AKTU

B.Tech 5th Sem

CS IT & CS Allied

Web Technology

ONE SHOT Revision

(Crash Course)

Unit-2: CSS, CSS Advanced

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Course Details(Paid) : All Subjects

1	Detailed Video Lectures
2	Pdf Notes
3	One Shot Revision
4	Unit wise set of PYQs
5	Important Questions

Paid Courses are available in Gateway Classes Application
Link in Description

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-1

Today's Target

- Cascading Style Sheets (CSS)
  - Creating Style Sheets
    - Syntax
    - Types
      - Inline, Internal, and External
    - Cascading in CSS
  - AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Web Technology

AKTU Syllabus UNIT – II

Cascading Style Sheets (CSS)

CSS: Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties)

CSS Advanced Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector), CSS Color, Creating page Layout and Site Designs.

AKTU PYQs : UNIT-2

1. Describe the advantages of CSS. (AKTU 2021-22)
2. Create a CSS rule that makes all the text 2 times larger than the base font of the system. Mention can you integrate CSS on a web page. (AKTU 2021-22)
3. Define box model in CSS with block diagram. (AKTU 2019-20)
4. Explain CSS. What are the CSS frameworks? Explain in brief. What are the different ways of using the stylesheet? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. (AKTU 2019-20)
5. What do you mean by CSS? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. Mention how can you integrate CSS on a web page? (AKTU 2018-19)

Web Designing (AKTU 2023 – 24)

1. Describe the syntax of CSS.
2. Describe the different properties to set the background of a web page.
3. How will you apply CSS to text?

Web Designing (AKTU 2022 – 23)

1. Explain different types of selectors in CSS.
2. Discuss advantages of CSS.
3. Illustrate Box model in CSS with block diagram.
4. Explain the CSS and its basic properties. Outline a CSS rule that makes all the text 2.5 times larger than the base font of the system.
5. Explain cascading and the style precedence rules when using multiple approaches.

…AKTU PYQs UNIT-2

Web Designing (AKTU 2021 – 22)

1. What are Style Sheets?
2. Explain different types of selectors in CSS with example.
3. What are the techniques to use W3.CSS framework? Explain.
4. Explain three different ways to implement CSS on web page.
5. What is CSS? List out the various CSS properties. Explain the various concepts of CSS properties with neat example.

Web Designing (AKTU 2020 – 21)

1. How external style sheet is useful in web page design?
2. Discuss the box model in CSS with block diagram.
3. What is pseudo class in CSS?
4. Explain any eight CSS text properties in detail.
5. What is CSS? What are the different ways to create them? Explain with the help of an example.
6. Apply CSS to a web page with the following requirements:
   i) Add a background image of a submarine.
   ii) Set a color to the span elements (differnt color for each class).

Cascading Style Sheets (CSS)

"Cascading Style Sheets (CSS) are used to define the presentation (styling) of HTML elements on a web page."

- Describes the appearance, layout, and presentation of information on a web page.
- Describes how information is to be displayed, not what is being displayed.
- It separates presentation (styling) from content (HTML).
- It can be embedded in HTML document or placed into separate '.css' file.
- CSS may save a lot of work. It can control the layout of multiple web pages all at once.

Cascading Style Sheets (CSS)

CSS Syntax:

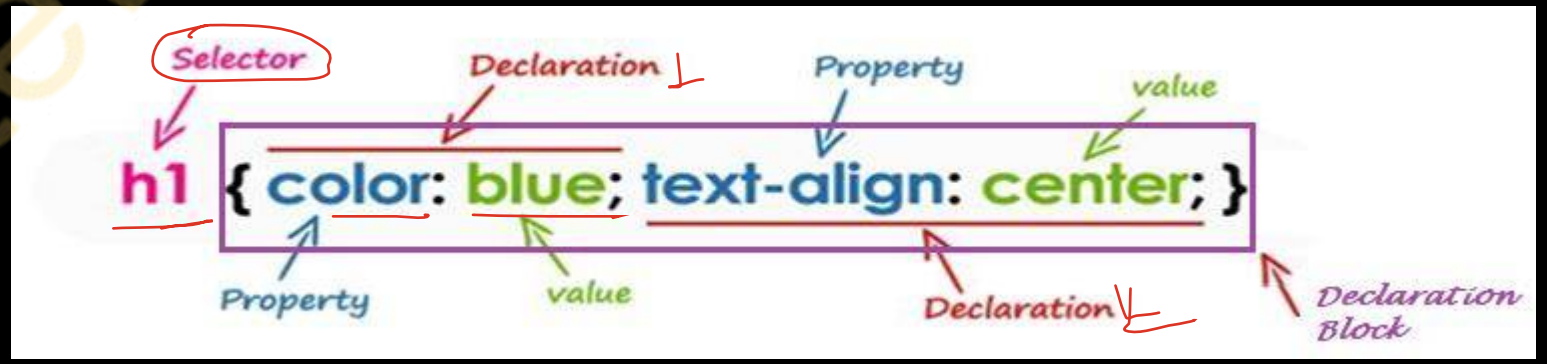
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

selector { property: value; property: value; }

Selector targets an HTML element, and properties (like `color` or `font-size`) define its appearance.

- Example:

```
h1 {
        color: blue;
        text-align: center;
}
```



Cascading Style Sheets (CSS)

Types of CSS

Three ways to use CSS on HTML document:

1. ***Inline,***
2. ***Internal, and***
3. ***External CSS.***

- Inline CSS: Inline CSS are directly applied on the HTML elements as they are used inside HTML tags using the `style` attribute.

  - Example:

```
<h1 style="color: red;">Hello World</h1>
```

  - ✓ It is the most prioritize CSS among these three.
  - ✓ This will override any external or internal CSS.
  - ✓ It is quick to write but harder to maintain.

Cascading Style Sheets (CSS)

Types of CSS

Three ways to use CSS on HTML document : Inline, Internal, and External CSS.

- Internal CSS: Internal CSS are placed in the `<style>` tag inside the `<head>` section of HTML. An internal style sheet may be used if one single HTML page has a unique style.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        h1 {
            color: red;
        }
    </style>
</head>
<body><h1>This is a heading</h1></body>
</html>
```

Cascading Style Sheets (CSS)

Types of CSS

(Three ways to use CSS on HTML document) : Inline, Internal, and External CSS.

- External CSS: External CSS are defined in a separate file that contains only CSS properties. This separate CSS file is then linked to the HTML document using <link> element:

```
<link rel="stylesheet" href="styles.css">
```

styles.css

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

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

- ✓ <link> element is placed inside the <head> section of an HTML page
- ✓ This is the recommended way to use CSS when you are working on projects. With an external style sheet, you can change the look of an entire website/ web application by changing just one file.

Cascading Style Sheets (CSS)

Types of CSS

(Three ways to use CSS on HTML document) : Inline, Internal, and External CSS.

Multiple conflicting styles for the same HTML element.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p { color: green; }  <!-- Internal CSS -->
  </style>
  <link rel="stylesheet" href="styles.css"> <!-- External CSS -->
</head>
<body>
  <p style="color: blue;">This is a paragraph.</p> <!-- Inline CSS -->
</body>
</html>
```

styles.css

```
p {
        color: red;
}
```

Cascading Style Sheets (CSS)

Cascading in CSS

Cascading refers to the priority order that browser follows when applying styles from different sources. It needs to determine which CSS rule is applied when there are multiple conflicting styles for the same HTML element.

Cascading Order

All the styles in a page will "cascade" as per the following order:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
   - Internal Stylesheet if internal style is defined after the link to the external style sheet.
   - External Stylesheet if the internal style is defined before the link to the external style sheet.
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSSDemo.html

```
<!DOCTYPE html>
<html>
<head>
	<title>Cascading Style Sheets (CSS)</title>

	<!-- EXTERNAL CSS -->
	<link rel="stylesheet" href="styles.css">

	<!-- INTERNAL CSS -->
	<style>
		h1 { text-align:center;}
		p {color:green}
	</style>

</head>

<body>
	<!-- INLINE CSS -->
	<h1 style="color:red">Cascading Style Sheets (CSS)</h1>
	<p style="color:blue"> CSS is used to style web pages. It controls how the
content will be displayed.</p>
</body>
</html>
```

styles.css

```
h1
{
	text-align:right;
}
p
{
	color:red;
}
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-2

Today's Target

- Cascading Style Sheets (CSS)
  - CSS Id and Class
  - CSS Properties
  - CSS Styling
    - Background
    - Text Format
    - Controlling Fonts
  - AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Id and Class:

A CSS selector selects the HTML element(s) to be styled.

'CSS Id' Selector

The id selector uses the *id attribute* of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element The hash (#) character followed by the id of the element is written as selector to select an element with a specific id.

Example:

```
#pid1 {
        color: gold;
        text-align: center;
}
```

Cascading Style Sheets (CSS)

CSS Id and Class:

A CSS selector selects the HTML element(s) to be styled.

'CSS Class' Selector

The class selector selects HTML elements with a specific value of the class attribute.

A period (.) character, followed by the class name is written to select elements with a specific class.

Example:

```
.right {
        text-align: right;
        color: gold;
}
```

We can also specify specific HTML elements to be affected by a class.

Example: p.class { … }

Cascading Style Sheets (CSS)

'CSS element' Selector:

The element selector selects HTML elements based on the element name.

Example:

```
h2 {
        text-align: center;
        color: gold;
}
```

The grouping selector selects all the HTML elements with the same style definitions.

Example:

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

Cascading Style Sheets (CSS)

Specificity and Cascading:

Cascading refers to how styles are applied in a cascading order, with more specific rules taking precedence over less specific ones.

- The cascade ensures that when there are conflicting CSS rules, the most specific rule (with the highest specificity) is applied.
- If specificity is the same, the order in which the rules appear determines the final style (i.e., the latter rule takes precedence).

Specificity Order:

1. ID Selectors have more specificity than class selector.
2. Class selectors are more specific than element selectors.
3. Element selectors have the lowest specificity.

Cascading Style Sheets (CSS)

CSS Properties

A CSS property styles an aspect of an HTML element. CSS properties allow web developers to control the appearance of elements on a web page. Example:

```
<div style="border: 1px solid black; font-size: 18px;">Style This</div>
```

A CSS property declaration consists of a property name followed by a colon, and then a property value.

Syntax

```
property-name: property-value;
```

When specifying multiple properties, each property-value pair is separated by a semicolon:

