CSS

What is CSS ?

CSS stands for "Cascading Style Sheets"

Cascading: refers to the procedure that determines which style will apply to a certain section, if you have more than one style rule.

Style: how you want a certain part of your page to look. You can set things like color, margins, font, etc for things like tables, paragraphs, and headings.

Sheets: the "sheets" are like templates, or a set of rules, for determining how the webpage will look.

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML.

CSS

History

- CSS1 was the first edition introduced in 1996.

- CSS2 was published in 1998 and provides enhancement over CSS1.

- CSS2.1 was the last 2nd generation edition of CSS.

- CSS 3 is the latest edition. Several new functionalities have been provided through CSS3.

  Functions like rounded corners, backgroubnd decoration, box shadows, which are demonstrated in the subsequesnt sections, are introduced in this version.

CSS

Advantages

- A web application will contains hundreds of web pages, which are created using HTML.

- Formatting these HTML pages will be a laborious process, as formatting elements need to be applied to each and every page.

- CSS saves lots of work as we can change the appearance and layout of all the web pages by editing just one single CSS file.
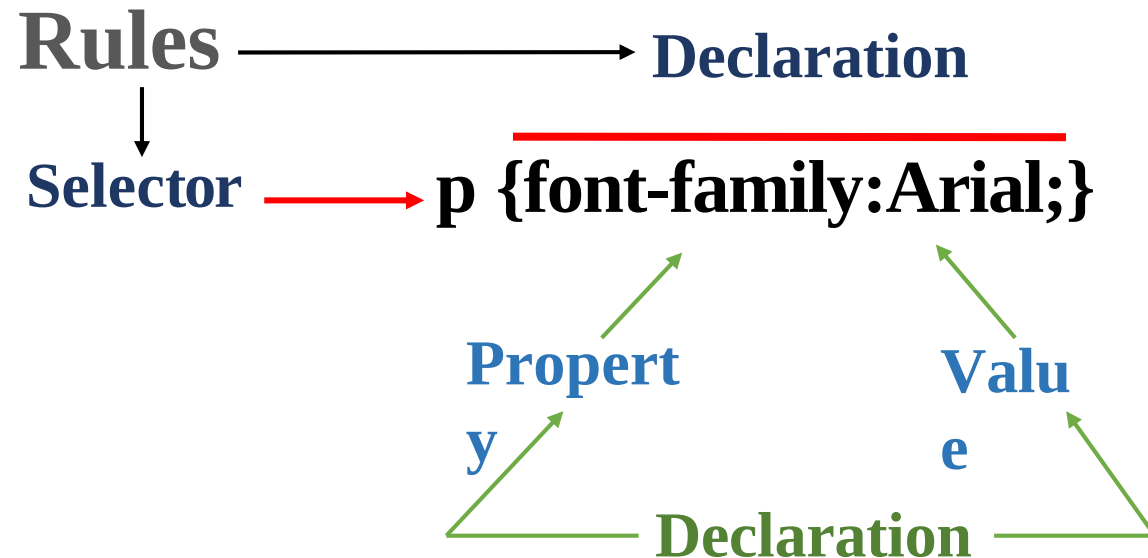
CSS Syntax Rules

Rule have two parts - Selector and declaration.

Selector: The HTML element you want to add style to.
<p> <h1> <table> etc

Declaration: The statement of style for that element. Made up of property and value.

Rules ⟶ **Declaration**

Selector ⟶ **p {font-family:Arial;}**

Property **Value**

Declaration

Refer Note Section-

CSS Style Example

Selector - I want the text color of my paragraph to be <u>red</u> and the background color to be <u>black</u>.

```
<html>
<head>
<style> p {font-family:Arial; color: red;
background-
color:black;} </style>
</head>
<body>
<p> <b> Welcome to Java Academy </b>
</p>
</body>
</html>
```

Welcome to Java Academy

CSS Selectors

CSS selectors allow you to select and manipulate HTML elements based on their id, class, type, attribute, and more.

Examples –

Declaring a CSS Rule for a Class Attribute

the HTML
Brochure

the CSS
.pdf {background: url(images/pdf.gif) no-repeat left 50%;}

use a period when writing a rule for a class

Declaring a CSS Rule for an Id Attribute

the HTML
<div id="wrapper">Main Content</div>

the CSS
#wrapper {width: 750px; margin: 0 auto;}

use a pound sign when writing a rule for a id

CSS Selectors (Cont.)

Declaring a CSS Rule for a Elements Attribute

It will style all the content of that element which you are selecting.

