UNIT 3
(CO 3)

OVERVIEW

- Review the history and concepts of CSS
- Explore inline styles, embedded styles, & external style sheets and understand style precedence and style inheritance
- Understand the use of color in CSS, and explore CSS styles for fonts and text
- Review and compare different image formats and learn to display an animated graphic
- Apply a background image to an element and float elements on a Web page
- Apply border styles to an element

Cascading Style Sheets (CSS)
(CO 3)

Objective: To understand CSS

Cascading **S**tyle **S**heets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages.

More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

CSS is used to control the style of a web document in a simple and easy way.

Cascading Style Sheets (CSS)
(CO 3)

Why to Learn CSS?

Cascading Style Sheets, fondly referred to as **CSS**, is a simple design language intended to simplify the process of making web pages presentable.

CSS is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Some of the key advantages of learning CSS:

- **Create Stunning Web site** - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

Cascading Style Sheets (CSS)
(CO 3)

- **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- **Control web** - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- **Learn other languages** - Once you understands the basic of HTML and CSS then other related technologies like javascript, php, or angular are become easier to understand.

Cascading Style Sheets (CSS)
(CO 3)

Advantages of CSS

- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Cascading Style Sheets (CSS)
(CO 3)

- **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

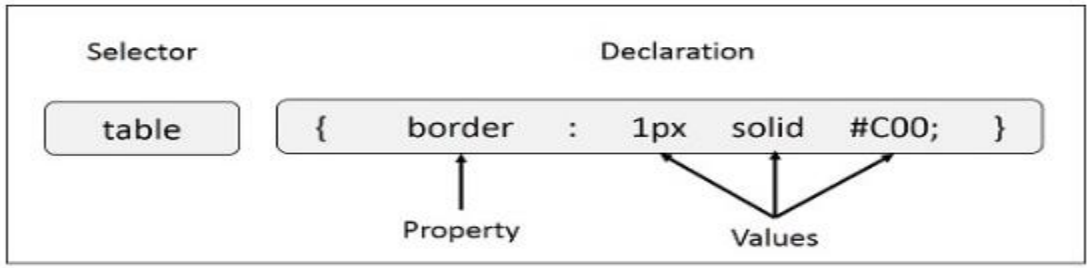
CSS – Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

- **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows −

selector { property: value }



Cascading Style Sheets (CSS)
(CO 3)

- **Selector => h1 Declaration => {color:blue;font size:12px;}**
  - 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.For Example:
    -> color is property and blue is value.
    -> font size is property and 12px is value.
  - A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Example :

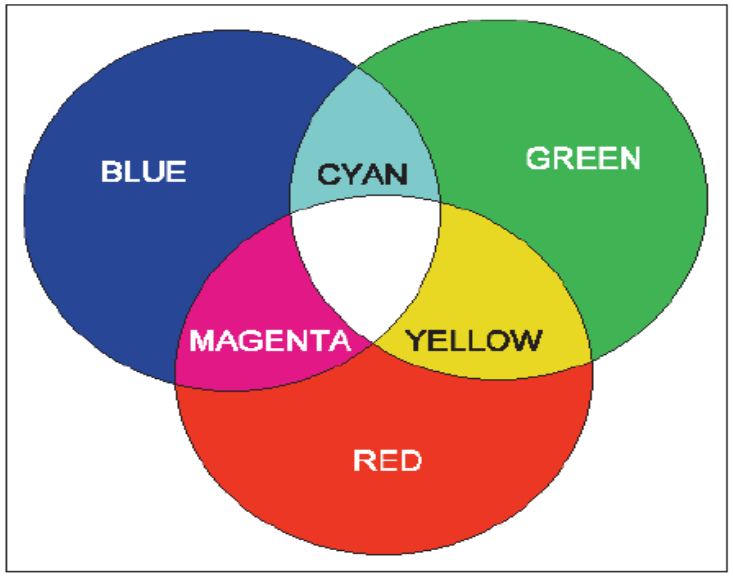
In the following example all p elements will be center-aligned, with a blue text color:

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

CSS Colors
(CO 3)

Objective: To understand CSS Colors

Standard	Hexadecimal
• White	• #ffffff
• Black	• #fff
• Blue	• #cccf0f3
• Fuchsia	
• Gray	
• Green	



Color Name	RGB Triplet	Hexadecimal	Color Name	RGB Triplet	Hexadecimal
Aqua	(0, 255, 255)	00FFFF	Olive	(128, 128, 0)	808000
Black	(0, 0, 0)	000000	Orange	(255, 165, 0)	FFA500
Blue	(0, 0, 255)	0000FF	Purple	(128, 0, 128)	800080
Fuchsia	(255, 0, 255)	FF00FF	Red	(255, 0, 0)	FF0000
Gray	(128, 128, 128)	808080	Silver	(192, 192, 192)	C0C0C0
Green	(0, 128, 0)	008000	Teal	(0, 128, 128)	008080
Lime	(0, 255, 0)	00FF00	White	(255, 255, 255)	FFFFFF
Maroon	(128, 0, 0)	800000	Yellow	(255, 255, 0)	FFFF00
Navy	(0, 0, 128)	000080			

Cascading Style Sheets (CSS)
(CO 3)

We can define selectors in various simple ways based on your comfort.

