



Unit 3

CSS CASCADING STYLE SHEETS

2CP08: WEB TECHNOLOGIES3

K J SHARMA

Introduction to CSS

- ▶ CSS stands for Cascading Style Sheets and is a simple styling language which allows attaching style to HTML elements.
- ▶ Simple design language intended to **simplify** the process of making web pages **presentable**.
- ▶ Defines how to display HTML contents.

Benefits of Cascading Style Sheets

- ▶ Powerful and flexible way to specify the formatting of HTML elements.
- ▶ Improves website presentation.
- ▶ Can define font, size, background color, background image , margins etc..
- ▶ Share style sheets across multiple documents or entire web site.

Understanding Style Rules

- ▶ The style characteristics for an HTML element are expressed by Style Rules .
- ▶ A set of style rules is called a Style Sheet.
- ▶ Style rules are contained in the <STYLE> element in the document's <HEAD> section.
- ▶ A Style Rule is composed of two parts: a selector and a declaration.

Understanding Style Rules

- ▶ The Selector indicates the element to which the rule is applied.
- ▶ The Declaration determines the property values of a selector.
- ▶ The Property specifies a characteristic, such as color, font-family, position, and is followed by a colon (:).
- ▶ The Value expresses specification of a property, such as red for color, arial for font family, 12 px for font-size, and is followed by a semicolon (;).
- ▶ The Style Sheet Property Names are case-sensitive.

Understanding Style Rules

- ▶ The <STYLE> element contains <TYPE> attribute. The value “text/css” defines the style language as Cascading Style Sheets.

Introduction to CSS

Selector

Declaration Block

```
body {  
    color: black;  
    background: white;  
    margin: 8px;  
}
```

Value

Attribute Name

Types of CSS

- ▶ Three Ways to Specifying CSS:
 - ▶ Inline styles
 - ▶ Internal styles
 - ▶ External style

1) Inline Style

- ▶ Inline styles are the least flexible type of style to implement
- ▶ Inline CSS has the **highest priority** out of external and internal CSS.
- ▶ This means that you can **override styles** that are defined in external or internal by using inline CSS.
- ▶ You can apply inline styles to any tag
- ▶ It suffers the some major drawback - that if you wanted to change the style properties, you would have to edit each and every instance of the style on every single page of your website.

2) Internal Style Sheets

- ▶ With internal style sheets, a web page's styles are all specified at the top of the page code, within the `<head>` tag.

3) External Style Sheets

- ▶ It allow you to put all of your styling information into a completely separate CSS file.
- ▶ We can then simply reference this file from within each web page, and the page's content will then be styled accordingly.
- ▶ The obvious huge advantage of this method is that you need only change a style in your stylesheet file(.css file), and the changes will cascade through the rest of your website.

3) External Style

Sheets(Cont.)

Linking to an External Stylesheet

- ▶ To link a Web page to an external style sheet a `<LINK>` element should be added within `<HEAD>` element of a document with the URL to a style sheet. It tells the browser to find a specified style sheet.

3) External Style Sheets(Cont.)

HREF attribute states the relative URL to the style sheet

The REL attribute describes the relationship between the current document and the document identified by the *href* attribute

```
<Head>  
<Title>Demo_1</Title>  
  
<Link href="style1.css" Rel="stylesheet">  
  
</Head>
```

3) External Style Sheets(Cont.)

► Example:

Demo.HTML

```
<h1>How are you?</h1>
```

```
</body>
```

```
</html>
```

test.CSS

```
h1{
```

```
color:blue;
```

```
}
```

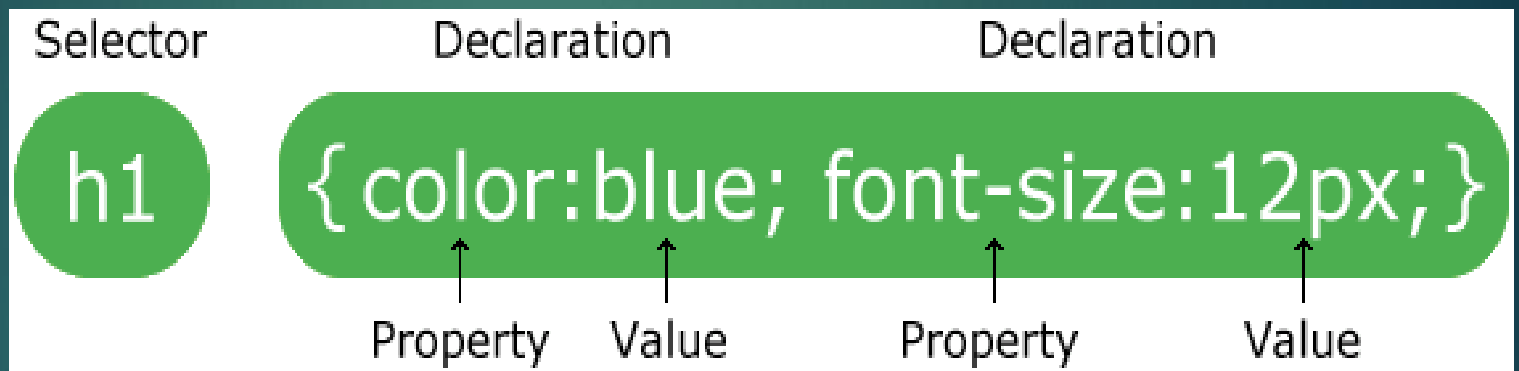
Output

Hello Friends

How are you?

Selectors

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



The element Selector

- ▶ The element selector selects elements based on the element name.
- ▶ It refers to HTML tags.
- ▶ For example:
 - ▶ You can select all <h1> elements on a page like this

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


The id Selector

- ▶ The id selector is used to specify a style for a **single, unique** element.
- ▶ The id selector uses the id attribute of the HTML element, and is defined with a “#” in CSS.
- ▶ The style rule below will be applied to the HTML element with id=“first”

CSS

```
#first{  
    color:blue;  
}
```

The class Selector

- ▶ The class selector selects elements with a specific class attribute.
- ▶ To select elements with a specific class, write a period (.) character, followed by the name of the class.
- ▶ For example all HTML elements with class="first" will be blue and text-aligned right.

```
CSS
.first{
    text-align : right;
    color : blue;
}
```

The class Selector

- ▶ You can also specify that only specific HTML elements should be affected by a class.
- ▶ For example only `<h1>` elements with `class="first"` will be blue and text-aligned right.

CSS

```
h1.first{  
    text-align : right;  
    color : blue;  
}
```

The class Selector

- ▶ HTML elements can also refer to more than one class.
- ▶ For example the `<h1>` element will be styled according to class: “first” and “second”

CSS

```
h1.first{
    text-align : right;
    color : blue;
}
h1.second{
    font-size: 20%;
}
```

Grouping Selectors

- ▶ If you have elements with the same style definitions then it will be better to group the selectors, to minimize the code.
- ▶ To group selectors, separate each selector with a comma.

Grouping Selectors

- ▶ For example if h1 and p have same style definitions then grouped the selectors like below:

Demo.HTML

```
<h1>How are you?</h1>  
  
</body>  
</html>
```

test.CSS

```
h1, p{  
    color:blue;  
}
```

Output

Hello Friends
How are you?

Pseudo-Elements

- ▶ A CSS pseudo-element is used to style specified parts of an element.
- ▶ For example, it can be used to:
 - ▶ Style the first letter, or line, of an element
 - ▶ insert content before, or after, the content of an element
- ▶ Syntax:

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

Pseudo-Elements

1. ::first-line

- ▶ The ::first-line pseudo-element is used to add a special style to the first line of a text.
- ▶ Syntax:

```
p::first-line {  
    color: red;  
    font-variant: small-caps;  
}
```


Pseudo-Elements

2. ::first-letter

- ▶ The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

```
p::first-letter{  
    color: #ff0000;  
    font-size: xx-large;  
}
```

- ▶ Pseudo-elements can be combined with CSS classes:

```
p.intro::first-letter {  
    color: #ff0000;  
    font-size:200%;  
}
```

Pseudo-Elements

3. ::before

- ▶ The ::before pseudo-element can be used to insert some content before the content of an element.
- ▶ Syntax:

```
h1::before {  
    content: url(smiley.gif);  
}
```

Pseudo-Elements

4. ::after

- ▶ The ::after pseudo-element can be used to insert some content after the content of an element.

Pseudo-Elements

5. ::selection

- ▶ The ::selection pseudo-element matches the portion of an element that is selected by a user.