Example-

The HTML
\<p\> Welcome to the Java Academy **\</p\>**
\<p\> \<b\>\<i\> Powered by – Foo **\</i\>\</b\>\</p\>**

The CSS
p {text-align: center; color: blue;**}**

CSS Selectors (Cont.)

Grouping Selectors

You can group all the selectors of same style to minimize the code. The selectors should be separated with comma.

Example-

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

Grouped Selectors-
h2, p {text-align: center; color: red; **}**

Inserting a StyleSheet

You can do in three different ways-

1. **External Style Sheet**

   Styles are specified in an external CSS file. you can change the looks of entire website by using single external style sheet.
   Eg.: **<head>** <link rel="stylesheet" type="text/css" href="ex1.css" /> **</head>**

2. **Internal Style Sheet**

   To Appy specific styles to a single HTML file inside the head section of an HTML page.
   Eg.: **<style>** **p { text-align**:left; **font-size**:24px; **} </style>**

3. **Inline Styles**

   Styles are specified inside an HTML tag/element.
   Eg.: **<p** style="**font-family**:Algerian; **font-size**:28px;"> Demo of Inline Style **</p>**

Inserting a StyleSheet

Multiple Style Sheets – It can be referenced inside an HTML document.

The questions is, what styles will be applicable when there is more than one style specified?

All styles cascade into a new virtual style sheet by applying the following rules, where the higher number has the greater priority:

1. Browser default.
2. External Stylesheet.
3. Internal Stylesheet (styles defined in head section).
4. Inline Style (styles defined in an HTML element).

Ref.Note Section-

Formatting with CSS Properties

CSS Background

We can use CSS Background properties to define the background effects of an element.

The following properties can be used for background effects :

a. background-color
b. background-image
c. background-repeat
d. background-position

Formatting with CSS Properties

CSS Background Image

You can use an image as the background for an element using background-image property.

Example-

```
body{
      background-image:url('java.png');
      }
```

By default, the image is repeated, both horizontally and vertically, so as to cover the entire body (or the element on which it is applied).

Formatting with CSS Properties

CSS Background Color

The **background-color** property is used to specify the background color of an element.

Example-

body {

    background-color:darkblue;

    }

Similarly, we can specify the background for any element (wherever applicable). p {

   background-color:orange;

   }

Formatting with CSS Properties

CSS Background Position

If the background image disturbs the text, i.e. if the text cannot be read clearly due to the image in the background, we can set the position of the background image.

Example-

```
body {
        background-image:url("snapdeal.jpg");  background-repeat:no-repeat;
        background-position:right top;
        }
```

Formatting with CSS Properties

CSS Background Shorthand

You can also specify all the properties in a single property. This property is known as shorthand property.

For specifying shorthand property, you just need to use **background**.

Example-

body {
    background:cyan url('snapdeal.jpg') **no-repeat right top**;
    }

Formatting with CSS Properties

Text Formatting

The following properties can be used for formatting
text :

1. Text Color
2. Text Alignment
3. Text Decoration
4. Text Transformation
5. Text Indentation

Formatting with CSS Properties

Text Alignment

We can either align the text to the left, right, center or we can make it justified.

Example-
p { **text-align**:left;} h1{**text-align**:center;}

Text Color

The color property is used to set the color of text.

Example-
body { **color**:blue;} p1 {**color**:magenta;}

Formatting with CSS Properties

Text Decoration

You can use **text-decoration** property to set or remove decorations from text.

Example-
p {**text-decoration:**overline;}
p {**text-decoration:**line-through;} p {**text-decoration:**underline;}

Text Transformation

You can use text-transform property to specify uppercase and lowercase letters of any text.

Example-
h1 {**text-transform:**uppercase;} h2 {**text-transform:**lowercase;} p {**text-transform:**capitalize;}

Formatting with CSS Properties

CSS Font

CSS font properties are used to define the font family, size, style and boldness of the text.
In CSS, there are two types of font family names:

generic family - a group of font families with a similar look (like "Serif" or "Monospace").
font family - a specific font family (like "Times New Roman" or "Arial").

Comments in CSS

/* comment */ - This is comment used in CSS.

Formatting with CSS Properties

CSS 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.
Example :
p { font-family:"Arial", Times, "Sans-serif ";}

CSS Font Style

You can use the property font-style to specify mostly italic text. It has three values – Normal, Italic, Oblique (similar to italic).

Formatting with CSS Properties

CSS Font Size

You can use the **font-size** property to set the size of text. The font-size value can be absolute or it can be relative.

Example- h1 {
    font-size: 30px;
}

p {
    font-size: 14px;
}

Formatting with CSS Properties

CSS Font Size with em (Relative Size)

You may face resizing problems, when you use older versions of browsers. To avoid such problems, you can use set font size using em, instead of pixels.