The Type Selectors

- This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings −
- h1 { color: Red }

The Universal Selectors

- Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type −
- * { color: #000000; }This rule renders the content of every element in our document in black.

The Descendant Selectors

- Suppose you want to apply a style rule to a particular element only when it lies inside a particular element.

  As given in the following example, style rule will apply to element only when it lies inside <ul> tag.

  ul em

  { color: Red}

  The tag is used to define emphasized text. The content inside is typically displayed in *italic*.

```
<!DOCTYPE html>
<html>
<body>
<h1>The em element</h1>
<p>You <em>have</em> to hurry up!</p>
```

The Class Selectors

- You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

.black

{ color: black}

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example −

h1.black

{ color: black }

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

- You can apply more than one class selectors to given element. Consider the following example −

This para will be styled by the classes *center* and *bold*.

Cascading Style Sheets (CSS)
(CO 3)

THE ID SELECTOR :

The id selector uses the id attribute of an HTML element to select a specific element.

- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id=”para1”:

```
#para1 {
        text-align: center;
        color: red;
}
```

Cascading Style Sheets (CSS)
(CO 3)

The true power of id selectors is when they are used as the foundation for descendant selectors, For example:

```
#black h2 {
        color: #000000;
    }
```

- In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having id attribute set to black.

NOTE: An id name cannot start with a number.

The Child Selectors

We have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example −

```
body > p
    { color: #000000; }
```

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like or <td> would not have any effect of this rule.

```
ul em
    { color: Red}
```

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example:

```
h1
  { color: red;
    font-weight: normal;
    letter-spacing: 4px;
    margin-bottom: 1px;
    text-transform: lowercase; }
```

Here all the property and value pairs are separated by a **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

Grouping Selectors

- You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example –

```
h1, h2, h3
   { color: #36C;
     font-weight: normal;
     letter-spacing: 4px;
     margin-bottom: 1px;
     text-transform: lowercase; }
```

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

Cascading Style Sheets (CSS)
(CO 3)

You can combine the various *id* selectors together as shown below −

```
#content, #footer, #supplement
    { position: center;
      left: 510px;
      width: 200px; }
```

Cascading Style Sheets (CSS)
(CO 3)

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>

</body>
</html>
```

Cascading Style Sheets (CSS)
(CO 3)

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
    background-color: yellow;
}
</style>
</head>
<body>
<h1>Welcome to My Homepage</h1>
 <p >My name is Donald.</p>
 <p >I live in Duckburg.</p>
 <p>My best friend is Mickey.</p>
</body>
</html>
```

Cascading Style Sheets (CSS)
(CO 3)

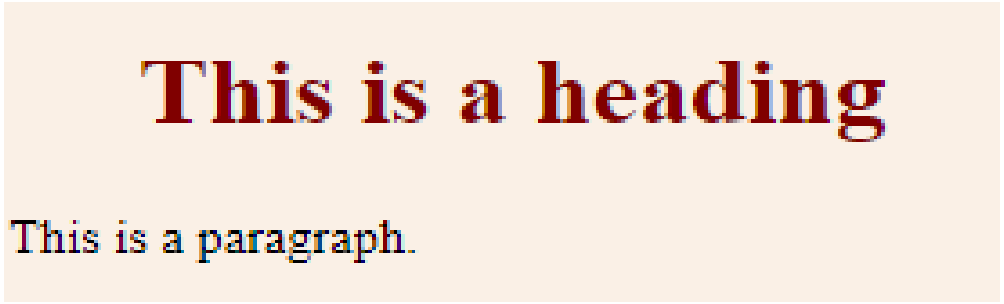
There are three ways to associate styles with your HTML document. Most commonly used methods are inline CSS and External CSS.

Embedded CSS - The <style> Element

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document.

Cascading Style Sheets (CSS)
(CO 3)

```
<!DOCTYPE html>
<html>
  <head>
    <style type = "text/css" media = "all">
      body {
        background-color: linen;
      }
      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



Cascading Style Sheets (CSS)
(CO 3)

Attributes

- Attributes associated with <style> elements are −

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This is required attribute.
media	screen tty tv projection handheld print braille aural all	Specifies the device the document will be displayed on. Default value is *all*. This is an optional attribute.

Cascading Style Sheets (CSS)
(CO 3)

Inline CSS - The *style* Attribute

- You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax

<element style = "...style rules...."> Attributes

Attribute	Value	Description
style	style rules	The value of *style* attribute is a combination of style declarations separated by semicolon (;).

Cascading Style Sheets (CSS)
(CO 3)

```
<html>
  <head>
  </head>

  <body>
    <h1 style = "color:blue">
      This is inline CSS
    </h1>
  </body>
</html>
```

This is inline CSS

Cascading Style Sheets (CSS)
(CO 3)

External CSS - The <link> Element

- The <link> element can be used to include an external stylesheet file in your HTML document.
- An external style sheet is a separate text file with **.css** extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.
- Here is the generic syntax of including external CSS file −

