Web Technology & Programming

CSS
(cascaded style sheet)

LECTURER: KRUNAL PATEL
ICCT ENGG. COLLEGE
NEW VALLABH VIDYANAGAR

Introduction

- CSS is the acronym for: ***'Cascading Style Sheets'.***
- CSS stands for Cascading Style Sheets and is a simple styling language which allows attaching style to HTML elements.
- Style Sheets are templates, very similar to templates in desktop publishing applications, containing a collection of rules declared to various selectors (elements).
- Cascade is a method of defining the weight (importance) of individual styling rules thus allowing conflicting rules to be sorted out should such rules apply to the same selector.

Benefits of Cascading Style Sheets

- Powerful and flexible way to specify the formatting of HTML elements.
- Can define font,size,background-color, background-image , margins etc..
- Share style Sheets across multiple documents or entire web site.
- Can specify a class definition for a style effectively defining new HTML elements.
- Rules are applied in a hierarchical manner (precedence rules).

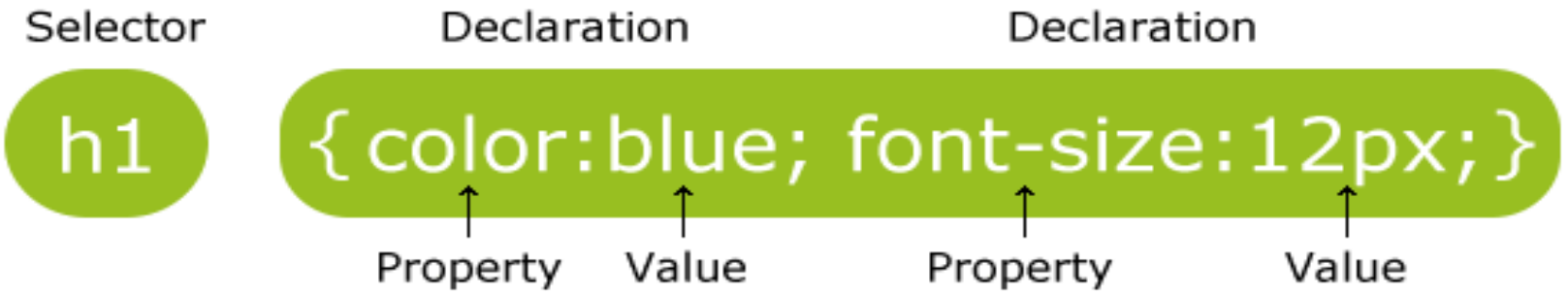
General form of rule

Selector {property : value}

Or

```
Selector {
        property1 : value1;
        property2 : value2;
        propertyN : valueN
    }
```

- "Property" is the CSS element you wish to manipulate and "VALUE" represents the value of the specified property.



Selector

Declaration Block

```
body {
     font-family: Tahoma, Arial, sans-serif;
     color: black;
     background: white;
     margin: 8px;
}
```

Value

Attribute Name

General form of rule

- 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.

```
<Head>
<Style type="text/css">
P
{color:blue;  font-size: 24pt;}
</Style>
</Head>
```

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 pt for font-size, and is followed by a semicolon (;).

Understanding Style Rules

- The <STYLE> element always contains <TYPE> attribute. The value “text/css” defines the style language as Cascading Style Sheets.
- External Style Sheet has a .css extension.

CSS Comments

- Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.
- A CSS comment begins with "/*", and ends with "*/", like this:

Example:

```
/*This is a comment*/
   p
   {
         text-align:center;
         /*This is another comment*/
         color:black;
         font-family:arial;
   }
```

CSS

- When a browser reads a style sheet, it will format the document according to it.
- **Three Ways to Insert CSS**
  - There are three ways of inserting a style sheet:
    - External style sheet
    - Internal style sheet
    - Inline style

External Style Sheet

- An external style sheet can be written in any text editor.
- The file should not contain any html tags.
- Your style sheet should be saved with a .css extension.