```
property1: value1;
property2: value2;
```

Although the last property doesn't require a semicolon, adding one helps when updating or adding new properties later.

Cascading Style Sheets (CSS)

CSS Styling (Background):

These are the set of properties that control the background appearance of elements, including color, images, position, and repetition. These properties help in enhancing the visual design and layout of a webpage by creating custom backgrounds.

Property	Description
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element using color values like red, #ff0000, or rgba(255,0,0,0.5).
background-image	Specifies the background image for an element, using a URL (url("image.jpg")) or values like none.
background-position	Sets the starting position of a background image, with values like top, center, bottom, or specific coordinates (e.g., 10px 20px).
background-repeat	Specifies how the background image repeats, with options like repeat, no-repeat, repeat-x, or repeat-y.

Cascading Style Sheets (CSS)

CSS Styling (Background):

Property	Description
background-size	Defines the size of the background image, with values such as auto, cover, contain, or specific dimensions (e.g., 100px 50%).
background	Sets all the background properties in one declaration. background-color background-image background-repeat background-position background-attachment

Example

```
div {
        background-color: #f0f0f0;
        background-image: url('background.jpg');
        background-repeat: no-repeat;
        background-position: center top;
        background-attachment: fixed;
        background-size: cover;

    }
```

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

The set of CSS properties used to control the appearance and formatting of text, such as color, alignment, decoration, and spacing.

Text Color

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

Example:

```
h1{
        color: gold; / color: rgb(255,0,0) / color: #ffff00
}
```

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Alignment and Text Direction

Text Alignment is to specify how text is horizontally/ vertically positioned within an element
Text Direction specifies the direction in which text is displayed.

Property	Description
text-align	Specifies the horizontal alignment of text
vertical-align	Sets the vertical alignment of an element
direction	Specifies the text direction/writing direction

Example:

```
h3 {
        text-align: right;/   text-align: left;/   text-align: center;/   text-align: justify;
        direction: rtl;
        vertical-align: baseline;/  vertical-align: text-top;/  vertical-align: text-bottom;/
        vertical-align: sub;/  vertical-align: super;
}
```

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Decoration:

A CSS property used to apply visual effects to text, such as underlining, overlining, strikethrough etc.

Property	Description
text-decoration-line	Specifies the kind of text decoration to be used
text-decoration-color	Specifies the color of the text-decoration
text-decoration-style	Specifies the style of the text decoration
text-decoration-thickness	Specifies the thickness of the text decoration line
text-decoration	Sets all the text-decoration properties in one declaration

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Decoration:

Example:

```
h3 {
        text-decoration-line: overline;/    text-decoration-line: line-through;/
        text-decoration-line: underline;/    text-decoration-line: overline underline;

        text-decoration-color: red;

        text-decoration-style: solid;/    text-decoration-style: double;/
        text-decoration-style: dotted;/    text-decoration-style: dashed;/
        text-decoration-style: wavy;

        text-decoration-thickness: 5px;

        text-decoration: underline red double 5px;
    }
```

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Transformation:

It is used to turn the text into uppercase or lowercase letters, or capitalize the first letter of each word:

Example

```
p {
        text-transform: uppercase;/
        text-transform: lowercase;/
        text-transform: capitalize;
}
```

AB
ab
Abc

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Spacing:

A CSS property that controls the spacing between characters, words, or lines of text, enhancing readability and layout.

Property	Description
text-indent	Specifies the indentation of the first line in a text-block
letter-spacing	Specifies the space between characters in a text
line-height	Specifies the line height
word-spacing	Specifies the space between words in a text
white-space	Specifies how to handle white-space inside an element

Cascading Style Sheets (CSS)

CSS Styling (Text Format):

Text Spacing:

A CSS property that controls the spacing between characters, words, or lines of text, enhancing readability and layout.

Example

```
p {
        text-indent: 50px;
        letter-spacing: 5px;
        line-height: 0.8;
        word-spacing: 10px;
        white-space: nowrap;
}
```

Cascading Style Sheets (CSS)

CSS Styling (Controlling Fonts):

Choosing the right font is key to how readers experience a website. It can improve readability. Font also play important role in enhancing the text's clarity and impact.

Property	Description
font-family	Specifies the font family (typeface) for text. Use multiple fonts as fallbacks
font-size	Specifies the font size of text. Can be specified in px, em (relative to parent), rem (relative to root (html), %, etc.
font-style	Sets the style of the font (normal, italic, oblique).
font-variant	Displays text in small-caps or normal style.
font-weight	Specifies the thickness (bold/ normal/ numeric value) of a font
font	Sets all the font properties in one declaration (font-style font-variant font-weight font-size/line-height font-family)

Cascading Style Sheets (CSS)

CSS Styling (Controlling Fonts):

Example

```
p {
        font-family: "Arial", sans-serif;
        font-size: 18px;
        font-style: italic;
        font-variant: small-caps;
        font-weight: bold;
        font: italic small-caps bold 18px/1.5 "Arial", sans-serif;
}
```

SelectorsDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Selectors Demo</title>
    <style>
        #heading {
            color:gold;
        }

        .headclass {
            color:green;
        }

        h1 {
            color:aqua;
        }

        p.para {
            color:blue;
        }

        h3, h4 {
            color:red;
        }

    </style>
</head>
<body>
    <h1>Selectors Demo</h1>
    <h1 class="para">Id Selector</h1>
    <h3>Let us look at element selectors now.</h3>
    <h4>Group Selectors</h4>
    <p class="para">I am demonstrating Id selector.</p>

</body>
</html>
```

BackgroundDemo.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>
```

```
            Background Demo
        </title>
        <style>
            .backgroudcls {
                background-color: gray;
                background-image: url('cssimage.png');
                background-repeat: no-repeat;
                background-position: right bottom;
                background-size: 100% 100%;
                background-attachment: scroll;

                /* background: aqua url('cssimage.png') no-repeat center center
fixed; */

                width: 500px;
                height: 500px;
                border: 1px solid black;

            }

        </style>
    </head>
    <body>
        <div class="backgroudcls">

        </div>
        <p>Background Demo</p>
    </body>
</html>
```

TextDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Text Formatting Demo</title>
    <style>
        .textcolor {
            color: red;
        }

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

```
        .textverticalalign {
            vertical-align: super;
        }

        .direction {
            direction: rtl;
        }

        .decoration {
            text-decoration-line:overline underline;
            text-decoration-color:red;
            text-decoration-style: solid;
            text-decoration-thickness: 5px;
            /* text-decoration: underline brown wavy 2px;*/
        }

        .tranform {
            text-transform: capitalize;
        }

        .spacing {
            text-indent: 50px;
            letter-spacing: 5px;
            line-height: 50px;
            word-spacing: 7px;
            white-space: wrap;
        }

        .fontcontrol {
            font-family: 'Times New Roman', Times, serif;
            font-size: 20px;
            font-style: italic;
            font-weight: bold;
            font-variant: small-caps;
            /* font: italic small-caps bold 10px/50px 'Times New Roman'; */
        }
    </style>

</head>
<body>
    <p class="textcolor textalign">I am demonstrating text formatting
properties.</p>
    <p class="textalign textcolor">X<span class="textverticalalign">2</span></p>
    <p class="direction">Demo of direction.</p>
```

```
    <p class="decoration">Hyper Text Markup Language</p>

    <p class="tranform">hyper text</p>

    <p class="spacing">We            are now demonstrating text spacing
properties of CSS. We have text-indent, letter-spacing, line-height etc.</p>


    <p class="fontcontrol">Font Demo</p>

    <p style="font-style: oblique;">Oblique Text</p>
    <p style="font-style: italic;">Oblique Text</p>

</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-3

Today's Target

- Cascading Style Sheets (CSS)
  - Working with Lists
  - Working with Tables
  - AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

Working with Lists

In HTML, there are three types of lists:

Ordered List (<ol>): This type of list displays items with sequence like (1, 2, 3). It is typically used when the order of the items is important.

Unordered List (<ul>): This list uses bullets, such as circles or squares, to display items. It is used when the order of the items does not matter.

Description List (<dl>): A description list is used to display a list of terms and their descriptions.

Styling Lists with CSS: *Styling lists with CSS makes them more visually engaging and helps improve the user experience on a web page. CSS can be used to customize how lists look by changing list markers, adding images as bullets, or altering the position of markers.*

Cascading Style Sheets (CSS)

Working with Lists

CSS Properties for Lists: CSS offers several properties for styling lists, allowing customization of both ordered (numbers) and unordered (bullets) lists. Here are the key properties for styling lists:

1. list-style-type

This property defines the type of marker (bullet, number, etc.) used for list items.

For Unordered Lists: disc (default), circle, square, none

For Ordered Lists: decimal, lower-alpha, upper-alpha, lower-roman, upper-roman, etc.

Examples:

```
ul {
        list-style-type: square;
}
ol {
        list-style-type: lower-alpha;
}
```

Cascading Style Sheets (CSS)

Working with Lists

2. list-style-position

This property controls whether the marker is inside or outside the content box.

Values:

inside (marker aligned with text)

outside (default, marker outside text area).

Example:

```
ul {
        list-style-position: inside;
}
```

Cascading Style Sheets (CSS)

Working with Lists

3. list-style-image

This property allows you to set a custom image as the list marker.

Value: url(image-path) or none

Example:

```
ul {
        list-style-image: url('bullet.png');
}
```

4. list-style (Shorthand)

You can combine the above three properties into a single declaration using the list-style shorthand.

Example:

```
ul {
        list-style: square inside url('bullet.png');
}
```

Cascading Style Sheets (CSS)

Working with Tables

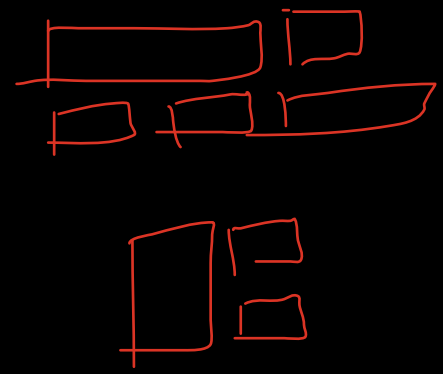
HTML tables are used to display tabular data, where content is arranged in rows and columns.

Basic Tags for HTML Table:

<table>:	Defines the table.
<tr>:	Defines a table row.
<th>:	Defines a header cell (bold, center-aligned by default).
<td>:	Defines a standard data cell (normal text, left-aligned by default).

Additional Tags for HTML Table:

<caption>:	Provides a title or description for the table.
<thead>:	Groups the header content in a table.
<tbody>:	Groups the body content (main data) in a table.
<tfoot>:	Groups the footer content in a table.



Key Attributes:

border:	Adds a border to the table and its cells.
colspan / rowspan:	Allows a cell to span across multiple columns or rows.

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

CSS provides a variety of properties to style HTML tables, allowing developers to control the layout, appearance, and behavior of table elements such as rows, columns, cells, and borders.

Below are the key CSS properties used for styling tables:

1. Border

   This property defines the appearance of the borders around the table, rows, and cells.

   Values: border-width, border-style, border-color

   Example:

```
table, th, td {
        border: 1px solid black;
}
```

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

2. border-collapse

This property controls whether the table borders are collapsed into a single border or separated.