- ▶ Syntax:

```
::selection{  
  color: red;  
  background: yellow;  
}
```

Multiple Pseudo-elements

- ▶ Several pseudo-elements can also be combined.
- ▶ In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}
```

```
p::first-line {  
  color: #0000ff;  
  font-variant: small-caps;  
}
```

Pseudo-classes

- ▶ A pseudo-class is used to define a special state of an element.
- ▶ For example, it can be used to:
 - ▶ Style an element when a user mouses over it
 - ▶ Style visited and unvisited links differently
 - ▶ Style an element when it gets focus
- ▶ Syntax:

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

Pseudo-classes

► Anchor Pseudo-classes

► Links can be displayed in different ways

1. a:link – a normal, unvisited link
2. a:visited – a link the user has visited
3. a:hover – a link when the user mouse over
4. a:active – a link the moment it is clicked

```
a:link{  
    color:red;  
    /* unvisited link */  
}
```

```
a:visited{  
    color:green;  
    /* visited link */  
}
```

```
a:hover{  
    color:pink;  
    /* mouse over link */  
}
```

```
a:active{  
    color:Green;  
    /* selected link */  
}
```

Pseudo-classes and CSS Classes

- ▶ Pseudo-classes can be combined with CSS classes:
- ▶ When you hover over the link in the example, it will change color:

```
a.highlight:hover {  
    color: Red;  
}
```


Pseudo-classes

- ▶ Hover on <div>
- ▶ An example of using the :hover pseudo-class on a <div> element:

```
div:hover {  
    background-color: blue;  
}
```

Pseudo-classes

- ▶ Simple Tooltip Hover
- ▶ Hover over a <div> element to show a <p> element

```
▶ p {  
  display: none;  
  background-color: yellow;  
  padding: 20px;  
}
```

```
div:hover p {  
  display: block;  
}
```

Pseudo-classes

first-child Pseudo-class

- ▶ The :first-child pseudo-class represents the first element among the group of sibling elements.
- ▶ In the following example, Select and style every <p> element that is the first child of its parent:

```
p:first-child{  
    background-color: yellow;  
}
```

Pseudo-classes

Match the first <i> element in all <p> elements

- ▶ In the following example, the selector matches the first <i> element in all <p> elements:
- ▶ Example

```
p i:first-child {  
    color: blue;  
}
```

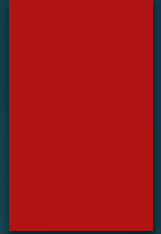
Pseudo-classes

Match all `<i>` elements in all first child `<p>` elements

- ▶ In the following example, select and style every `<i>` element of every `<p>` element, where the `<p>` element is the first child of its parent:
- ▶ Example

```
p:first-child i {  
    color: blue;  
}
```

Background Property



	Property Name
▶ Background Color	(background-color)
▶ Background Image	(background-image)
▶ Background Image Repeat repeat)	(background-
▶ Fixed Background Image attachment)	(background-
▶ Background Image Positioning	(background-position)

Background Color

- ▶ The **background-color** property specifies the background color of an element.
- ▶ The background color of a page is defined in the body selector
- ▶ Below is example of CSS backgrounds

Test.css

```
body
{
    background-color : red;
    background-color : #FF0000;
    background-color : rgb(255,0,0);
}
```

Background Image

- ▶ The **background-image** property specifies an image to use as the background of an element.
- ▶ For example,

Test.css

```
body  
{  
  
}
```


Background Image Repeat

- ▶ By default the background-image property repeats an image both horizontally and vertically.
- ▶ It can be change according to requirement by specifying different values like:
 - ▶ Repeat
 - ▶ Repeat-x(image is repeated only horizontally)
 - ▶ Repeat-y(image is repeated only vertically)

Test.css

```
body
{
    background-image : url(image1.jpg);
    background-repeat : no-repeat;
}
```

Fixed Background Image

- ▶ The **background-attachment** property sets whether a background image is fixed or scrolls with the rest of the page.

scroll : The background image scrolls with the rest of the page.

fixed : The background image is fixed

Test.css

```
body
{
    background-image : url(image1.jpg);
    background-repeat : no-repeat;
    background-attachment : fixed;
}
```

Background Image Positioning

- ▶ It specifies how to position a background-image.
- ▶ By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.
- ▶ It contain different values like: left top, left center, left bottom, right top, right center, right bottom, center top, center center, center

Test.css

```
body
{
    background-image : url(image1.jpg);
    background-repeat : no-repeat;
    background-position : 20px 10px;
    background-position : 30%30%;
    background-position : top center;
}
```

Background - Shorthand property

- ▶ To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.
- ▶ The shorthand property for background is simply **"background"**

Test.css

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

Font Property

- ▶ CSS font properties define the font family, boldness, size and the style of a text.

Property Name

- | | |
|-----------------|----------------|
| 1. Font color | (color) |
| 2. Font Family | (font-family) |
| 3. Font Size | (font-size) |
| 4. Font Style | (font-style) |
| 5. Font Weight | (font-weight) |
| 6. Font Variant | (font-variant) |

Font Property

► Font Color

- Set the text-color for different elements
- The default color for a page is defined in the body selector.