```
<head>
    <link type = "text/css" href = "..." media = "..." />
</head>
```

Cascading Style Sheets (CSS)
(CO 3)

Attributes

Attributes associated with <style> elements are –

Attribute	Value	Description
type	text css	Specifies the style sheet language as a content-type (MIME type). This attribute is required.
href	URL	Specifies the style sheet file having Style rules. This attribute is a required.
media	screen tty tv projection handheld print braille aural all	Specifies the device the document will be displayed on. Default value is *all*. This is optional attribute.

Cascading Style Sheets (CSS)
(CO 3)

Example

Consider a simple style sheet file with a name mystyle.css having the following rules −

```
h1, h2, h3 {
   color: blue;
   font-weight: normal;
   letter-spacing: .4em;
   margin-bottom: 1em;
   text-transform: lowercase;
}
```

Now you can include this file *mystyle.css* in any HTML document as follows −

```
<head>
      <link type = "text/css" href = "mystyle.css" media = " all" />
 </head>
```

Cascading Style Sheets (CSS)
(CO 3)

styles.css

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

```
<!DOCTYPE html>
<html>
<head>
 <link rel="stylesheet" href="styles.css">
</head>
<body>

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

Cascading Style Sheets (CSS)
(CO 3)

Inline CSS Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>Inline CSS</title>
   </head>

   <body>
      <p style = "color:#009900; font-size:50px;
            font-style:italic; text-align:center;">
         WELCOME
      </p>
   </body>
</html>
```

Embedded CSS Example

```
<!DOCTYPE html>
<html>
   <head>
      <title>Embedded CSS</title>
      <style>
         .main {
            text-align:center;
         }
         .GFG {
            color:#009900;
            font-size:50px;
            font-weight:bold;
         }
         .wd {
            font-style:bold;
            font-size:20px;
         }
```

Cascading Style Sheets (CSS)
(CO 3)

External CSS Example

```
body {
   background-color: powderblue;
}
.main {
   text-align:center;
}
.GFG {
   color:#009900;
   font-size:50px;
   font-weight:bold;
}
#wd {
   font-style:bold;
   font-size:20px;
}
```

save with mystyle.css

Cascading Style Sheets (CSS)
(CO 3)

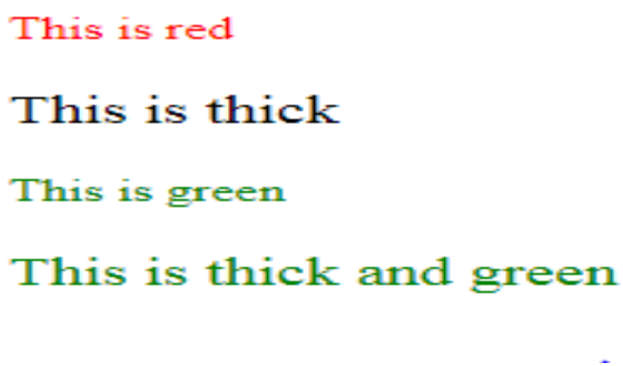
External CSS Example

```
<!DOCTYPE html>
<html>
   <head>
      <link rel="stylesheet" href="mystyle.css"/>
   </head>

   <body>
      <div class = "main">
         <div class ="GFG">Elective subject</div>
         <div id ="wd">
            Web Designing
         </div>
      </div>
   </body>
</html>
```

Cascading Style Sheets (CSS)
(CO 3)

External CSS Example

```
.red {
  color: red;
}
.thick {
  font-size:20px;
}
.green {
  color:green;
}
```




```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML External CSS</title>
    <link rel = "stylesheet" type = "text/css" href = "s.css">
  </head>
  <body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
  </body>
</html>
```

Cascading Style Sheets (CSS)
(CO 3)

Internal CSS Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Internal CSS</title>

    <style type = "text/css">
      .red {
        color: red;
      }
      .thick{
        font-size:20px;
      }
      .green {
        color:green;
      }
    </style>
  </head>

  <body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
  </body>
```

This is red

This is thick

This is green

This is thick and green

Cascading Style Sheets (CSS)
(CO 3)

Inline CSS Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Inline CSS</title>
  </head>

  <body>
    <p style = "color:red;">This is red</p>
    <p style = "font-size:20px;">This is thick</p>
    <p style = "color:green;">This is green</p>
    <p style = "color:green;font-size:20px;">This is thick and green</p>
  </body>

</html>
```

This is red

This is thick

This is green

This is thick and green

Cascading Style Sheets (CSS)
(CO 3)

Properties of CSS:

Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.

- As Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.
- Internal or Embedded stands second in the priority list and overrides the styles in the external style sheet.
- External style sheets have the least priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

Cascading Style Sheets (CSS)
(CO 3)

CSS Comments

- Many times, you may need to put additional comments in your style sheet blocks. So, it is very easy to comment any part in style sheet. You can simple put your comments inside /*.....this is a comment in style sheet.....*/.
- You can use /* ....*/ to comment multi-line blocks in similar way you do in C and C++ programming languages.

Cascading Style Sheets (CSS)
(CO 3)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        color: red;
        /* This is a single-line comment */
        text-align: center;
      }
      /* This is a multi-line comment */
    </style>
  </head>

  <body>
    <p>Hello World!</p>
  </body>
</html>
```

CSS - Measurement Units

CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. You need these values while specifying various measurements in your Style rules
e.g. **border = "1px solid red"**.