Values: collapse (single border), separate (default)

Example:

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

3. Padding

This property controls the space between the content of the cell and its border.

Values: Length values like px, em, etc.

Example:

```
td {
        padding: 10px;
}
```

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

4. text-align

This property defines the horizontal alignment of text inside table cells.

Values: left, center, right

Example:

```
th, td {
        text-align: left;
}
```

5. vertical-align

This property controls the vertical alignment of content within table cells.

Values: top, middle, bottom

Example:

```
td {
        vertical-align: middle;
}
```

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

6. caption-side

This property defines the position of the table caption.
Values: top (default), bottom
Example:

```
caption {
        caption-side: bottom;
}
```

7. empty-cells

This property defines whether or not borders should be displayed around empty cells. *Values:* show (default), hide
Example:

```
table {
        empty-cells: hide;
}
```

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

8. border-spacing

This property in CSS defines the space between the borders of adjacent table cells. It applies only when border-collapse is set to separate (the default value for tables).

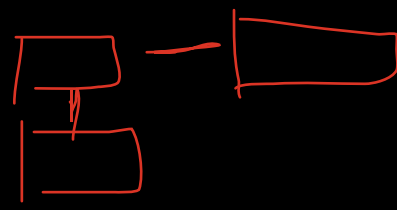
Values:

Single value: The same spacing is applied both horizontally and vertically.

Two values: The first value sets the horizontal spacing, and the second sets the vertical spacing.

Example:

```
table {    border-collapse: separate;   border-spacing: 15px 10px; }
```



9. width and height

These properties control the dimensions of the table, columns, or cells.

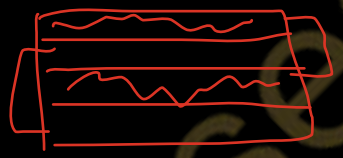
Values: Length values in px, %, etc.

Example:

```
table {
        width: 100px;      height: 100px;
}
```

Cascading Style Sheets (CSS)

Working with Tables



CSS Table Properties

Zebra-Striped Tables with CSS

Zebra-striped tables improve readability by alternating row colors.
This is achieved using the background-color property along with nth-child() selectors to target even and odd rows.

Example:

```
tr:nth-child(even) {  background-color: #f2f2f2; /* Light gray for even rows */}
tr:nth-child(odd) {  background-color: #ffffff; /* White for odd rows */}
```

nth-child(even): Selects all even-numbered rows.
nth-child(odd): Selects all odd-numbered rows.

This alternating pattern enhances visual clarity, making it easier to scan rows of data in the table.

Cascading Style Sheets (CSS)

Working with Tables

CSS Table Properties

Hoverable Tables with CSS

Hoverable tables enhance user interaction by highlighting rows when the user hovers over them.

This effect is created using the :hover selector on <tr> in combination with the background-color property.

Example:

```
tr:hover {  background-color: #f5f5f5; /* Light gray background on hover */}
```

- :hover is applied when the mouse pointer is over a table row (<tr>).
- Adds visual feedback to improve user experience, especially for tables with many rows.
- By applying this simple rule, rows will change color as the user hovers, improving navigation and readability.

CSS_List_Demo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS List Demo</title>
    <style>
        ul {
            /* list-style-type: none;
            list-style-position: inside;
            list-style-image: url('tick_bullet.png'); */

            list-style: none outside url('tick_bullet.png');

        }
        ol {
            list-style-type: lower-greek;
            list-style-position: inside;
        }
        li {
            border: 1px dashed black;
        }
    </style>

</head>
<body>
    <ul>
        <li>HTML</li>
        <li>CSS</li>
    </ul>

    <ol>
        <li>HTML</li>
        <li>XML</li>
        <li>CSS</li>
    </ol>
</body>
</html>
```

CSS_Tables_Demo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Tables Demo</title>
```

```
    <style>
        table {
            border: 1px solid black;
            border-collapse: separate;
            border-spacing: 10px 10px;
            caption-side: top;
            empty-cells: show;
        }
        /* tr:nth-child(even){
            background-color: aliceblue;
        }
        tr:nth-child(odd) {
            background-color: lightgray;
        } */
        tr:hover {
            background-color: aqua;
        }
        td, th {
            border: 1px solid black;
            text-align:center;
            vertical-align: middle;

            padding-left: 10px;
            padding-top: 10px;
            padding-bottom: 10px;
            padding-right: 10px;

            /* width:100px;
            height: 100px; */
        }

    </style>

</head>
<body>
    <table>
        <caption>Students List</caption>
        <tr>
            <th>Roll NO.</th>
            <th>Name</th>
        </tr>
        <tr>
            <td>1</td>
            <td></td>
        </tr>
```

```
        <tr>
            <td>2</td>
            <td style="background-color: lightgray;">Akshay Tomar</td>
        </tr>
        <tr>
            <td>3</td>
            <td>Subhash Tomar</td>
        </tr>
    </table>

</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-4

Today's Target

- Cascading Style Sheets (CSS)
- Working with Block elements and objects
  - Box Model
    - Introduction,
    - Border properties,
    - Padding Properties,
    - Margin properties
- AKTU PYQs

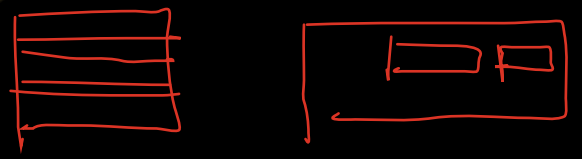
By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

Working with block elements and objects

It involves understanding how block-level elements behave in web design. Block elements take up the full width of their container and can be styled to control layout, spacing, and content organization, making them essential for creating structured and visually appealing web pages.



Block Elements

Block elements are HTML elements that take up the full width available and start on a new line. Examples are , <p>, <h1> to <h6>, <form> etc.

They can contain other block or inline elements.

Example:

```
<div style="width: 500px; height: 500px;">This is a block element. <p style="background-color: gray;">This is another block element.</p></div>
```

Inline elements

Only take up as much width as necessary, do not start on a new line.

Example:

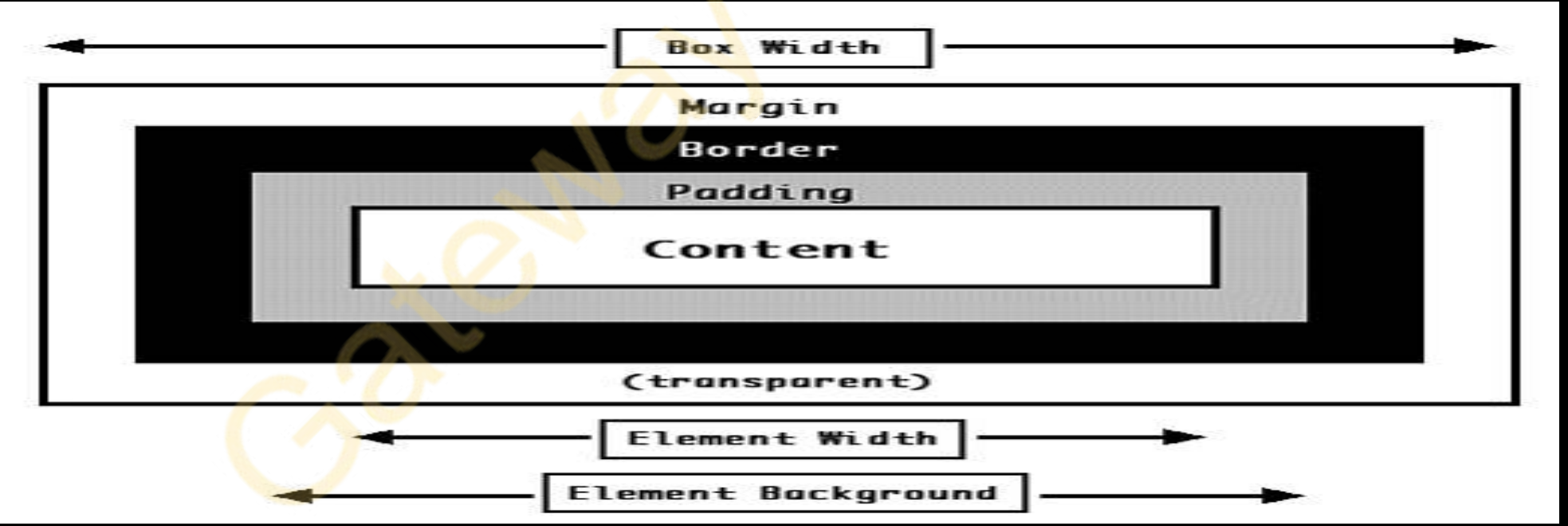
```
<p>The<span style="text-decoration-line: underline;">HTML</span>language.</p>
```

Cascading Style Sheets (CSS)

Box Model

The CSS Box Model is a fundamental concept that defines how elements are structured and spaced on a web page. Each element is represented as a rectangular box, which consists of four areas: content, padding, border, and margin. Understanding the box model helps in controlling layout, spacing, and the overall appearance of elements in web design.

The image below illustrates the box model:



Cascading Style Sheets (CSS)

Box Model

- Content: This includes the actual content of the element, such as text, an image, and so on.
- Padding: This includes any padding that has been set around the content by the padding-top, padding-bottom, padding-left, padding-right, or padding properties.
- Border: This includes any border that has been set around the element's content and padding. Borders are set by the border-top, border-bottom, border-left, border-right, or border properties.
- Margin: This includes any margin that has been set around the element's content, padding, and border. Margins are set by the margin-top, margin-bottom, margin-left, margin-right, and margin properties.

Cascading Style Sheets (CSS)

Padding Properties

Padding is the space between the content and the border.

It is transparent and can be applied individually or using shorthand.

Key Properties:

padding-top, padding-right, padding-bottom, padding-left:

Sets the padding for each side.

Shorthand: padding

(e.g., padding: 10px 20px 30px 40px;/ padding: 10px 15px;)

Example:

```
div {  padding: 10px 15px; /* Top/Bottom = 10px, Left/Right = 15px */}
```

1 TBLR

2 TB LR

4 TBLR

Cascading Style Sheets (CSS)

Border Properties

The border surrounds the padding. You can customize its width, style, and color.

It is transparent and can be applied individually or using shorthand.

Key Properties:

border-width, border-style, border-color: Sets individual border properties.

Shorthand: border (e.g., border: 2px solid blue;)

We can set individual borders for each side of an element using the following properties: border-top, border-right, border-bottom, border-left

Example:

```
div {  border: 2px solid red; /* Width, Style, and Color in one */}
```

Cascading Style Sheets (CSS)

Margin Properties

The margin is the outermost layer of the box model, which creates space between elements.

It is transparent and can be applied individually or using shorthand.

Key Properties:

margin-top, margin-bottom, margin-left, margin-right: Sets margins for each side.

Shorthand: margin (e.g., margin: 20px 10px;)

Example:

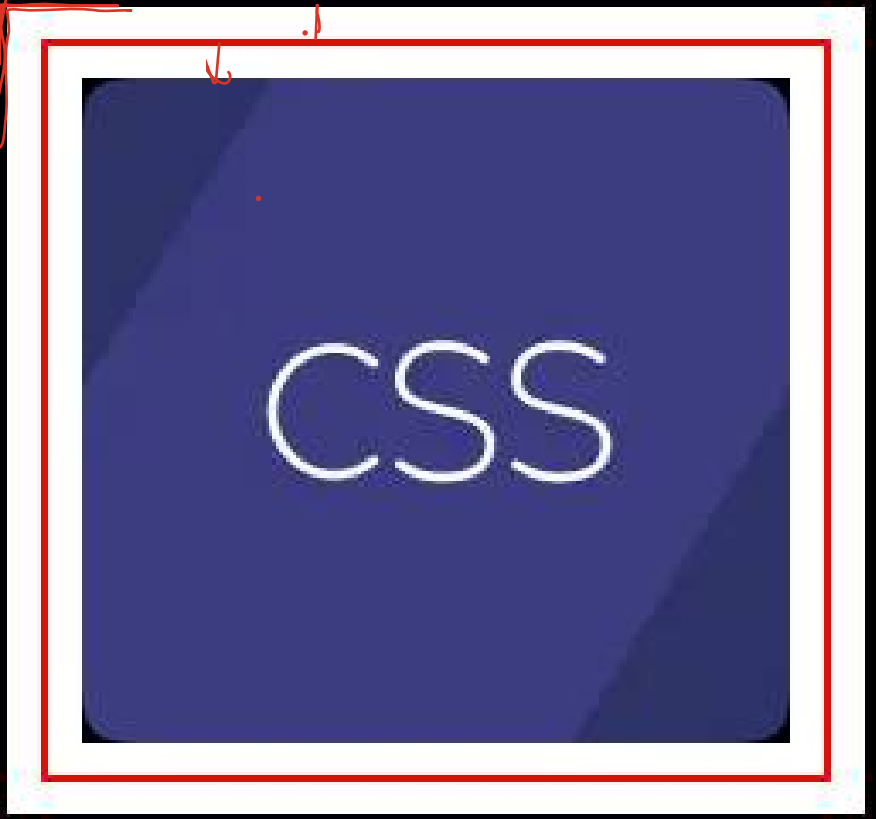
```
div {
        margin: 10px 5px; /* Top/Bottom = 10px, Left/Right = 5px */
}
```

Cascading Style Sheets (CSS)

Setting Padding, Borders, and Margins for Images and Other Objects

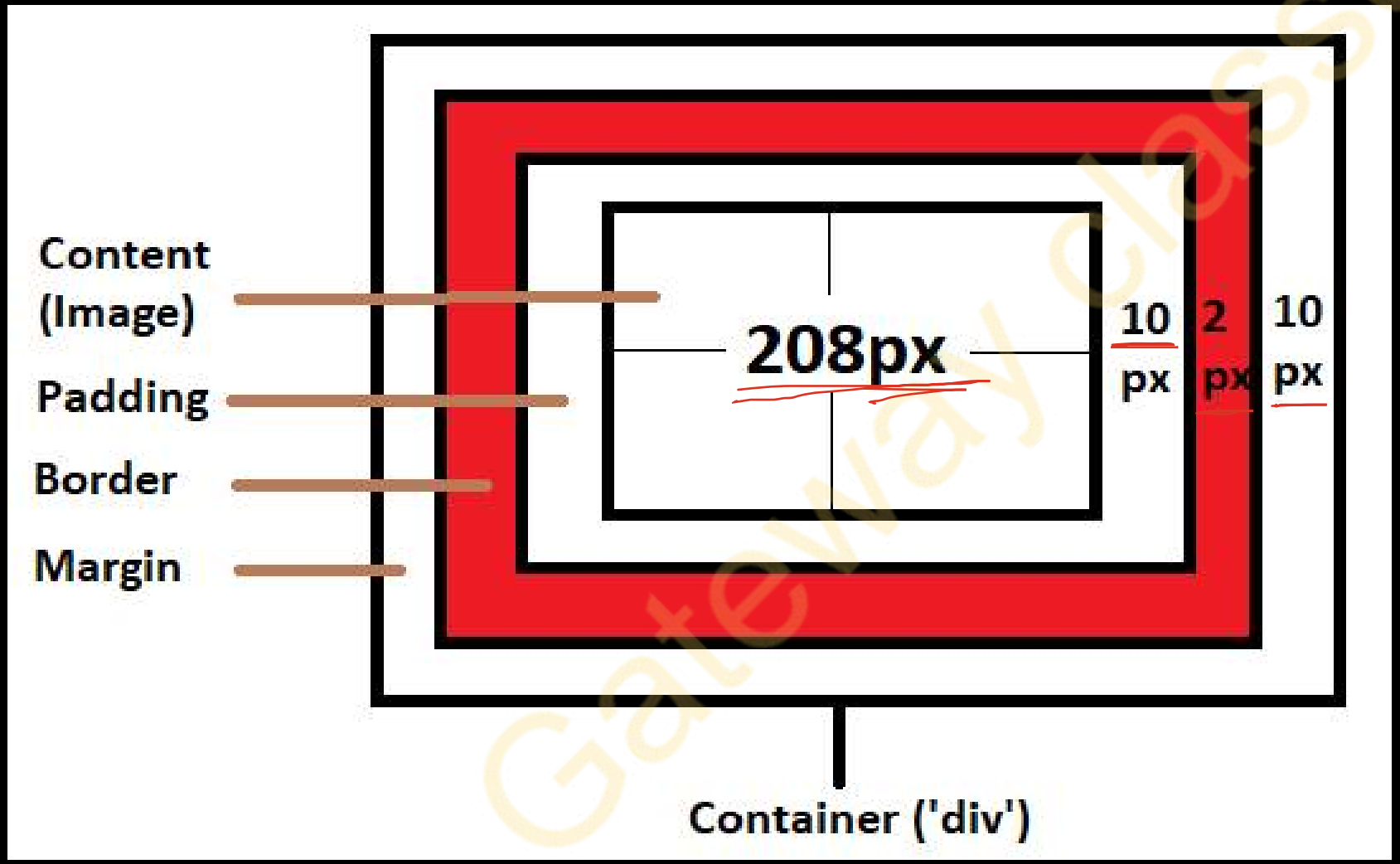
```
<!DOCTYPE html>
<html>
<head>
   <title>Box Model Demo</title>
   <style>
      div {  width: 252px;    height: 252px;    border: 2px solid black;  }
      img {  border: 2px solid red;  padding: 10px;  margin:10px;   }
   </style>
</head>
<body>
   <div>
      <img src="imgcss.png">
   </div>
</body>
</html>
```

OUTPUT



Cascading Style Sheets (CSS)

Setting Padding, Borders, and Margins for Images and Other Objects



Width of Container ('div') =

208px (width of content area)
+ 20px (left padding + right padding)
+ 4px (left border + right border)
+ 20px (left margin + right margin)
= **252px**

Height of Container ('div') =

208px (height of content area)
+ 20px (top padding + bottom padding)
+ 4px (top border + bottom border)
+ 20px (top margin + bottom margin)
= **252px**

Cascading Style Sheets (CSS)

Setting Padding, Borders, and Margins for Images and Other Objects

The <object> tag in HTML allows embedding external resources, like PDF files, multimedia, or other content into the web page.
Just like images, you can style the <object> tag using CSS to set padding, borders, and margins, controlling its spacing and visual appearance on the page.

Example

CSS

```
object {
                    border: 3px solid green;      padding: 15px;      margin: 20px;
    }
```

HTML

```
<object data="WT U-2 Lec3 Code.pdf" type="application/pdf" width="600" height="400">
    <p>Your browser does not support PDFs. <a href="WT U-2 Lec3 Code.pdf">Download the PDF</a>.</p>
</object>
```

BlockElementsDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Box Model Demo</title>
    <style>
        div {
            width:500px;
            height: 500px;

            border: 2px solid black;
        }

        p {
            background-color: gray;
        }
    </style>
</head>
<body>
    <div>
        <p>The
            <span style="text-decoration-line:
underline;">
                HTML
            </span>
            Language.</p>
        <p>Cascading Style Sheets</p>
    </div>
    </body>
</html>
```

BoxModelDemonstration.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Box Model Properties
Demonstration</title>
    <style>
        div {
            width: 252px;
            height: 252px;

            border: 2px solid black;
        }

        img {

            /* padding: 10px; */

            /* padding-top:10px;
            padding-bottom: 10px;
            padding-left: 10px;
            padding-right: 10px; */

            padding: 10px 10px;

            margin: 10px 10px 10px 10px;

            border: 2px solid red;

            /* border-top: 5px dashed blue;
```

```
            border-bottom: 10px dotted green;
            border-left: 15px double red;
            border-right: 20px solid brown; */
        }

        object {
            width: 600px;
            height: 600px;

            border: 5px red solid;
        }
    </style>
</head>
<body>
    <!-- <div>
        <img src="imgcss.png">
    </div>  -->

    <object data="WT U-2 Lec3 Code.pdf"
type="application/pdf">
        <p>This is not supported!</p>
    </object>
</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-5

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Grouping
  - Dimension
  - Display
  - Navigation Bar
- AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Grouping'

CSS Grouping allows you to apply the same styles to multiple selectors, simplifying your code and making it more efficient.

Grouping Selectors: By separating selectors with commas, you can apply the same style to multiple elements.

Example:

```
h1, h2, h3 {    color: blue; text-decoration-line: underline    }
```

In this example, all <h1>, <h2>, and <h3> elements will have the same text color and font style.

Benefits:

1. Grouping reduces code duplication.
2. Grouping makes styles easier to maintain and read.

Cascading Style Sheets (CSS)

CSS Advanced – 'Grouping'

Example: Complex Grouping in CSS

In this example, we apply the same styles to *different types of selectors*: *element* (p), *class* (.highlight), *ID* (#main-content), and *pseudo-class* (a:hover).

```
p, .highlight, #main-content, a:hover
{   background-color: #f0f8ff;   color: #333;   border: 1px solid red;   padding: 10px;   }
```

p:	Targets all <p> elements.
.highlight:	Targets all elements with the class highlight.
#main-content:	Targets the element with the ID main-content
a:hover:	Targets links (<a>) when hovered over by the mouse.

Cascading Style Sheets (CSS)

CSS Advanced – 'Dimension'

CSS Dimension properties control the size and spacing of elements on a page.

The width and height properties are essential for controlling how much space an element occupies.

- width: Specifies the width of an element's content area.
- height: Specifies the height of an element's content area.

Example: div { width: 300px; height: 200px; }

- min-width / min-height: Ensures the element does not shrink below a certain size.

Example: div { min-width: 150px; }

- max-width / max-height: Prevents the element from expanding beyond a certain size.