```
h4{  
    color : red;  
}
```

► Font Family

- The font family of a text is set with the font-family property.

```
h4{  
    font-family : arial;  
}
```

Font Property

► Font Size

- The font size property sets the size of the text.

```
h4{  
    font-size :120%;  
    font-size :10px;  
    font-size : small;  
    font-size : smaller;  
    font-size : x-small;  
    font-size : xx-small;  
    font-size : large;  
    font-size : larger;  
    font-size : x-large;  
    font-size : xx-large;  
    font-size : medium;  
}
```

Font Property

► Font style

- The font-style property is mostly used to specify italic text.

```
h4{  
    font-style : italic;  
}
```

► Font Weight

- The font-weight property sets how thick or thin characters in text should be displayed.

```
h4{  
    font-weight : 300;  
    font-weight : bolder;  
    font-weight : lighter;  
}
```

► Font Variant

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.

```
h4{  
    font-variant : small-caps;  
}
```


Text Property

- ▶ While CSS Font covers most of the traditional ways to format your text, CSS Text allows you to control the spacing, decoration and alignment of your text.

Property Name

- | | |
|--------------------|-------------------|
| 1. Text Decoration | (text-decoration) |
| 2. Text Indent | (text-indent) |
| 3. Text Align | (text-align) |
| 4. Text Transform | (text-transform) |
| 5. White Space | (white-space) |
| 6. Word Spacing | (word-spacing) |
| 7. Letter Spacing | (letter-spacing) |
| 8. Line height | (line-height) |

Text Property

► Text Decoration

- The text-decoration property is used to set or remove decorations from text.
- The text-decoration property is mostly used to remove underlines from links for design purposes.

```
h4{  
    text-decoration : line-through;  
    text-decoration : overline;  
    text-decoration : underline;  
    text-decoration : underline overline dotted red;  
    text-decoration : none;  
}
```

► Text Indent

- The text-indentation property is used to specify the indentation of the first line of a text block.

```
h4{  
    text-indent : 20px;  
}
```

Text Property

► Text Align

- The text-align property is used to set the horizontal alignment of a text.

```
h4{  
    text-align : right;  
    text-align : justify;  
    text-align : left;  
    text-align : center;  
}
```

► Text Transform

- The text-transform property is used to specify upper

```
h4{  
    text-transform : uppercase;  
    text-transform : lowercase;  
    text-transform : capitalize;  
}
```

Text Property

▶ White Space

- ▶ The white-space attribute allows you to prevent text from wrapping until you place a break `
` into your text.

```
h4{  
    white-space : nowrap;  
}
```

▶ Word Spacing

- ▶ With the CSS attribute word-spacing you are able to specify the exact value of the spacing between your words. Word-spacing should be defined with exact values.

```
h4{  
    word-spacing : 10px;  
}
```

Text Property

▶ Letter Spacing

- ▶ With the CSS attribute letter-spacing you are able to specify the exact value of the spacing between your letters. Letter-spacing should be defined with exact values.

```
h4{  
    letter-spacing : 3px;  
}
```

▶ Line Height

- ▶ The line-height attribute will set the height of the line in the page.

```
h4{  
    line-height : 10px;  
}
```

List Property

- ▶ The CSS list properties allow you to:
 - Set different list item markers for ordered and unordered lists
 - Set an image as the list item marker
 - Set the position of the marker
- ▶ CSS List Style Type

```
ul,ol{  
    list-style-type : circle;  
    list-style-type : disc;  
    list-style-type : square;  
    list-style-type : armenian;  
    list-style-type : decimal;  
    list-style-type : decimal-leading-zero;  
    list-style-type : lower-alpha;  
    list-style-type : lower-greek;  
}
```

List Property

▶ CSS List with Image

```
ol{  
    list-style-image : url('imgpath');  
}
```

▶ CSS List Position

```
ol{  
    list-style-position : outside;  
    list-style-position : inside;  
}
```

List Property

► CSS List Shorthand Property

- It is also possible to specify all the list properties in one, single property.