Cascading Style Sheets (CSS)
(CO 3)

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 16pt; line-height: 125%;}
cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}
em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 7em;}
ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {font-size: 24pt; line-height: 3ex;}
in	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 20pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

CSS - Colors

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element. They can also be used to affect the color of borders and other decorative effects.

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

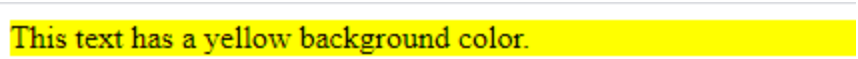
Cascading Style Sheets (CSS)
(CO 3)

CSS – Backgrounds

- the following are background properties of an element –
- The **background-color** property is used to set the background color of an element.
- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The **background** property is used as a shorthand to specify a number of other background properties.

Set the Background Color

```
<html>
  <head>
  </head>

  <body>
    <p style = "background-color:yellow;">
      This text has a yellow background color.
    </p>
  </body>
</html>
```



Set the Background Image

- We can set the background image by calling local stored images as shown below

```
<html>
  <head>
    <style>
      body  {
        background-image: url("pic.jpeg");
        background-color: #cccccc;
      }
    </style>
  </head>

  <body>
    <h1>Hello World!</h1>
  </body>
<html>
```

Repeat the Background Image

- How to repeat the background image if an image is small. You can use *no-repeat* value for *background-repeat* property if you don't want to repeat an image, in this case image will display only once.
- By default *background-repeat* property will have *repeat* value.

```
<html>
  <head>
    <style>
      body {
        background-image: url("pic.jpeg");
        background-repeat: no-repeat;
      }
    </style>
  </head>

  <body>
    <p>Welcome</p>
  </body>
</html>
```

to repeat the background image vertically: repeat-y

to repeat the background image horizontally: repeat-x

Set the Background Image Position

The following example demonstrates how to set the background image position 100 pixels away from the left side.

```
<html>
  <head>
    <style>
      body {
        background-image: url("pic.jpeg");
        background-repeat: no-repeat;
        background-position:100px;
      }
    </style>
  </head>

  <body>
    <p>Tutorials point</p>
  </body>
</html>
```

Cascading Style Sheets (CSS)
(CO 3)

The following example demonstrates how to set the background image position 100 pixels away from the left side and 200 pixels down from the top.

```
<html>
  <head>
    <style>
      body {
        background-image: url("pic.jpeg");
        background-repeat: no-repeat;
        background-position:100px 200px;
      }
    </style>
  </head>

  <body>
    <p>Tutorials point</p>
  </body>
</html>
```

Set the Background Attachment

Background attachment determines whether a background image is fixed or scrolls with the rest of the page.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-image: url('pic.jpeg');
        background-repeat: no-repeat;
        background-attachment: fixed;
      }
    </style>
  </head>

  <body>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
   </body>
</html>
```

demonstration how to set the scrolling background image.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-image: url('pic.jpeg');
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-attachment:scroll;
      }
    </style>
  </head>

  <body>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>
    <p>The background-image is fixed. Try to scroll down the page.</p>

  </body>
</html>
```

CSS – Fonts

- You can set following font properties of an element −
- The **font-family** property is used to change the face of a font.
- The **font-style** property is used to make a font italic or oblique.
- The **font-variant** property is used to create a small-caps effect.
- The **font-weight** property is used to increase or decrease how bold or light a font appears.
- The **font-size** property is used to increase or decrease the size of a font.
- The **font** property is used as shorthand to specify a number of other font properties.

Set the Font Family

how to set the font family of an element. Possible value could be any font family name.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-family:georgia,garamond,serif;">
      This text is rendered in either georgia, garamond, or the
      default serif font depending on which font  you have at your system.
    </p>
  </body>
</html>
```

Set the Font Style

Demonstration of how to set the font style of an element. Possible values are *normal, italic and oblique*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-style:italic;">
      This text will be rendered in italic style
    </p>
  </body>
</html>
```

Set the Font Variant

Demonstration of how to set the font variant of an element. Possible values are *normal and small-caps*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-variant:small-caps;">
      This text will be rendered as small caps
    </p>
  </body>
</html>
```

Set the Font Weight

Demonstration of how to set the font weight of an element. The font-weight property provides the functionality to specify how bold a font is. Possible values could be *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-weight:bold;">
      This font is bold.
    </p>

    <p style = "font-weight:bolder;">
      This font is bolder.
    </p>

    <p style = "font-weight:500;">
      This font is 500 weight.
    </p>
  </body>
</html>
```

Set the Font Size

Demonstration of how to set the font size of an element. The font-size property is used to control the size of fonts. Possible values could be *xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font-size:20px;">
      This font size is 20 pixels
    </p>

    <p style = "font-size:small;">
      This font size is small
    </p>

    <p style = "font-size:large;">
      This font size is large
    </p>
  </body>
</html>
```

This font size is 20 pixels

This font size is small

This font size is large

Set the Font Size Adjust

Demonstrates how to set the font size adjust of an element. This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

```
<html>
 <head>
 </head>

 <body>
   <p style = "font-size-adjust:0.61;">
     This text is using a font-size-adjust value.
   </p>
 </body>
</html>
```

Shorthand Property

You can use the *font* property to set all the font properties at once.