Example: div { max-width: 600px; }

Cascading Style Sheets (CSS)

CSS Advanced – 'Display'

The display property controls the layout behavior of an element on the page and specifies how an element is shown on a web page.

Default 'Display': Every HTML element has a default display value, which is generally either block or inline, depending on the element type.

Common display values:

- block: The element takes up the full width available and starts on a new line.
- inline: The element only takes up as much width as necessary and does not start on a new line.
- inline-block: Combines the characteristics of inline and block—allows the element to be inline but still have block-level dimensions (width and height).

Cascading Style Sheets (CSS)

CSS Advanced – 'Display'

Overriding the Default 'Display': You can override the default display value using CSS to change how elements behave on the page. This allows for more flexible layouts, enabling you to switch between block, inline, or other display types.

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way.

Example: Making <li> elements inline for horizontal menus:

```
li {
        display: inline;
}
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Display'

Hiding an Element

display: none

Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:

Example

```
.hidden {    display: none;    }
```

visibility: hidden

It also hides an element. However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

Example

```
.hidden {    visibility: hidden;    }
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Navigation Bar'

A navigation bar is a list of links typically used to navigate between different sections of a website.

It can be horizontal or vertical, and is often styled for better user interaction and design consistency.

Creating a Navigation Bar:

- Navbars are usually created using unordered lists (<ul>), with each link wrapped inside a list item (<li>).
- CSS is used to style the navbar, control its layout and appearance.

Cascading Style Sheets (CSS)

CSS Advanced – 'Navigation Bar'

Example:

```
<style>

    ul.navbar {    list-style-type: none;    margin: 0;    padding: 0;
                   background-color: #333;  }

    ul.navbar li {    display: inline;    } /* Horizontal Navigation bar. */

    ul.navbar li a {    display: inline-block;    color: white;    text-align: center;
                        padding: 14px 20px;    text-decoration: none;    }

    ul.navbar li a:hover {    background-color: #111;    }

</style>
```

CSS_Grouping_Demo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Advanced - Grouping - Demo</title>
    <style>
        p, .highlight, #pid, a:hover {
            background-color: pink;
            color: red;
            padding: 10px;
            border: 2px solid black;
        }
    </style>
</head>
<body>
    <h1 class="highlight">CSS Advanced</h1>
    <h2 id="pid">Grouping</h2>
    <p>I am demonstrating grouping of selectors
in CSS advanced.</p>

    <a href="www.google.com">Google</a>
</body>
</html>
```

CSS - Display.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS - Advanced - Display -
Demo</title>
```

```
    <style>
        p {
            background-color: pink;
        }
        .inlineelements {
            display: inline;
        }

        .blockelements {
            display: block;
        }

        .inlineblock {
            display: inline-block;
            width: 100px;
            height: 100px;
        }

        .hidden {
            display: none;
            /* visibility: hidden; */
        }
    </style>
</head>
<body>
    <p class="inlineelements">Hyper Text Markup
Language</p>
    <p class="inlineelements">I am demonstrating
display property under CSS Advanced.</p>
```

```
    <div class = "hidden" style="border: 2px
solid black; width:200px; height: 200px;">
        This element will be hidden.
    </div>


    <p>eXtensible Markup Language
        <span class="blockelements" style="text-
decoration-line: underline; background-color:
aqua;">(XML)</span></p>


    <p>Java Script is a <span class="inlineblock"
style="background-color: gray;">very
popular</span>scripting language.</p>
</body>
</html>
```

NavBarDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Navigation Bar Demo</title>

    <style>
        ul.navbar {
            list-style-type: none;
            margin:0px;
            padding: 0px;
            background-color: beige;
        }
```

```
        ul.navbar li {
            display: block;
        }

        ul.navbar li a {
            display: inline-block;
            background-color: gray;
            padding:20px;
            width: 50px;
            height: 50px;
            text-align: center;
            text-decoration: none;
        }

        ul.navbar li a:hover {
            background-color: orange;
        }

    </style>

</head>
<body>
    <ul class="navbar">
        <li><a href="#">HTML</a></li>
        <li><a href="#">XML</a></li>
        <li><a href="#">CSS</a></li>
        <li><a href="#">Java Script</a></li>
    </ul>
</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-6

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Pseudo Classes
    - Hover
    - Active
    - Focus
    - Link, Visited
    - First-child, Nth-child, Last-child
- AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Pseudo Class'

Pseudo-classes in CSS are used to select and style elements based on their state or position within the document tree.

Syntax:

```
selector : pseudo-class {
    property: value;
}
```

Example: `a:hover { color: red; }`

Additionally, a functional pseudo-class has a pair of parentheses to define the arguments.

Example: :nth-child(2)

Cascading Style Sheets (CSS)

CSS Advanced – 'Pseudo Classes'

Hover:

The pseudo-class :hover is used to specify mouse hovering state of an element. This used to style element while users mouse passes through an element in document.

Example:

```
p:hover {
    color: red;
    border: 2px solid black;
}
```

Active:

The pseudo-class :active will apply style to an element when user activates the element by clicking or tapping on it.

Example:

```
li: active {
    background-color: red;
}
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Pseudo Classes'

Focus:

The pseudo-class :focus pseudo-class applies a style to an element when it receives focus, such as when a user clicks on an input field.

Before entering text into an input element, the user must click inside the input area to activate the cursor, which is referred to as focusing on the element.

Example:

```
input: focus {        background-color: gray;
                      border: 2px dashed green;      }
```

Link:

The pseudo class :link represents a link that has not yet been visited.

Visited:

The pseudo class :visited applies once the link has been visited.

}

Cascading Style Sheets (CSS)

CSS Advanced – 'Pseudo Classes'

Example:

```
a:link      {
                color: blue;
                text-decoration: none;
            }
a:visited   {
                color: green;
            }

a:hover     {
                color: red;
                text-decoration: underline;
            }

a:active    {
                color: orange;
            }
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Pseudo Classes'

First-child:

The :first-child pseudo-class matches a specified element that is the first child of the parent element.

Nth-child:

The :nth-child pseudo-class matches an element that is at the specified position (or number) among its siblings within the same parent element.

Last-child:

The :last-child pseudo-class matches a specified element that is the last child of the parent element.

Examples:

```
p:first-child    {    font-style: italic;          color: red;       }
p:nth-child(2) {    text-decoration-line: underline;          font-weight: bold;       }
p:last-child     {    font-style: italic;          color: blue;       }
```

PseudoClassesDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Pseudo Classes Demo</title>

    <style>
        p:hover {
            color: red;
            border: 2px dashed black;
        }

        li:active {
            background-color: pink;
        }

        input:focus {
            background-color: gray;
        }

        a:link {
            color:blue;
            text-decoration: none;
        }

        a:visited {
            color:green;
        }

        a:hover {
```

```
            color:red;
            text-decoration-line: underline;
        }

        a:active {
            color:brown;
            background-color:pink;
        }

        p:first-child {
            font-style:italic;
            color:blue;
        }

        p:last-child {
            font-style:italic;
            color: red;
        }

        p:nth-child(2) {
            font-weight: bold;
            background-color: pink;
        }
    </style>
</head>
<body>
    <p>Pseudo Classes Demo</p>

    <ul>
        <li>HTML</li>
        <li>XML</li>
```

```
        <li>CSS</li>
    </ul>

    <input type="text">
    <br>
    <a href="https:\\www.google.co.in">Google</a>

    <div>
        <p>Hyper Text Markup Language (HTML)</p>
        <p>eXtensible Markup Language (XML)</p>
        <p>Cascading Style Sheets (CSS)</p>
    </div>
</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-7

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Attribute Selectors
    - [attribute]
    - [attribute="value"]
    - [attribute*="value"]
    - [attribute^="value"]
    - [attribute$="value"]
    - [attribute~="value"]
    - [attribute|="value"]
    - Multiple Attribute Selectors
  - AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

Attribute Selector in CSS is used to select HTML elements with specific attribute or attribute value.

- Attribute selectors are enclosed in square brackets [ ].

Example:

```
input[type] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all input elements with type attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute] Selector

This selector selects elements that have specified attribute, regardless of its value or type of element.

Example:

```
[placeholder] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute="value"] Selector

This selector selects elements that have a specific attribute with a specific value.

Example:

```
[placeholder="Enter text here ..."] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value "Enter text here ..." of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute="value"] Selector*

This selector selects elements that have a specific attribute with a value containing a specific substring.

Example:

```
[placeholder*="text"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value containing substring "text" in the value of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute^="value"] Selector

This selector selects elements that have a specific attribute with a value that starts with a specific string.

Example:

```
[placeholder^="Enter"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value starting with the string "Enter" in the value of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute$="value"] Selector

This selector selects elements that have a specific attribute with a value that ends with a specific string.

Example:

```
[placeholder$="…"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value ending with the string "…" in the value of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute~="value"] Selector

This selector is used to select elements that have a specific attribute with a value containing a specified word. The word should be a standalone word, surrounded by spaces or at the beginning or end of the attribute value.

Example:

```
[placeholder~="here"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value containing the word "here" in the value of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

[attribute|="value"] Selector

The [attribute|=value] selector matches an attribute value that either exactly equals the specified value or begins with the specified value followed by a hyphen (-).

Example:

```
[placeholder|="text"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements with the value "text" or starting with "text-" as the value of the placeholder attribute.

Cascading Style Sheets (CSS)

CSS Advanced – 'Attribute Selector'

Multiple Attribute Selectors

CSS multiple attribute selectors allow you to select elements by combining several attribute conditions. Each attribute selector must be satisfied for the styles to be applied to the element.

Example:

```
[type][placeholder*="text"] {
    border: 2px solid red;
}
```

A border with the specified style will be applied to all elements having type attribute and value having substring "text" in the value of the placeholder attribute.

AttributeSelectorsDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Attribute Selectors</title>
    <style>
        [placeholder] {
            border: 1px solid red;
        }

        [placeholder="Enter text here ..."] {
            font-weight:bold;
            color: blue;
        }

        [placeholder*="xt h"] {
            color:blueviolet;
        }

        [placeholder^="Enter"] {
            text-decoration-line:line-through;
        }

        [placeholder$="e .."] {
            font-style:italic;
        }

        [placeholder~="name"]
        {
            border: 5px dashed brown;
```

```
        }

        [placeholder|="Enter"]
        {
            border:10px solid pink;
        }

        [type="password"][placeholder^="Enter "]
{
            color:green;

        }

        div[data-category|="webtech"]{
            border: 2px solid green;
        }
    </style>
</head>
<body>
    <form>
        <input type="text" placeholder="Enter
text here ..">
        <input type="text" placeholder="Enter ">

        <input type="password" placeholder="Enter
password here...">


        <div data-category="webtech-
CSS">CSS</div>
```

```
        <div data-category="webtech-
HTML">HTML</div>
        <div data-category="dbms-
normalization">Normalization</div>
    </form>

</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-8

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Positioning (static, relative, absolute, fixed, sticky)
  - Floating (float, clear)
  - CSS Color (named, hex, rgb, rgba, hsl, hsla)
- AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Positioning'

CSS Positioning allows us to place and arrange HTML elements on a webpage in various ways. There are different types of positioning in CSS, which provide precise control over an element's position in relation to its containing element or the entire web page.