An example of a style sheet file is shown below:

```
hr {color:red;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

External Style Sheet

- An external style sheet is ideal when the style is applied to many pages.
- With an external style sheet, ***you can change the look of an entire Web site by changing one file.***
- Each page must link to the style sheet using the ***<link> tag***.
- The <link> tag goes inside the head section:

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

Internal Style Sheet

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

```
<style type="text/css">
        hr
        {color:sienna;}
        p
        {margin-left:20px;}
        body
        {background-image:url("images/back40.gif");}
</style>
```

Inline Styles

- An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!
- To use inline styles you use the style attribute in the relevant tag.
- The style attribute can contain any CSS property.
- The example shows how to change the color and the left margin of a paragraph:

```
<p style="color:red;margin-left:20px">
        This is a paragraph.
</p>
```

- Example : internal style sheet

```
<html>
    <head>
        <style type="text/css">
            body
            {
                    background-color:#b0c4de;
            }
        </style>
    </head>
    <body>
        <h1>My CSS web page!</h1>
        <p>Hello world! This is a CSS example.</p>
    </body>
</html>
```

First.css

```
body
{
background-
   color:#b0c4de;
}
```

```
<html>
        <head>
          <link rel="stylesheet"
          type="text/css"
          href=“first.css" />
        </head>
<body>
<h1>My CSS web page!</h1>
<p>Hello world! This is a CSS
   Example.</p>

</body>
</html>
```

Example: Inline Styles

```
<html>
   <body style="background-color:#b0c4de" >
      <h1>My CSS web page!</h1>
      <p>Hello world! This is a W3Schools.com example.</p>
   </body>
</html>
```

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 "#".
- The style rule below will be applied to the element with id="para1":

Example

- #para1
  {
  text-align:center;
  color:red;
  }
- <h1 id= “para1" > MBICT Engineering College</h1>

Class selector

- The class selector is used to specify a ***style for a group of elements***.
- Unlike the id selector, the class selector is most often used on several elements.
- This allows you to set a particular style for any HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."
- In the example below, all HTML elements with class="center" will be center-aligned:

Example

- .center {text-align:center;}

Class selector

- P.right {text-align:right}
- P.left{text-align:left}

- To actually use this within HTML document use class attribute:

 This para will be right aligned
 This para will be left aligned

To define a global style class omit the element name

.left{ text-align:left}

<h1 class="left">this heading will be left aligned</h1>
 This para will be left aligned

pseudo-class Selector

- CSS pseudo-classes are used to add special effects to some selectors.

Syntax

- The syntax of pseudo-classes:

selector:pseudo-class {property:value;}

- CSS classes can also be used with pseudo-classes:

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

CSS Background

- CSS background properties are used to define the background effects of an element.
- CSS properties used for background effects:
  - **background-color**
  - **background-image**
  - **background-repeat**
  - **background-attachment**
  - **background-position**

CSS Background

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:

Example

- body {background-color:#b0c4de;}

- The background color can be specified by:
  - **name** - a color name, like "red"
  - **RGB** - an RGB value, like "rgb(255,0,0)"
  - **Hex** - a hex value, like "#ff0000“

Example

- h1 {background-color:#6495ed;}
  p {background-color:#e0ffff;}
  div {background-color:#b0c4de;}

Background Image

- The background-image property specifies an image to use as the background of an element.
- By default, the image is repeated so it covers the entire element.
- The background image for a page can be set like this:

Example

```
body
{
        background-image:url('paper.gif');
}
```

Background Image - Repeat Horizontally or Vertically

background-repeat

- By default, the background-image property repeats an image both horizontally and vertically.
- Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Background Image - Repeat Horizontally or Vertically

Example

- body
  {
  background-image:url('gradient2.png');
  }

  If the image is repeated only horizontally (repeat-x), the background will look better:

Example

- body
  {
  background-image:url('gradient2.png');
  background-repeat:repeat-x;
  }

Background Image - Repeat Horizontally or Vertically

- **repeat** → The background image will be repeated both vertically and horizontally. This is default
- **repeat-x** → The background image will be repeated only horizontally
- **repeat-y** → The background image will be repeated only vertically
- **no-repeat** → The background-image will not be repeated

Background-attachment

- 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. This is default
- ***Fixed*** → The background image is fixed

Background-position

- The background-position property sets the starting position of a background image.
    1. **left top**
       left center
       left bottom
       right top
       right center
       right bottom
       center top
       center center
       center bottom
    2. **If you only specify one keyword, the second value will be "center"**

Background-position

- **x% y%**

  The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. . Default value is: 0% 0%

- ***xpos ypos***

  The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units. If you only specify one value, the other value will be 50%. You can mix % and positions

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":

Example

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

-When using the shorthand property the order of the property values are:

background-color
background-image
background-repeat
background-attachment
background-position

It does not matter if one of the property values is missing, as long as the ones that are present are in this order.

Property	Description	Values
background	Sets all the background properties in one declaration	*background-color* *background-image* *background-repeat* *background-attachment* *background-position* inherit
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed inherit
background-color	Sets the background color of an element	*color-rgb* *color-hex* *color-name* transparent inherit
background-image	Sets the background image for an element	url(*URL*) none inherit
background-position	Sets the starting position of a background image	left top left center left bottom right top right center right bottom center top center center center bottom *x% y%* *xpos ypos* inherit
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat inherit

Grouping

- If multiple selectors share the same CSS properties, they can be declared together. This is called "***grouping***".
- For example, if <h1>, <h2>, and <h3> share they same style, they can be declared together as follows:
- h1, h2, h3 {
  property:value;
  …
  }
- div,p
  {
  background-color:yellow;
  }

Descendant Selectors

- We can specify the style of an element only when it is inside another element. To accomplish this, we use the concept of descendant selectors.
- The syntax for declaring a descendant selector is:
- [Parent Selector] [Child Selector] {

  property:value;

  …

  }
- The style defined above will apply to child selectors only when they are inside the parent selector.
- Such declarations can go on for more than two levels.

Descendant Selectors

For example, in the declaration below,

- li b {
  color:yellow;
  }
- means that text in the <b> element inside the <li> element will be yellow.
- div p
  {
  background-color:yellow;
  }

Manipulating Text

Text Color

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

The default color for a page is defined in the body selector.

Text Alignment

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

- Text can be centered, or aligned to the left or right, or justified.