```
<html>
  <head>
  </head>

  <body>
    <p style = "font:italic small-caps bold 15px georgia;">
      Applying all the properties on the text at once.
    </p>
  </body>
</html>
```

APPLYING ALL THE PROPERTIES ON THE TEXT AT ONCE.

CSS – Text

- You can set following text properties of an element −
- The **color** property is used to set the color of a text.
- The **direction** property is used to set the text direction.
- The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
- The **word-spacing** property is used to add or subtract space between the words of a sentence.
- The **text-indent** property is used to indent the text of a paragraph.
- The **text-align** property is used to align the text of a document.
- The **text-decoration** property is used to underline, overline, and strikethrough text.
- The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
- The **white-space** property is used to control the flow and formatting of text.
- The **text-shadow** property is used to set the text shadow around a text.

Set the Text Color

The following example demonstrates how to set the text color. Possible value could be any color name in any valid format.

```
<html>
  <head>
  </head>

  <body>
    <p style = "color:red;">
      This text will be written in red.
    </p>
  </body>
</html>
```

This text will be written in red.

Set the Text Direction

The following example demonstrates how to set the direction of a text. Possible values are *ltr or rtl*.

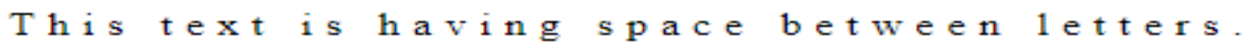
```
<html>
  <head>
  </head>

  <body>
    <p style = "direction:rtl;">
      This text will be rendered from right to left
    </p>
  </body>
</html>
```

This text will be rendered from right to left

Set the Space between Characters

The following example demonstrates how to set the space between characters. Possible values are *normal or a number specifying space.*

```
<html>
  <head>
  </head>

  <body>
    <p style = "letter-spacing:5px;">
      This text is having space between letters.
    </p>
  </body>
</html>
```



Set the Space between Words

- Possible values are *normal or a number specifying space*.

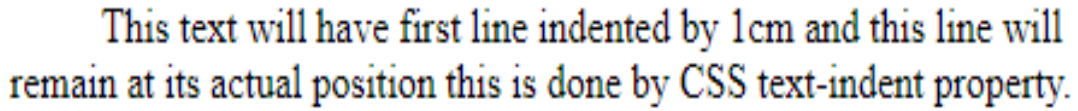
```
<html>
  <head>
  </head>

  <body>
    <p style = "word-spacing:10px;">
      This text is having space between words.
    </p>
  </body>
</html>
```

This text is having space between words.

Set the Text Indent

Indent the first line of a paragraph. Possible values are *% or a number specifying indent space*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "text-indent:1cm;">
      This text will have first line indented by 1cm and this line will remain at
      its actual position this is done by CSS text-indent property.
    </p>
  </body>
</html>
```

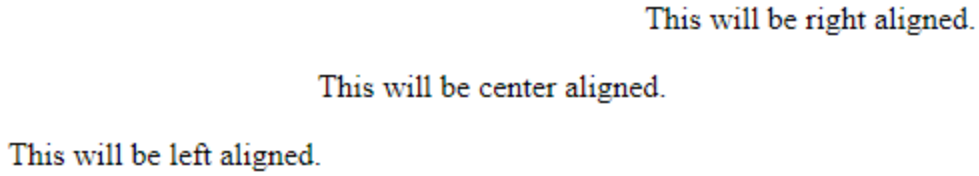


Cascading Style Sheets (CSS)
(CO 3)

Set the Text Alignment

Possible values are *left, right, center, justify*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "text-align:right;">
      This will be right aligned.
    </p>

    <p style = "text-align:center;">
      This will be center aligned.
    </p>

    <p style = "text-align:left;">
      This will be left aligned.
    </p>
  </body>
</html>
```



Decorating the Text

Possible values are *none, underline, overline, line-through, blink*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "text-decoration:underline;">
      This will be underlined
    </p>

    <p style = "text-decoration:line-through;">
      This will be striked through.
    </p>

    <p style = "text-decoration:overline;">
      This will have a over line.
    </p>

    <p style = "text-decoration:blink;">
      This text will have blinking effect
    </p>
  </body>
</html>
```

This will be underlined

~~This will be striked through.~~

This will have a over line.