Types of CSS Positioning

Static (default): Elements are placed according to the normal document flow. This is the default positioning method.

Relative: The element is positioned relative to its normal position in the document flow, allowing slight movement without affecting other elements' layout.

Example: div { position: relative; top: 20px; left: 30px;}

The div moves 20px down and 30px to the right from its original position, without affecting the surrounding elements.

Cascading Style Sheets (CSS)

CSS Advanced – 'Positioning'

Types of CSS Positioning

Absolute: The element is positioned relative to its closest positioned ancestor (non-static) or relative to the initial containing block if no positioned ancestors exist.

Example: div { position: absolute; top: 50px; left: 50px;}

The div is placed 50px from the top and left of its nearest positioned ancestor (or the page if no ancestor exists).

Fixed: The element is positioned relative to the browser window. It stays in place even when the page is scrolled.

Example: div { position: fixed; top: 0; right: 0; width: 100px;}

The div stays fixed at the top-right corner of the window, even when the page is scrolled.

Cascading Style Sheets (CSS)

CSS Advanced – 'Positioning'

Types of CSS Positioning

Sticky: The element switches between relative and fixed positioning, depending on the scroll position. It "sticks" in place when you reach a certain scroll point.

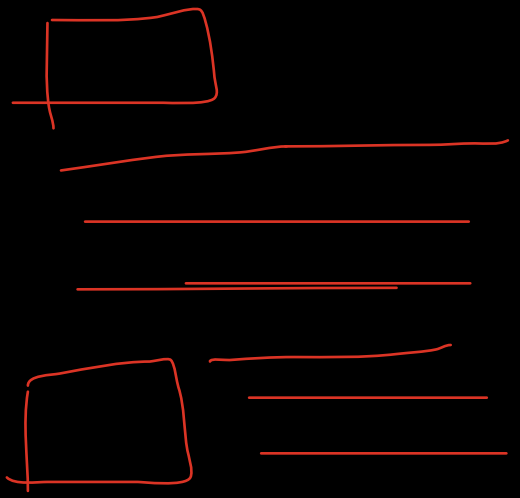
Example:
```
div {  position: sticky;  top: 10px;}
```

The div behaves normally until the page is scrolled to where the top is 10px from the viewport (user's visible area of a webpage), and then it "sticks" in place.

Cascading Style Sheets (CSS)

CSS Advanced – 'Floating'

The float property in CSS allows an element to be moved to the left or right within its container, allowing other elements to wrap around it.



Values

left:	Floats the element to the left of its container.
right:	Floats the element to the right of its container.
none:	Default value, the element does not float.

Example:

```
<img src="CSS.png" style="float: right">
<p>
CSS is a style sheet language used to describe the presentation and design of web pages.
</p>
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Floating'

Clearing Floats with the 'clear' *Property*

The clear property is used to avoid wrapping of text around a floated element.

Values

left:	Prevents wrapping around left-floated elements.
right:	Prevents wrapping around right-floated elements.
both:	Prevents wrapping around elements floated both left and right.

Example:

```
<img src="CSS.png" style="float: right">
<p style="clear: right">
CSS is a style sheet language used to describe the presentation and design of web pages.
</p>
```

Cascading Style Sheets (CSS)

CSS Advanced – 'CSS Color'

Using CSS Colors we may color the HTML elements, like text, backgrounds, borders, etc. Colors in CSS can be specified using a variety of formats: named colors, hex values, RGB, RGBA, HSL, and HSLA.

Named Colors: Predefined color names like red, blue, green.

Hexadecimal Colors: Represent colors using a hex code, e.g., #ff0000 for red.

RGB (Red, Green, Blue): Colors are defined by specifying the amount of red, green, and blue, e.g., rgb (255, 0, 0) for red.

RGBA (Red, Green, Blue, Alpha): The alpha value defines the opacity of the color, ranging from 0 (fully transparent) to 1 (fully opaque).

Cascading Style Sheets (CSS)

CSS Advanced – 'CSS Color'

HSL (Hue, Saturation, Lightness): Defines colors based on

Hue: The type of color on the color wheel.

(Red: 0°, Green: 120°, Blue: 240°).

Saturation: The intensity of the color;

Lightness: The brightness of the color;

HSLA (Hue, Saturation, Lightness, Alpha):

Adds an alpha value for transparency, just like in rgba.

Example:

```
<style>
    .NamedRed {
        color: red;
    }
    .RGBRed {
        color: rgb(255,0,0);
    }
    .RGBARed {
        color: rgba(255,0,0,1.0);
    }
    .HSLRed {
        color: hsl(0, 100%, 50%)
    }
    .HSLARed {
        color: hsla(0, 100%, 50%,1.0)
    }
</style>
```

PositioningDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Positioning</title>
    <style>
        .static {
            position: static;
        }
        .relative {
            position: relative;
            top: 50px;
            left: 50px;
        }

        .fixed {
            position: fixed;

            top: 30px;
            left: 40px;
        }

        .absolute {
            position: absolute;

            top: 20px;
            left: 20px;
        }

        .div {
```

```
            width: 200px;
            height: 200px;
            overflow: auto;

            border: 2px solid red;
        }

        .sticky {
            position: sticky;

            top: 10px;
        }
    </style>
</head>
<body>
    <h1>Positioning Types in CSS</h1>
    <p>We are demonstrating the different types
of positioning in CSS.</p>

    <!-- <div class="div relative">
        <div class="div absolute">

        </div>
    </div> -->

    <div class="div" >
        <p>Web Technology</p>
        <p>We are learning about the different
web technologies
            that are used in web development.
```

```
            The different topics covered in this
course are HTML,
            XML, CSS, Java Script, Node.js,
MongoDB, Java Servlets etc.
        </p>
        <h1 class="sticky">HTML</h1>
        <p>
            HTML is the standard language used to
structure and display content on the web.
            It defines elements like headings,
paragraphs, links, images, and more.
            HTML is the foundation of web pages,
organizing the content and giving it meaning.
            However, it primarily focuses on
structure, not styling or formatting.
        </p>
    </div>

</body>
</html>
```

FloatingDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Floating</title>
    <style>
        .float {
            float: left;
        }
```

```
        .wrapclear {
            clear: left;
        }
    </style>
</head>
<body>
    <img src="HTML.png" class="float">
    <p>
        HTML is the standard language used to
structure and display content on the web.
        It defines elements like headings,
paragraphs, links, images, and more.
        HTML is the foundation of web pages,
organizing the content and giving it meaning.
        However, it primarily focuses on
structure, not styling or formatting.
    </p>
    <p class="wrapclear">I am done with the web
technology HTML. I am done with the web
technology HTML.I am done with the web technology
HTML.I am done with the web technology HTML.</p>
</body>
</html>
```

ColorsDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Colors</title>
    <style>
```

```
        .div {
            width: 200px;
            height: 200px;
            border: 5px solid brown;

            position:fixed;

            top: 10px;
            left: 10px;

            /* background-color: red; */
            /* background-color: rgb(0, 255, 0,
0.3); */
            background-color: rgba(0, 255, 0,
0.3);
            /* background-color: #0000FF; */
            /* background-color: hsl(240,100%,
50%,); */
            /* background-color: hsla(240,100%,
50%, 0.8); */
        }
    </style>
</head>
<body>
    <img src="HTML.png">
    <div class="div"></div>

</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-9

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Image Sprites
  - Align
  - CSS Frameworks
- AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Image Sprites'

An Image Sprite is a single image file that contains multiple images, often icons or small graphics.

To display a single image from the combined image, you could use the CSS background-position property, defining the exact position of the image to be displayed.

Advantage of using CSS Image Sprite

A web page with many images, particularly many small images, such as icons, buttons, etc. can take a long time to load and generates multiple server requests.

Using the image sprites instead of separate images will significantly reduce the number of HTTP requests a browser makes to the server, which can be very effective for improving the loading time of web pages and overall site performance.

Cascading Style Sheets (CSS)

CSS Advanced – 'Image Sprites'

Steps to Use Image Sprites

1. Create or Obtain a Sprite: Combine all the required images in one file.

   Example:

2. CSS Styling with Background-Position:
   a) Use background-image to load the sprite image.
   b) Adjust background-position to display the specific image segment you want.

Cascading Style Sheets (CSS)

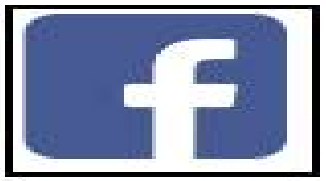
CSS Advanced – 'Image Sprites'

Example:

```
.facebook, .instagram, .whatsapp, .youtube {
        width: 145px;          height: 145px;
        background-image: url('ImageSprite/Sprite.png');
        background-repeat: no-repeat;        border: 5px solid;
    }
    .facebook  {        background-position:   0px      0px;    }
    .instagram {        background-position:   -145px 0px;     }
    .whatsapp  {        background-position:   -290px 0px;     }
    .youtube   {        background-position:   -435px 0px;     }
```

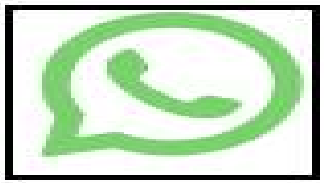
Output

Facebook Logo

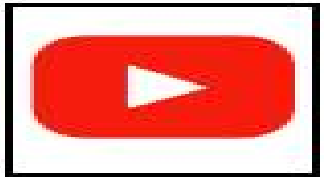


Instagram Logo

Whatsapp Logo

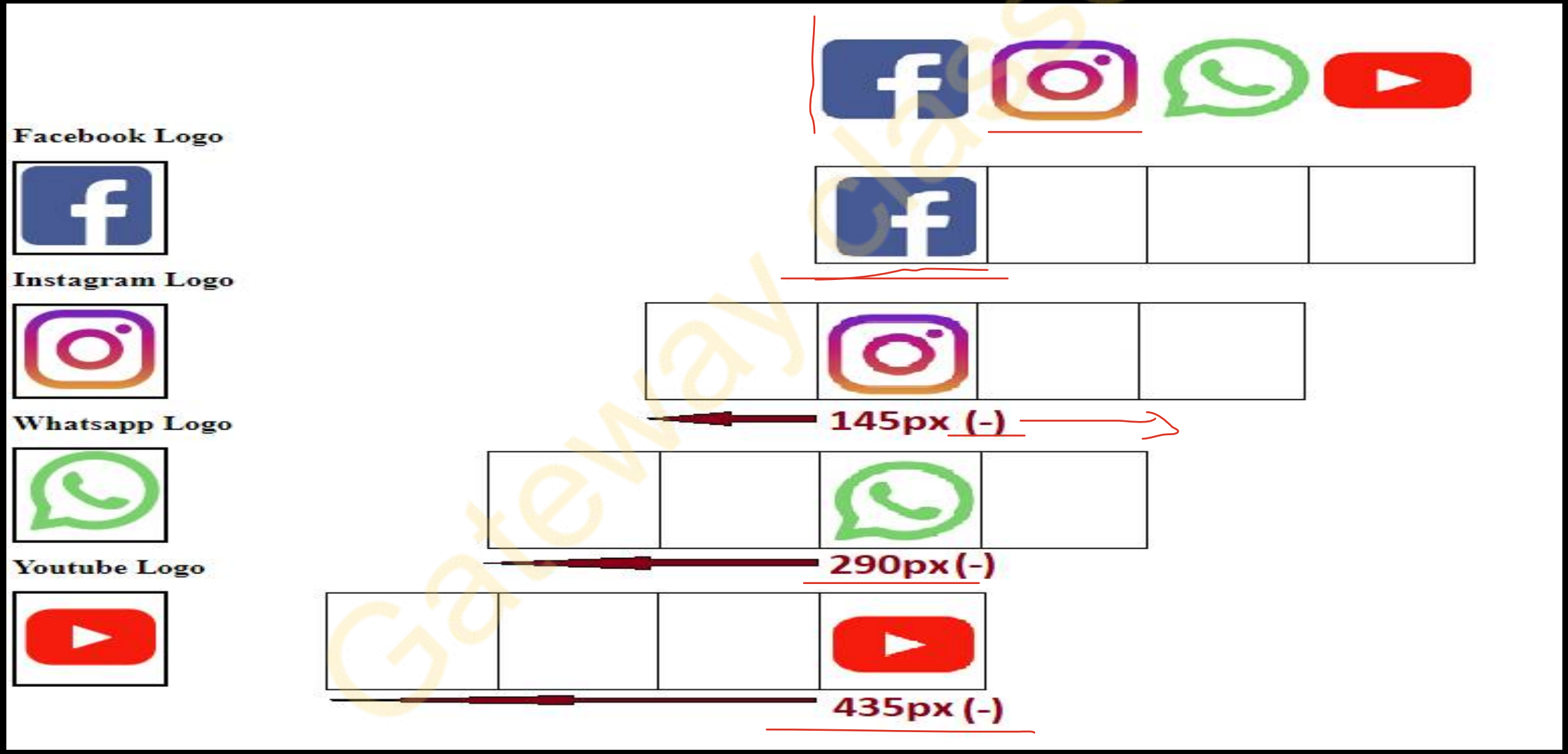


Youtube Logo



Cascading Style Sheets (CSS)

CSS Advanced – 'Image Sprites'



Cascading Style Sheets (CSS)

CSS Advanced – 'Align'

CSS alignment is used to position content and elements within their containers, either horizontally or vertically.

There are *two main aspects of alignment*:

Horizontal alignment: This refers to the positioning of elements along the horizontal axis, which typically runs from left to right. Common horizontal alignment options include:

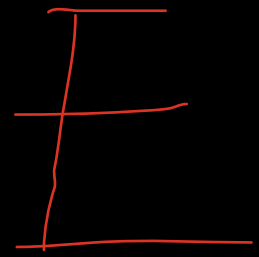
Left alignment: Elements are aligned to the left side of a container or layout.

Center alignment: Elements are positioned in the center of a container or layout horizontally.

Right alignment: Elements are aligned to the right side of a container or layout.

Cascading Style Sheets (CSS)

CSS Advanced – 'Align'



There are *two main aspects of alignment*:

Vertical alignment: This refers to the positioning of elements along the vertical axis, which typically runs from top to bottom. Common vertical alignment options include:

Top alignment: Elements are aligned to the top of a container or layout.

Middle alignment: Elements are centered vertically within a container or layout.

Bottom alignment: Elements are aligned to the bottom of a container or layout.

Cascading Style Sheets (CSS)

CSS Advanced – 'Align'

CSS Alignment Properties:

text-align	Aligns text horizontally within its container (e.g., left, right, center, justify).
vertical-align	Aligns inline elements vertically within their line (e.g., top, middle, bottom); does not work on block elements.
margin	Adds space outside the element's border to center block elements with auto value (e.g., margin: 0 auto;).
padding	Adds space inside the element's border; can indirectly adjust alignment within a container.
line-height	Sets space between lines, useful for vertically centering text within inline elements.
float	Floats an element to the left or right, allowing text or other elements to wrap around it.
position	Positions elements relative to the document or another element (e.g., absolute, relative, fixed, sticky).

Cascading Style Sheets (CSS)

CSS Advanced – 'CSS Frameworks'

A pre-prepared library of CSS styles and components that simplify web design and development, offering ready-made styles, grids, and UI elements.

CSS framework comprises several CSS stylesheets ready for use by web developers and designers. The popular CSS Frameworks are *Bootstrap*, *Tailwind CSS*, *Bulma* etc.

Advantages of using CSS Frameworks

- Developers and designers can use CSS frameworks to implement various advanced features and visual elements on a website.
- CSS frameworks make creating websites compatible with multiple browsers and browser versions.
- Since these frameworks have ready-to-use stylesheets, using them allows faster and more convenient web development.
- Developers can quickly generate a user-friendly and visually appealing UI that can be modified throughout a project without starting from scratch.

Cascading Style Sheets (CSS)

CSS Advanced – 'CSS Frameworks'

W3.CSS Framework

W3.CSS is a modern, responsive, mobile first CSS framework.
It provides equality for all browsers: Chrome. Firefox. Edge. IE. Safari. Opera.
It provides equality for all devices: Desktop. Laptop. Tablet. Mobile.
It is standard CSS only.

Using W3.CSS

```
<!DOCTYPE html>
<html>
<head><link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css"></head>
<body>
<div class="w3-container w3-blue">
  <h1>W3.CSS Demo</h1>  <p>It is an easy to use framework.</p>
</div>
</body>
</html>
```

ImageSprites.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Image Sprites</title>
    <style>
        .facebook, .instagram, .whatsapp,
.youtube {
            background-image:
url('ImageSprite/Sprite.png');
            background-repeat: no-repeat;

            width: 145px;
            height: 145px;

            border: 5px solid red;
        }

        .facebook {
            background-position: 0px 0px;
        }

        .instagram {
            background-position: -145px 0px;
        }

        .whatsapp {
            background-position: -290px 0px;
        }
```

```
        .youtube {
            background-position: -435px 0px;
        }
    </style>
</head>
<body>
    <h1>Facebook</h1>
    <div class="facebook">
    </div>

    <h1>Instagram</h1>
    <div class="instagram"></div>

    <h1>Whatsapp</h1>
    <div class="whatsapp"></div>

    <h1>Youtube</h1>
    <div class="youtube"></div>
</body>
</html>
```

AlignDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Align</title>
    <style>
        div {

            width:500px;
```

```
            height: 200px;
            margin: 0 auto;
            border: 2px dashed red;
            padding-top: 100px;
        }

        p {
            width: 200px;
            height: 100px;
            margin: 0 auto;
            line-height: 100px;
            border: 2px solid brown;
            text-align: center;
        }
    </style>
</head>
<body>
    <div>
        <p>
            Alignment
        </p>
    </div>
</body>
</html>
```

CSSFrameworksDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
```

```
    <link rel='stylesheet'
href='https://www.w3schools.com/w3css/4/w3.css'>

</head>
<body>
    <div class="w3-container w3-blue">
        <h1>Frameworks</h1>
        <p>W3.css</p>
    </div>
</body>
</html>
```

AKTU

B.Tech 5th Sem

GATEWAY CLASSES

CS IT & CS Allied

Web Technology

UNIT-2 Lecture-10

Today's Target

- Cascading Style Sheets (CSS) – Advanced
  - Creating Page Layout and Site Designs
    - Page Layout Sections
    - Flexbox
    - CSS Grid
- AKTU PYQs

By Amol Sharma sir

- Pursuing Ph.D. from IIT (BHU)
- Wipro Certified Faculty (WCF)

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Page Layout and Site Design

The webpages of a website can be divided into various sections comprising of *header, menus, content, and footer* based on which there are many different layout designs available for developers.

Page Layout Sections

Header

The header is located at the top of the webpage and typically contains the website's title, logo, and branding elements. It serves as a consistent starting point for users, providing context for the site's content.

Navigation Bar

The navigation bar is usually positioned below the header and includes links to the main sections of the website. It allows users to easily navigate between different pages or categories, enhancing usability.

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Page Layout Sections

Content

The content area occupies the main central space of the webpage, displaying the primary information or features of the site. It can include text, images, videos, and other media relevant to the selected topic.

Footer

The footer is situated at the bottom of the webpage and often contains copyright information, contact details, and additional links. It serves as a closing section, providing users with supplementary information and resources.

Cascading Style Sheets (CSS)

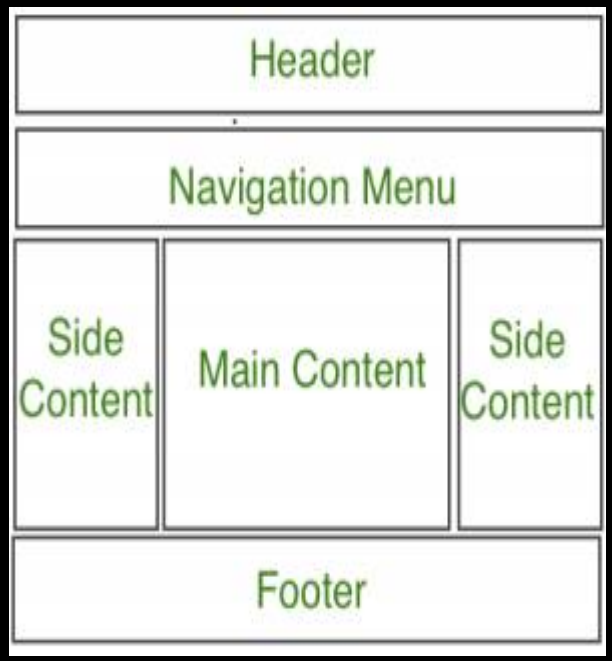
CSS Advanced – 'Creating page Layout and Site Designs'

Page Layouts

The content section may be divided in n number of parts as needed.
The most common layouts are:

1-Colum Layout
For Mobile devices with small screen size)

2-Colum Layout
For Tablets/ Laptops with medium screen size)



3-Colum Layout
For Desktops with large screen size)

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Modern tools for page layout and site designs

Flexbox (Flexible Box Layout)

It is a CSS layout mode designed for efficient arrangement of items within a container. It allows for responsive and flexible layouts that adapt to different screen sizes and orientations.