The em size unit is a W3C recommendation.1 em is equal to the current font size.
 The default text size is 16 px. So, the default size of 1 em is 16 px.

Example
h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
}

p {

Formatting with CSS Properties

CSS Links

You can use CSS styles to style any link. Links can be
  styled in different ways by using any CSS property like color, font-family etc.

Links can be in one of the following states :

- **a: link** – Unvisited link
- **a: visited** – A link that the user has visited
- **a: hover** – A link over which the mouse pointer is moving
- **a: active** – A link, which has been just clicked

Links can be styled according to their states.

Formatting with CSS Properties

CSS Links

```
a {
 font-weight:
 bold;
}
a:link {
  color: black;
}
a:visited {
 color: gray;
}
a:hover {
 text-decoration: none;
  color: white;
 background-color:
 navy;
}
a:active {  color:
 aqua;
background-color:
 navy;
}
```

Styling Links

link - before a visit

visited - after it has been visited

hover - when your mouse is over it but you have not clicked

active - you have clicked it and you have not yet seen the new page

Formatting with CSS Properties

CSS List

You can use CSS list properties for

➢ Setting different list item markers for ordered lists

➢ Setting different list item markers for unordered lists

➢ Set an image as the list item marker

Values-
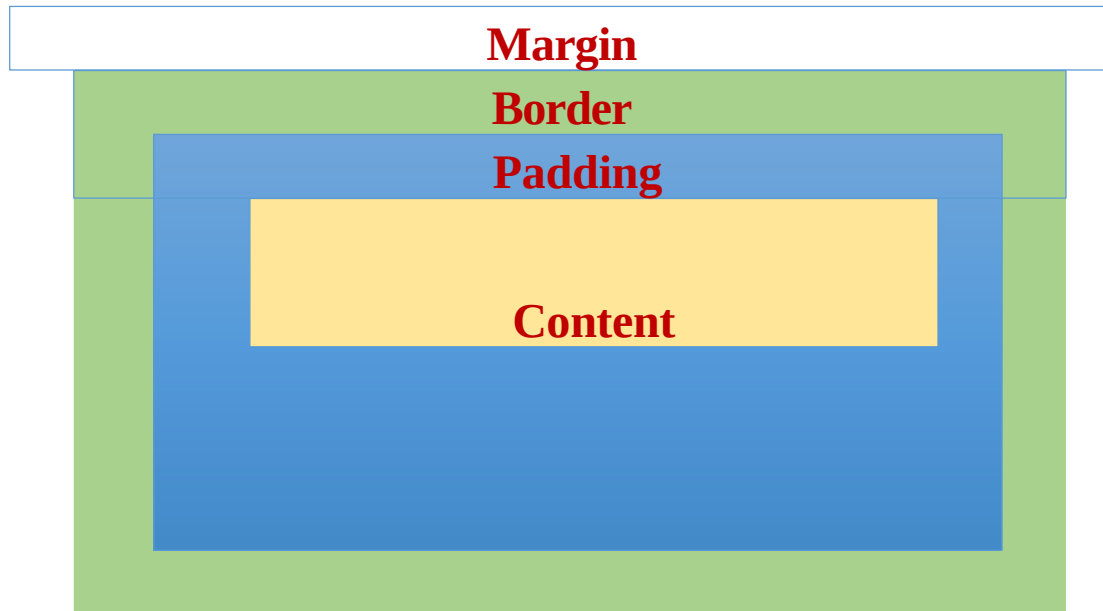
❖ list-style-type
❖ list-style-image

Formatting with CSS Properties

Box Model : Introduction

Box model is useful for designing the layout of an HTML Page.
  CSS Box model describes a box that wraps around HTML elements.

Using this model, we can define the margins, borders, padding and the actual content. We can place border around elements and space elements in relation to each other.

Margin

Border

Padding

Content

Box Model : Illustration

You can set the height and width of an element using the **height** and **width** properties.

Formatting with CSS Properties

CSS Padding

You can use the CSS padding properties to define the space between the element border and the element content. It is possible to change the top, right, bottom and left padding independently using separate properties.

You can also use a shorthand padding property to change all paddings in a single statement. Individual padding properties can be specified as follows :

padding-top:20px; padding-bottom:30px; padding-right:25px; padding-left:10px;

In shorthand-
padding : 20px 30px 25px 10px;

Formatting with CSS Properties

CSS Border

You can use the CSS Border properties to specify the style and color of an element's border.

Values-

border-style border-width border-color

Formatting with CSS Properties

CSS Margin

Using CSS Margin properties you can specify the space around elements.