This text will have blinking effect

Set the Text Cases

Possible values are *none, capitalize, uppercase, lowercase*.

```
<html>
  <head>
  </head>

  <body>
    <p style = "text-transform:capitalize;">
      This will be capitalized
    </p>

    <p style = "text-transform:uppercase;">
      This will be in uppercase
    </p>

    <p style = "text-transform:lowercase;">
      This will be in lowercase
    </p>
  </body>
</html>
```

This Will Be Capitalized

THIS WILL BE IN UPPERCASE

this will be in lowercase

CSS - Tables

You can set following properties of a table −

- The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The **border-spacing** specifies the width that should appear between table cells.
- The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
- The **empty-cells** specifies whether the border should be shown if a cell is empty.
- The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

The border-collapse Property

- This property can have two values *collapse* and *separate*.

```
<html>
  <head>
    <style type = "text/css">
      table.one {border-collapse:collapse;}
      table.two {border-collapse:separate;}

      td.a {
        border-style:dotted;
        border-width:3px;
        border-color:#000000;
        padding: 10px;
      }
      td.b {
        border-style:solid;
        border-width:3px;
        border-color:#333333;
        padding:10px;
      }
    </style>
  </head>

  <body>
    <table class = "one">
```

Collapse Border Example

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example

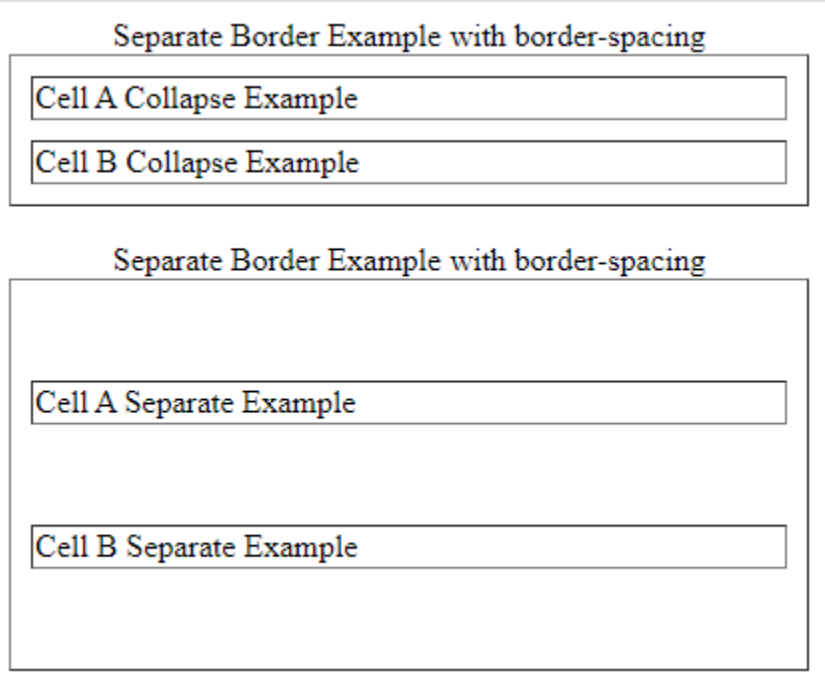
Cell A Separate Example

Cell B Separate Example

The border-spacing Property

- The border-spacing property specifies the distance that separates adjacent cells'. borders. It can take either one or two values; these should be units of length.
- If you provide one value, it will applies to both vertical and horizontal borders. Or you can specify two values, in which case, the first refers to the horizontal spacing and the second to the vertical spacing

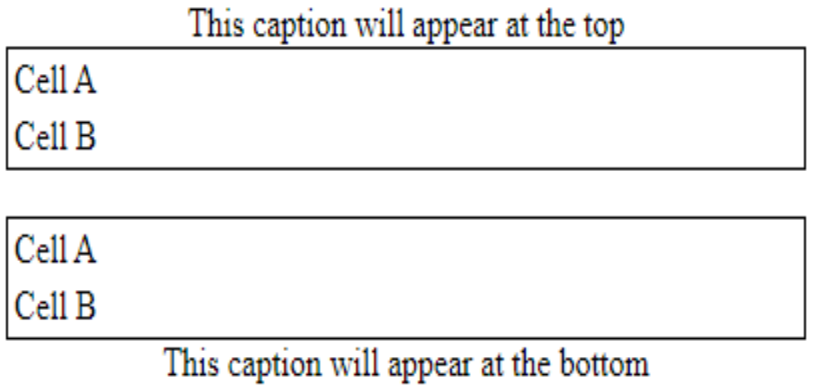
```
<html>
  <head>
    <style type = "text/css">
      table.one {
        border-collapse:separate;
        width:400px;
        border-spacing:10px;
      }
      table.two {
        border-collapse:separate;
        width:400px;
        border-spacing:10px 50px;
      }
    </style>
```



The caption-side Property

- The caption-side property allows you to specify where the content of a <caption> element should be placed in relationship to the table. This property can have one of the four values *top, bottom, left* or *right*.

```
<html>
  <head>
    <style type = "text/css">
      caption.top {caption-side:top}
      caption.bottom {caption-side:bottom}
    </style>
  </head>

  <body>

    <table style = "width:400px; border:1px solid black;">
      <caption class = "top">
        This caption will appear at the top
      </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr>
    </table>
    <br />
```



The empty-cells Property

- The empty-cells property indicates whether a cell without any content should have a border displayed.
- This property can have one of the two values - *show, hide*.

```
<html>
  <head>
    <style type = "text/css">
      table.empty {
        width:350px;
        border-collapse:separate;
        empty-cells:hide;
      }
      td.empty {
        padding:5px;
        border-style:solid;
        border-width:1px;
        border-color:#999999;
      }
    </style>
  </head>

  <body>

    <table class = "empty">
```