```
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
```

Direction

- Set the text direction to "right-to-left":
- div

  {

  direction:rtl;

  }
- Ltr→ The writing direction is left-to-right. This is default
- rtl→ The writing direction is right-to-left

letter-spacing

Example

- Set the letter spacing for h1 and h2 elements:
- h1 {letter-spacing:2px}
  h2 {letter-spacing:-3px}

line-height

- The line-height property specifies the line height.
- **Example**
- Set the line height in percent:
- p.small {line-height:90%}
  p.big {line-height:200%}

text-decoration

The text-decoration property specifies the decoration added to text.

- None → Defines a normal text. This is default
- underline →Defines a line below the text
- overline →Defines a line above the text
- line-through→ Defines a line through the text
- Blink→ Defines a blinking text

text-indent

- The text-indent property specifies the indentation of the first line in a text-block.
- Example:

  Indent the first line of paragraphs with 50 pixels:

- ```
  p
  {
        text-indent:50px;
  }
  ```
  ```

text-transform

- The text-transform property controls the capitalization of text.

Example

- Transform text in different elements:
- h1 {text-transform:uppercase}
  h2 {text-transform:capitalize}
  p {text-transform:lowercase}

vertical-align

- The vertical-align property sets the vertical alignment of an element.
- Vertical align an image:
- img

```
{
        vertical-align:text-top;
}
```

Table Properties in CSS

Table Color

```
<style type="text/css">
   table, td, th
   {
         border:1px solid green;
   }
   th
   {
         background-color:green;
         color:white;
   }
</style>
```

Table Properties in CSS

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;
}   or    padding-left:50px;
```

Table Properties in CSS

Table Borders

```
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 Properties in CSS

Table Width & Text Alignment

```
Table
{
        width:100%;
}
Th  { height:75px;  }
td
{
        text-align:right;
        Vertica l-align:bottom;
}
```

word-spacing

- The word-spacing property increases or decreases the white space between words.
- **Example**
- Specify that the space between words in paragraphs should be 30 pixels:
- p

```
{
        word-spacing:30px;
}
```

line-height

- p.small {line-height:90%}
  p.big {line-height:200%}

line-alignment

- Left→ Aligns the text to the left
- right →Aligns the text to the right
- Center→ Centers the text justify Stretches the lines so that each line has equal width (like in newspapers and magazines)

Font

font	Sets all the font properties in one declaration	*font-style* *font-variant* *font-weight* *font-size/line-height* *font-family* caption icon menu message-box small-caption status-bar inherit
font-family	Specifies the font family for text	*family-name* *generic-family* inherit
font-size	Specifies the font size of text	xx-small x-small small medium large x-large xx-large smaller larger *length* % inherit
font-style	Specifies the font style for text	normal italic oblique inherit
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal small-caps inherit
font-weight	Specifies the weight of a font	normal bold bolder

CSS Lists

Property	Description	Values
list-style	Sets all the properties for a list in one declaration	*list-style-type* *list-style-position* *list-style-image* inherit
list-style-image	Specifies an image as the list-item marker	URL none inherit
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow	inside outside inherit
list-style-type	Specifies the type of list-item marker	none disc circle square decimal decimal-leading-zero armenian georgian lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman inherit