Values:

**margin-top:50px; margin-bottom:30px; margin-right:25px;
margin-left:10px;**

In shorthand-
margin:50px 30px 25px 10px;

Formatting with CSS Properties

CSS Margin

Using CSS Margin properties you can specify the space around elements.

Values:

margin-top:50px; margin-bottom:30px; margin-right:25px; margin-left:10px;

In shorthand-
margin:50px 30px 25px 10px;

Formatting with CSS Properties

Pseudo-Class

A pseudo-class is used to define a special state of an element.
- Style an element when a user mouses over it.
- Style visited and unvisited links differently.

```
/* 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;
}
```

Formatting with CSS Properties

Pseudo-Elements

A CSS pseudo-element is used to style specified parts of an element.

- Style the first letter, or line, of   an element
- Insert content before, or after, the content of   an element

The **::first-line** pseudo-element is used to add a special style to the first line of

a text. All Pseudo Elements-

Selector	Example	Example description
::after	p::after	Insert content after every <p> element
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

Formatting with CSS Properties

Media Types

The @media rule makes it possible to define different style rules for different media types in the same stylesheet.

Example-

```
@media screen {
    p { font-family: verdana, sans-serif;  font-size: 20px;
} }
```

```
@media print {
    p { font-family: georgia, serif;  font-size: 15px;
    color: blue;
} }
```

CSS3 Introduction

Several new functionalities have been added in CSS 3 which has been split into "modules".

It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.

Some Important CSS3 modules are:

Selectors

Box Model

Backgrounds and Borders

Image Values and Replaced Content Text Effects

2D/3D Transformations Animations

Multiple Column Layout

User Interface

CSS3
Gradient

Now with CSS3 you can use gradients effects in your document. For which earlier you have to use images. It reduces download time and bandwidth usage. Gradient is generated by browser. Its of two types –

- Linear Gradients (goes down/up/left/right/diagonally).

- Radial Gradients (defined by their center).

Example- (This is by-default for Top-Bottom).

#grad {background: linear-gradient(red, blue); } /* Standard Syntax*/

For Different browser support we need to specify the prefix for them-

#grad {background: **-webkit-**linear-gradient(red, blue); /* For Safari 5.1 to 6.0 */

background: **-o-linear-**gradient(red, blue); /* For Opera 11.1 to 12.0 */

background: **-moz-**linear-gradient(red, blue); /* For Firefox 3.6 to 15 */}

CSS3 Text

CSS3 contains several new text features. Their are following text properties- **text-overflow, word-wrap, word-break**

text-overflow – This property specifies how overflowed content that is not displayed should be signaled to the user.

Eg.:

```
p.test1 {
    white-space: nowrap;  width: 200px;
    border: 1px solid #000000;  overflow: hidden
    text-overflow: clip;
}
```

It can be clipped:

> This is some long text that

or it can be rendered as an ellipsis (…):

> This is some long text th…

```
p.test2 {
    white-space: nowrap;
    width: 200px;
    border: 1px solid #000000;
     overflow: hidden;
    text-overflow: ellipsis;
}
```

On hovering over element – It shows overflowed content
```
div.test:hover {
     text-overflow: inherit;
overflow: visible;
}
```

CSS3 Transforms

CSS3 transforms allow you to translate, rotate, scale, and skew elements. CSS3 supports 2D and 3D
transformations.

CSS3 2D Transforms

It has following 2D transformation methods:

transform: translate() transform: rotate() transform: scale() transform: skewX() transform:

skewY() transform: matrix()

Example-
div {transform: translate(50px,100px); }
div { transform: rotate(20deg); }

CSS3 Transforms

CSS3 3D Transforms

It has following 3D transformation methods:

transform: rotateX()   transform: rotateY()   transform: rotateZ()

Example-

div {

    transform: rotateX(150deg);

    }

CSS3 Transitions

It allows you to change property values smoothly (from one value to another),
over a given duration. 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.

Example-
div {
     width: 100px;
     height: 100px; background: red;

     -webkit-transition: width 2s; /* Safari */   transition:

     width 2s;
}

div:hover
{ width:
300px;

On hovering over

CSS3 Animation

CSS3 animations allows animation of most HTML elements without using JavaScript or Flash! An animation lets an element gradually change from one style to another. To use CSS3 animation, you must first specify some **@keyframes** for the animation.

Example-

/* The animation code */

@keyframes example {

    from {background-color: red;} to {background-color: yellow;}

}

/* The element to apply the animation to */

div {

    width: 100px; height: 100px;

    background-color: red;

    animation-name: example;

    animation-duration: 4s;

}