	Title one	Title two
Row Title	value	value
Row Title	value	

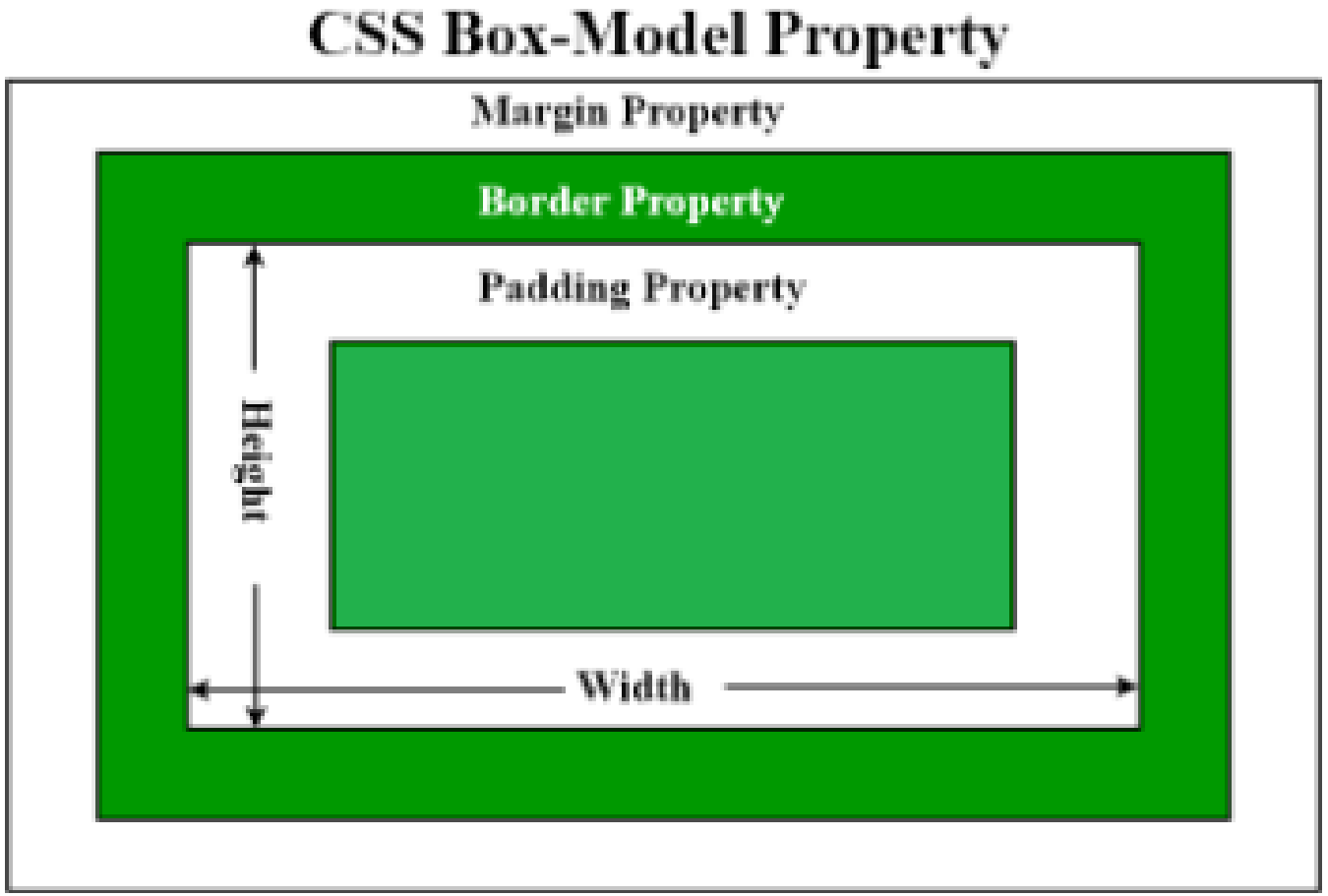
Box Model(CSS)
(CO 3)

Box Model

- CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.
  Box-Model has multiple properties in CSS. Some of them are given below:
- borders
- margins
- padding
- Content

Box Model(CSS)
(CO 3)