CSS Links

Styling Links

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).
- Special for links are that they can be styled differently depending on what state they are in.
- The four links states are:

  a:link - a normal, unvisited link

  a:visited - a link the user has visited

  a:hover - a link when the user mouses over it

  a:active - a link the moment it is clicked

CSS Links

- **Example**
- a:link {color:#FF0000;} /* unvisited link */
  a:visited {color:#00FF00;} /* visited link */
  a:hover {color:#FF00FF;} /* mouse over link */
  a:active {color:#0000FF;} /* selected link */

```
<html>
<head>
<style type="text/css">
        a:link {color:#FF0000;}    /* unvisited link */
        a:visited {color:#00FF00;} /* visited link */
        a:hover {color:#FF00FF;}   /* mouse over link */
        a:active {color:#0000FF;}  /* selected link */
</style>
</head>

<body>
<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>
</body>
</html>
```

```
<html>
<head>
<style type="text/css">
a:link,a:visited
{
    display: block;
    font-weight:bold;
    color:#FFFFFF;
    background-color:#98bf21;
    width:120px;
    text-align:center;
    padding:4px;
    text-decoration:none;
}
a:hover,a:active
{
   background-color:#7A991A;
}
</style>
</head>

<body>
<a href="default.asp" target="_blank">This is a link</a>
</body>
</html>
```

adlink.html

```
<html>
<head>
<style type="text/css">
a.one:link {color:#ff0000;}
a.one:visited {color:#0000ff;}
a.one:hover {color:#ffcc00;}

a.two:link {color:#ff0000;}
a.two:visited {color:#0000ff;}
a.two:hover {font-size:150%;}

a.three:link {color:#ff0000;}
a.three:visited {color:#0000ff;}
a.three:hover {background:#66ff66;}

a.four:link {color:#ff0000;}
a.four:visited {color:#0000ff;}
a.four:hover {font-
family:monospace;}

a.five:link {color:#ff0000;text-
decoration:none;}
a.five:visited {color:#0000ff;text-
decoration:none;}
a.five:hover {text-
decoration:underline;}
</style>
</head>
```

```
<body>
<p>Mouse over the links to see them change layout.</p>


<p><b><a class="one" href="default.asp" target="_blank">This link changes
   color</a></b></p>
<p><b><a class="two" href="default.asp" target="_blank">This link changes font-
   size</a></b></p>
<p><b><a class="three" href="default.asp" target="_blank">This link changes
   background-color</a></b></p>
<p><b><a class="four" href="default.asp" target="_blank">This link changes font-
   family</a></b></p>
<p><b><a class="five" href="default.asp" target="_blank">This link changes text-
   decoration</a></b></p>
</body>


</html>
```

view

CSS Box Model

CSS Box Model

- 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.

```
width:250px;
padding:10px;
border:5px solid gray;
margin:10px;
```

Margin Properties

Property	Description
margin	Sets all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element

Padding Properties

Property	Description
padding	Sets all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

Border and Outline Properties

Property	Description
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders
outline	Sets all the outline properties in one declaration
outline-color	Sets the color of an outline
outline-style	Sets the style of an outline
outline-width	Sets the width of an outline

Border style

Property Values

Value	Description
none	Specifies no border
hidden	The same as "none", except in border conflict resolution for table elements
dotted	Specifies a dotted border
dashed	Specifies a dashed border
solid	Specifies a solid border
double	Specifies a double border
groove	Specifies a 3D grooved border. The effect depends on the border-color value
ridge	Specifies a 3D ridged border. The effect depends on the border-color value
inset	Specifies a 3D inset border. The effect depends on the border-color value
outset	Specifies a 3D outset border. The effect depends on the border-color value
inherit	Specifies that the border style should be inherited from the parent element

```
<html>
<head>
<style type="text/css">
p {border-style:solid;}
p.none {border-bottom-style:none;}
p.dotted {border-bottom-style:dotted;}
p.dashed {border-bottom-style:dashed;}
p.solid {border-bottom-style:solid;}
p.double {border-bottom-style:double;}
p.groove {border-bottom-style:groove;}
p.ridge {border-bottom-style:ridge;}
p.inset {border-bottom-style:inset;}
p.outset {border-bottom-style:outset;}
</style>
</head>
<body>
<p class="none">No bottom border.</p>
<p class="dotted">A dotted bottom
border.</p>
<p class="dashed">A dashed bottom
border.</p>
<p class="solid">A solid bottom border.</p>
<p class="double">A double bottom
border.</p>
<p class="groove">A groove bottom
border.</p>
<p class="ridge">A ridge bottom border.</p>
<p class="inset">An inset bottom border.</p>
<p class="outset">An outset bottom
border.</p>
</body>
</html>
```

view

Border width property

- The border-bottom-width property sets the width of an element's bottom border. Borderwidth.html

