border-style: dotted;**
  - all four borders are dotted

- To shorten the code, it is also possible to specify all the individual border properties in one property.
- p {
- border-style:solid
- border-color:red
  border: 5px solid red;
  }

Margin

- The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.
- The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.

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

.ex {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 50px;
}
</style>
</head>
<body>

<p>This is a paragraph with no specified margins.</p>
<p class="ex">This is a paragraph with specified margins.</p>

</body>
</html>
```

- The CSS padding properties define the space between the element border and the element content. The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.
- The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

- p {
  padding-top: 25px;
  padding-right: 50px;
  padding-bottom: 25px;
  padding-left: 50px;
  }

CSS Dimension

- p.ex {
  height: 100px;
  width: 100px;
  }
-

CSS Display and Visibility

- The display property specifies if/how an element is displayed, and the visibility property specifies if an element should be visible or hidden.
- Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:
- visibility: hidden hides an element, but it will still take up the same space as before. The element will be hidden, but still affect the layout:

- display:none hides an element, and it will not take up any space. The element will be hidden, and the page will be displayed as if the element is not there:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1.hidden {
    display: none;
}
</style>
</head>
<body>

<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading does not take up space.</p>

</body>
</html>
```

- **This is a visible heading**
- Notice that the hidden heading does not take up space.

- **CSS DISPLAY - BLOCK AND INLINE ELEMENTS**

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
    display: inline;
}
</style>
</head>
```

- <body>

- Display a list of links as a horizontal menu:
-
- HTML
- CSS
- JavaScript
-

- </body>
- </html>

- Display a list of links as a horizontal menu:
- HTML CSS JavaScript

```
<!DOCTYPE html>
<html>
<head>
<style>
span {
    display: block;
}
</style>
</head>
<body>

<span>A display property with a value of "block" results in</span> <span>a line break between the two elements.</span>

</body>
</html>
```

- A display property with a value of "block" results in a line break between the two elements.

Center Aligning Using the margin Property

- .center {
  margin-left: auto;
  margin-right: auto;
  width: 70%;
  background-color: #b0e0e6;
  }

- **Descendant Selector:**matches all element that are descendants of a specified element.
- div p {
  background-color: yellow;
  }
- **Child Selector:**selects all elements that are the immediate children of a specified element.

- div > p {
  background-color: yellow;
  }
- **Adjacent Sibling Selects:**selects all elements that are the adjacent siblings of a specified element.
- div + p {
  background-color: yellow;
  }

- **General Sibling Selector:**selects all elements that are siblings of a specified element.