```
ul{  
    list-style : square url('imgpath');  
}
```


Table Property

► Table Borders

- To specify table borders in CSS, use the border property.

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

- The border-collapse property sets whether the table borders are collapsed into a single border or separated

Table Property

► Table Height, Width & Text Alignment

- To specify width and height of table or cell use the width and height property.
- To specify horizontal and vertical alignment, use text-align and vertical-align attribute.

```
table, th, td{
    width : 100%;
}
th{
    height : 75px;
}
td{
    text-align : right;
    vertical-align : bottom;
}
```

Table Property

► Table Color

- Use to specify the background color and text color.

```
th{  
    background-color : green;  
    color : white;  
}
```

► Table Padding

- To control the space between the border and content in a table, use the padding property on td and th elements

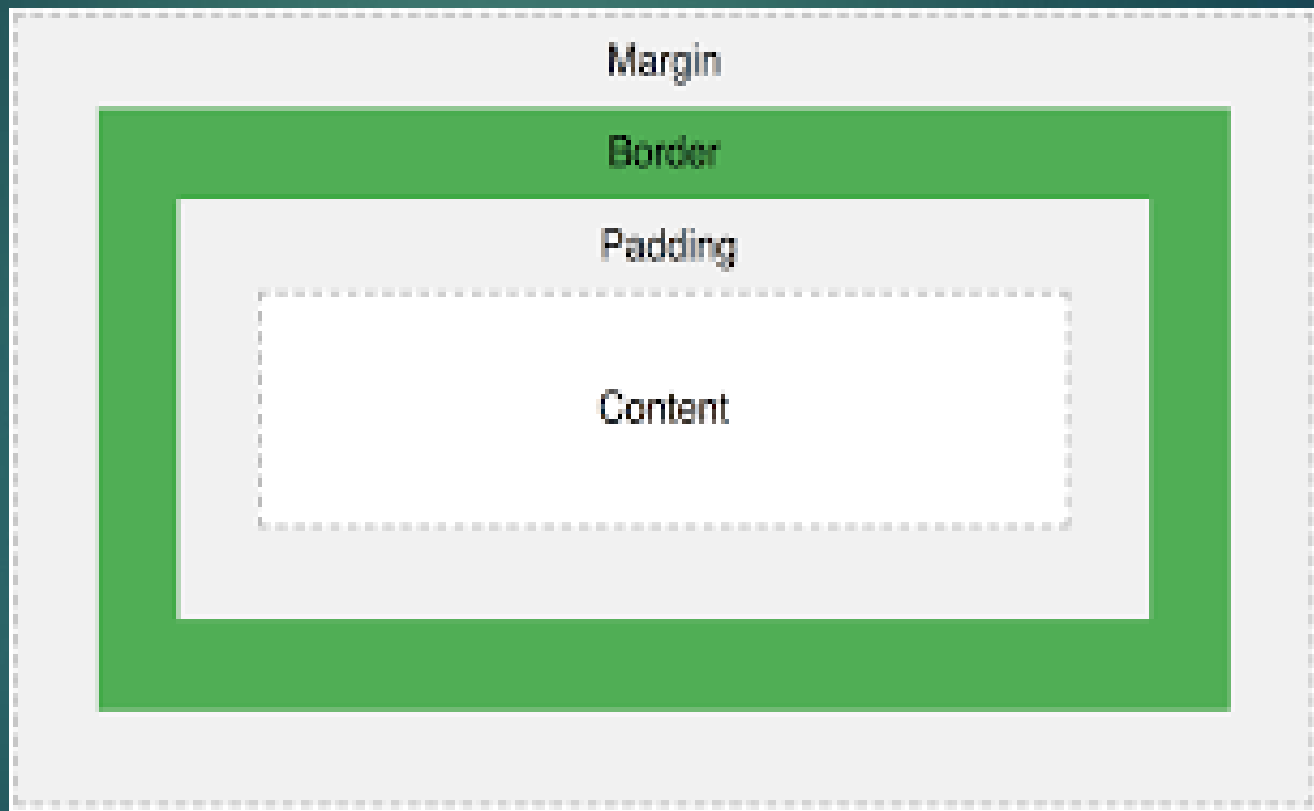
```
td{  
    padding : 15px;  
}
```

The CSS Box Model

- ▶ All HTML elements can be considered as boxes. In CSS, the term “box model” is used when talking about design and layout.
- ▶ The CSS box model is essentially a box that wraps around HTML elements, and it consists of: **margins**, **borders**, **padding**, and the actual **content**.
- ▶ The box model allows us to place a border around elements and space elements in relation to other elements.

The CSS Box Model

- The image below illustrate the box model:



CSS Padding

- ▶ The CSS padding properties defines the space between the element border and the element content.
- ▶ The top, right, bottom and left padding can be changed independently using separate properties.
- ▶ A shorthand padding property can also be used to, change all padding at once.