Features of Flexbox

1. It easily creates responsive layouts that adjust according to the available space.
2. It provides powerful alignment options for both horizontal and vertical alignment of items.
3. It allows changing the visual order of items without altering the HTML structure.
4. It distributes space effectively between items using properties like justify-content and align-items.
5. It enables items to grow or shrink based on the available space with flex-grow and flex-shrink.

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Modern tools for page layout and site designs

Flexbox (Flexible Box Layout)

Example:

```
.container {
    display: flex;                    /* Enables Flexbox */
    justify-content: space-between;   /* Distributes space between items */
    align-items: center;              /* Vertically centers items */
    flex-wrap: wrap;                  /* Allows items to wrap onto multiple lines */
}
.item {
    flex: 1;                          /* Makes each item grow equally */
    margin: 10px;                     /* Adds space around items */
    padding: 20px;                    /* Adds space inside items */
    background-color: lightblue;      /* Background color for visibility */
}
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Modern tools for page layout and site designs

CSS Grid

CSS Grid Layout is a two-dimensional layout system for the web that enables the creation of complex layouts using rows and columns.

It allows designers to control the positioning and sizing of items within a grid, making responsive designs simpler and more intuitive.

Features of CSS Grid

1. It facilitates the creation of complex layouts with rows and columns in a straightforward way.
2. It allows precise placement of items within the grid using properties like grid-column and grid-row.
3. It supports responsive design by adjusting the layout based on screen size.
4. It simplifies alignment of items in both horizontal and vertical directions with properties like align-items and justify-items.

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Modern tools for page layout and site designs

CSS Grid

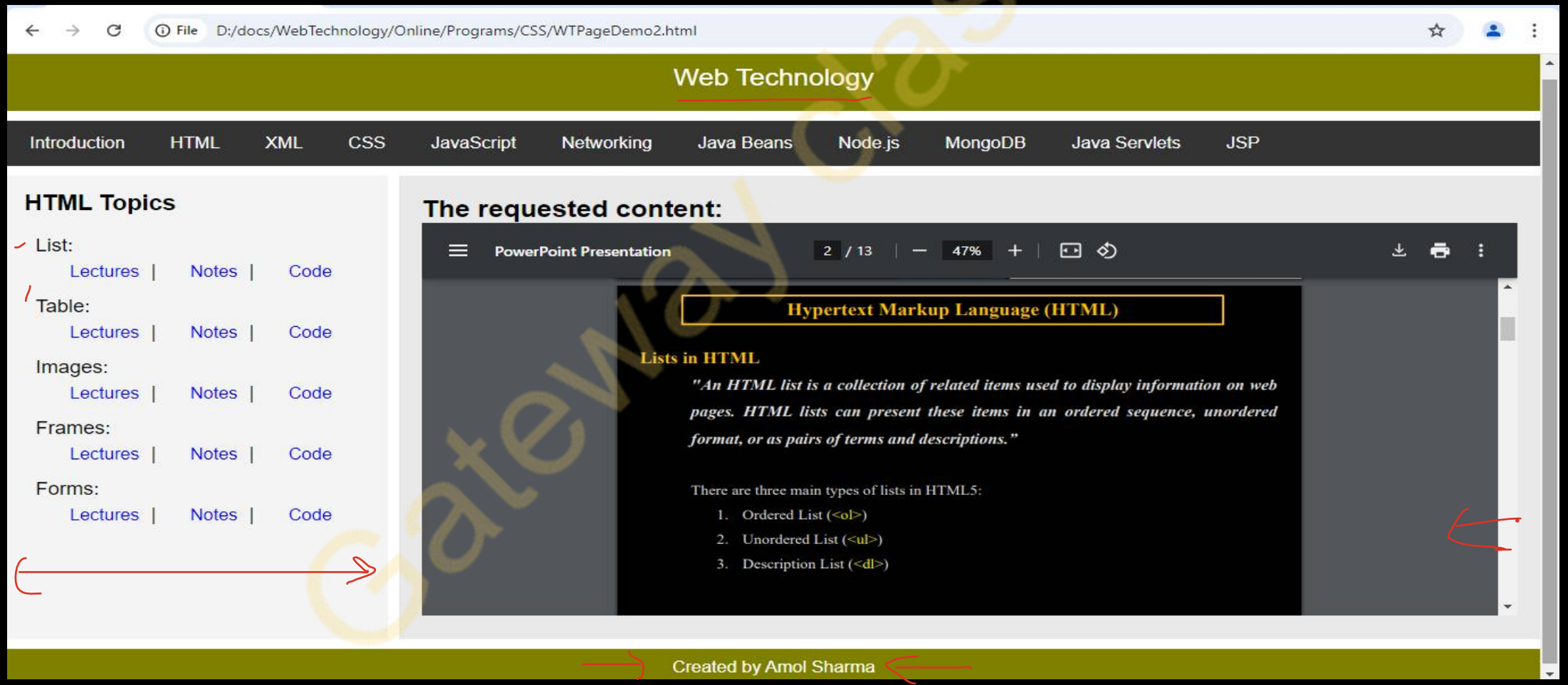
Example:

```
.container {
    display: grid;                          /* Enables Grid layout */
    grid-template-columns: repeat(3, 1fr);  /* Creates three equal columns */
    grid-gap: 10px;                         /* Adds space between grid items */
}
.item {
    background-color: lightgreen;           /* Background color for visibility */
    padding: 20px;                          /* Adds space inside items */
}
.item1 {
    grid-column: span 2;                    /* Makes this item span 2 columns */
    grid-row: 1;                            /* Places this item in the first row */
}
```

Cascading Style Sheets (CSS)

CSS Advanced – 'Creating page Layout and Site Designs'

Example



FlexboxDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Flex Box Demo</title>
    <style>
        .container {
            display: flex;
/* Enables Flexbox */
            justify-content: space-between;
/* Distributes space between items */
            align-items: center;
/* Vertically centers items */
            flex-wrap: wrap;
/* Allows items to wrap onto multiple lines */
        }
        .item {
            flex: 1;
/* Makes each item grow equally */
            margin: 10px;
/* Adds space around items */
            padding: 20px;
/* Adds space inside items */
            background-color: lightblue;
/* Background color for visibility */
        }

    </style>
</head>
<body>
```

```
    <div class="container">
        <div class="item">Item 1</div>
        <div class="item">Item 2</div>
        <div class="item">Item 3</div>
    </div>
</body>
</html>
```

CSSGridDemo.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Grid Demo</title>
    <style>
        .container {
            display:
grid;
/* Enables Grid layout */
            grid-template-columns: repeat(3,
1fr);
/* Creates three equal columns */
            grid-gap:
10px;
/* Adds space between grid items */
        }
        .item {
            background-color:
lightgreen;
/* Background color for visibility */
```

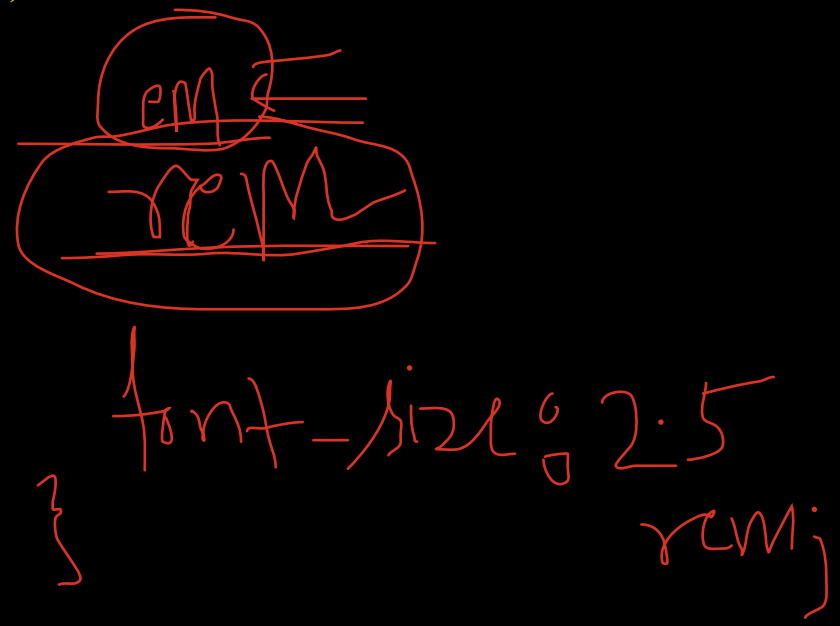
```
            padding:
20px;
/* Adds space inside items */
        }
        .item1 {
            grid-column: span
2;
/* Makes this item span 2 columns */
            grid-row: 1;
/* Places this item in the first row */
        }

    </style>

</head>
<body>
    <div class="container">
        <div class="item item1">Item 1 (Spans 2
columns)</div>
        <div class="item">Item 2</div>
        <div class="item">Item 3</div>
        <div class="item">Item 4</div>
        <div class="item">Item 5</div>
    </div>
</body>
</html>
```

AKTU PYQs

1. Describe the advantages of CSS. (AKTU 2021-22)
2. Create a CSS rule that makes all the text 2 times larger than the base font of the system. Mention can you integrate CSS on a web page. (AKTU 2021-22)
3. Define box model in CSS with block diagram. (AKTU 2019-20)
4. Explain CSS. What are the CSS frameworks? Explain in brief. What are the different ways of using the stylesheet? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. (AKTU 2019-20)
5. What do you mean by CSS? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. Mention how can you integrate CSS on a web page? (AKTU 2018-19)



Web Designing (AKTU 2023 – 24)

1. Describe the syntax of CSS.
2. Describe the different properties to set the background of a web page.
3. How will you apply CSS to text?

Web Designing (AKTU 2022 – 23)

1. Explain different types of selectors in CSS.
2. Discuss advantages of CSS.
3. Illustrate Box model in CSS with block diagram.
4. Explain the CSS and its basic properties. Outline a CSS rule that makes all the text 2.5 times larger than the base font of the system.
5. Explain cascading and the style precedence rules when using multiple approaches.

AKTU PYQs

Web Designing (AKTU 2021 – 22)

1. What are Style Sheets?
2. Explain different types of selectors in CSS with example.
3. What are the techniques to use W3.CSS framework? Explain.
4. Explain three different ways to implement CSS on web page.
5. What is CSS? List out the various CSS properties. Explain the various concepts of CSS properties with neat example.

Web Designing (AKTU 2020 – 21)

1. How external style sheet is useful in web page design?
2. Discuss the box model in CSS with block diagram.
3. What is pseudo class in CSS?
4. Explain any eight CSS text properties in detail.
5. What is CSS? What are the different ways to create them? Explain with the help of an example.
6. Apply CSS to a web page with the following requirements:
   i) Add a background image of a submarine.
   ii) Set a color to the span elements (differnt color for each class).


Thank you
Gateway Classes