Property Values

Value	Description
thin	Specifies a thin bottom border
medium	Specifies a medium bottom border. This is default
thick	Specifies a thick bottom border
length	Allows you to define the thickness of the bottom border
inherit	Specifies that the border width should be inherited from the parent element

outline-style Property

- An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".
- view

Property Values

Value	Description
none	Specifies no outline
dotted	Specifies a dotted outline
dashed	Specifies a dashed outline
solid	Specifies a solid outline
double	Specifies a double outliner
groove	Specifies a 3D grooved outline. The effect depends on the outline-color value
ridge	Specifies a 3D ridged outline. The effect depends on the outline-color value
inset	Specifies a 3D inset outline. The effect depends on the outline-color value
outset	Specifies a 3D outset outline. The effect depends on the outline-color value
inherit	Specifies that the outline style should be inherited from the parent element

outline-style Property

- The display property specifies if/how an element is displayed, and the visibility property specifies if an element should be visible or hidden.

Hiding an Element - display:none or visibility:hidden

- Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:
- visibility:hidden hides an element, but it will still take up the same space as before. The element will be hidden, but still affect the layout.
- **Example**
- h1.hidden {visibility:hidden;}

- display:none hides an element, and it will not take up any space. The element will be hidden, and the page will be displayed as the element is not there:
- **Example**
- h1.hidden {display:none;}

```
<html>
<head>
<style type="text/css">
h1.hidden {display:none;}
h2.hidden {visibility:hidden;}
</style>
</head>

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

<h2>This is a visible heading</h2>
<h2 class="hidden">This is a hidden heading</h2>
<p>Notice that the hidden heading still takes up space.</p>
</body>

</html>
```

Display.html

CSS Positioning

- **positioning**
  - The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.
  - Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.
  - There are four different positioning methods.

CSS Positioning

There are four different positioning methods.

- **Static Positioning**
  - HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.
  - Static positioned elements are not affected by the top, bottom, left, and right properties.

- **Fixed Positioning**
  - An element with fixed position is positioned relative to the browser window.
  - It will not move even if the window is scrolled:

CSS Positioning

Example

- P

```
{
        position:fixed;
        top:30px;
        right:5px;
}
```

CSS Positioning

- **Relative Positioning**
  - A relative positioned element is positioned relative to its normal position

  - h2.pos_left
    {
    position:relative;
    left:-20px;
    }
    h2.pos_right
    {
    position:relative;
    left:20px;
    }

CSS Positioning

- **Absolute Positioning**
  - An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>: position

  - h2
    {
    position:absolute;
    left:100px;
    top:150px;
    }

CSS Positioning

- **Overlapping Elements**
  - When elements are positioned outside the normal flow, they can overlap other elements.
  - The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
  - An element can have a positive or negative stack order:

```
img
{
    position:absolute;
    left:0px;
    top:0px;
    z-index:-1
}
```

CSS Positioning

CSS Float

- With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.
- Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.
- The elements after the floating element will flow around it.
- If an image is floated to the right, a following text flows around it, to the left:

```
img {   float:right;  }
```

CSS Positioning

Floating Elements Next to Each Other

- If you place several floating elements after each other, they will float next to each other if there is room.

```
.thumbnail
{
    float:left;
    width:110px;
    height:90px;
    margin:5px;
}
```

CSS Positioning

No Floating

- The clear property specifies which sides of an element other floating elements are not allowed.
- Add a text line into the image gallery, using the clear property:

```
.text_line
{
        clear:both;
}
```

CSS Layout and Structure

- The idea behind CSS was to separate the formatting and styling rules from the content.
- Structure of document can be maintain by breaking the page into logical sections with div elements.

CSS Navigation Bars

- With CSS you can transform boring HTML menus into good-looking navigation bars.

Navigation Bar = List of Links

- A navigation bar needs standard HTML as a base.
- A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense

CSS Layout and Structure

Vertical Navigation Bar

- To build a vertical navigation bar we only need to style the <a> elements in addition to the list code.

```
a
{
        display:block;
        width:60px;
}
```

- ***display:block*** - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width

CSS Layout and Structure

Horizontal Navigation Bar

- There are two ways to create a horizontal navigation bar.
- Using **inline** or **floating** list items.
- if you want the links to be the same size, you have to use the floating method.

CSS Layout and Structure

Inline List Items

- To build a horizontal navigation bar is to specify the <li> elements as inline

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

- ***display:inline***- By default, <li> elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line

CSS Layout and Structure

Floating List Items

- For all the links to have an equal width, float the <li> elements and specify a width for the <a> elements
- li

  { float:left; }

  a

  { display:block; width:60px; }
- ***float:left*** - use float to get block elements to slide next to each other
- ***display:block*** - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width