```
h4{  
    padding : 10px;  
}
```

```
h4{  
    padding-top : 10px;  
    padding-right : 20px;  
    padding-bottom : 30px;  
    padding-left : 40px;  
}
```

```
h4{  
    padding : 10px 20px 30px 40px;  
}
```

CSS Border

- ▶ The CSS border properties allow you to specify the style and color of an element's border.
- ▶ Border Style Types
 - The `border-style` properties specifies what kind of border to display.
- ▶ Border Width
 - The `border-width` property is used to set the width of the border.
- ▶ Border color
 - The `border-color` property is used to set the color of the border.
 - Border colors can be any color defined by RGB, hexadecimal or key terms.
- ▶ The top, right, bottom and left border can be changed independently using separate properties.

```
h4{  
    border : 1px solid red;  
}
```

```
h4{  
    border-style : solid;  
    border-style : dotted;  
    border-style : double;  
}
```

```
h4{  
    border-width : 7px;  
}
```

```
h4{  
    border-color : red;  
}
```

```
h4{  
    border-top : 1px solid red;  
}
```

CSS Margin

- ▶ The CSS margin properties define the space around elements.
- ▶ The top, right, bottom and left margin can be changed independently using separate properties.
- ▶ A shorthand margin property can also be used, to change all margins at once.

```
h4{  
    margin : 10px;  
}
```

```
h4{  
    margin-top : 10px;  
    margin-right : 20px;  
    margin-bottom : 30px;  
    margin-left : 40px;  
}
```

```
h4{  
    margin : 10px 20px 30px 40px;  
}
```


CSS Positioning

▶ Absolute Positioning

- ▶ With absolute positioning, you define the exact pixel value where the specified HTML element will appear.
- ▶ The point of origin is the top-left of the browser's viewable area, so be sure you are measuring from that point.

```
h4{  
    position : absolute;  
    left : 50px;  
    top : 100px;  
}
```

▶ Relative Positioning

- ▶ Relative positioning changes the position of the HTML element relative to where it normally appears.

```
h4{  
    position : relative;  
    left : 50px;  
    top : 100px;  
}
```

▶ Fixed Positioning

- ▶ The element is positioned relative to the browser window, in fixed position, element will be in the same place even we scroll the screen.

```
h4{  
    position : fixed;  
    left : 50px;  
    top : 100px;  
}
```

CSS Layers

- ▶ CSS allows you to control which item will appear on top with the use of layers.
- ▶ In CSS, each element is given a priority.
- ▶ If there are two overlapping CSS positioned elements, the element with the higher priority will appear on top of the other.
- ▶ To manually define a priority, set the z-index value. The larger the value, the higher the priority the element will have.

HTML

```
<div id="division1">  
</div>  
<div id="division2">  
</div>
```

```
#division1{  
    position : absolute;  
    height : 100px;  
    width : 100px;  
    left : 100px;  
    top : 150px;  
    background-color :  
red;  
    z-index : 5;  
}
```

```
#division2{  
    position : absolute;  
    height : 200px;  
    width : 200px;  
    left : 50px;  
    top : 100px;  
    background-color :  
blue;  
    z-index : 2;  
}
```

Bookmark in CSS

- ▶ Bookmarks can be useful if a web page is very long.
- ▶ To create a bookmark - first create the bookmark, then add a link to it.
- ▶ When the link is clicked, the page will scroll down or up to the location with the bookmark.

Bookmark in CSS

- ▶ Example:
- ▶ First, use the id attribute to create a bookmark:
`<h2 id="bottom">Bottom</h2>`
- ▶ Then, add a link to the bookmark ("Jump to bottom"), from within the same page:
`Jump to Bottom`

Thumbnail Image

- ▶ A thumbnail is a small image that represents a larger image (when clicked on), and is often recognized with a border around it:
- ▶ How To Create a Thumbnail Image
 - ▶ Use an `` element and wrap an `<a>` element around it. Style the image with a border and add a hover effect