Box Model

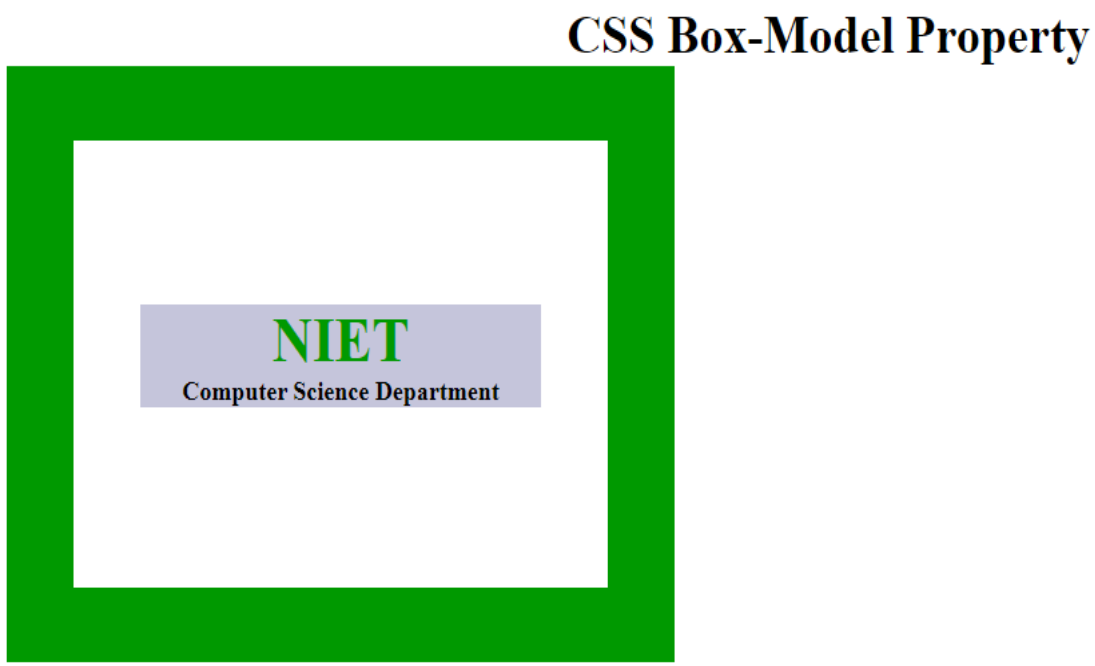


Box Model(CSS)
(CO 3)

- **Border Area:** It is the area between the box's padding and margin. Its dimensions are given by the width and height of border.
- **Margin Area:** This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.
- **Padding Area:** It includes the element's padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.
- **Content Area:** This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

Box Model(CSS)
(CO 3)

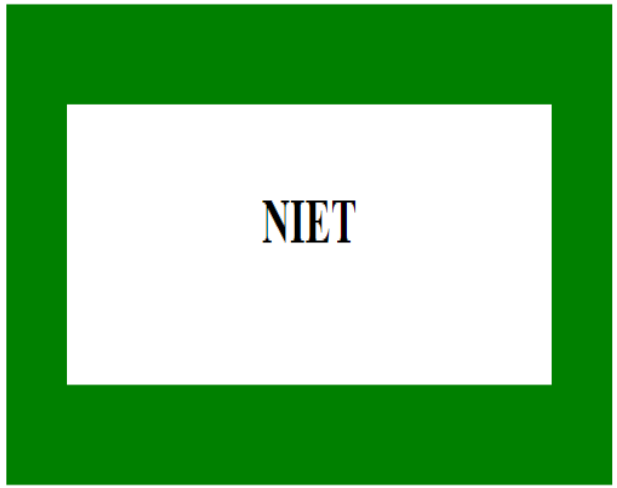
```
<!DOCTYPE html>
  <head>
    <title>CSS Box Model</title>
    <style>
      .main {
        font-size:36px;
        font-weight:bold;
        Text-align:center;
      }
      .gfg {
        margin-left:60px;
        border:50px solid #009900;
        width:300px;
        height:200px;
        text-align:center;
        padding:50px;
      }
      .gfg1 {
        font-size:42px;
        font-weight:bold;
        color:#009900;
        margin-top:60px;
        background-color:#c5c5db;
      }
      .gfg2 {
        font-size:18px;
        font-weight:bold;
        background-color:#c5c5db;
```



Box Model(CSS)
(CO 3)

```
<!DOCTYPE html>
  <head>
    <style>
    .main {
      font-size:32px;
      font-weight:bold;
      text-align:center;
    }
    #box {
      padding-top:40px;
      width: 400px;
      height: 100px;
      border: 50px solid green;
      margin: 50px;
      text-align:center;
      font-size:32px;
      font-weight:bold;
    }
    </style>
  </head>

  <body>
    <div class="main">CSS Box-Model Property</div>
    <div id="box">NIET</div>
  </body>
</html>
```

CSS Box-Model Property



CSS – Dimension
(CO 3)

We have the following properties that allow you to control the dimensions of a box.

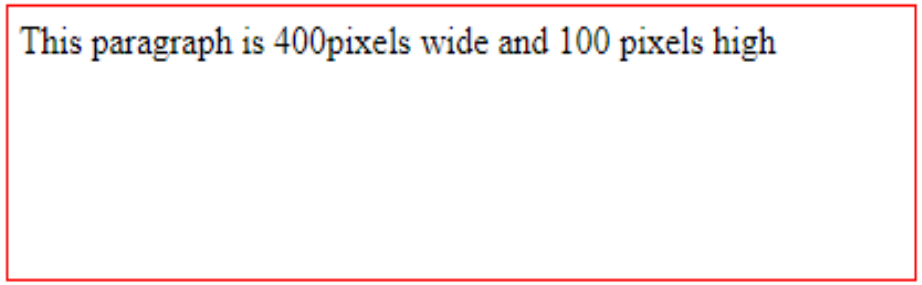
- The **height** property is used to set the height of a box.
- The **width** property is used to set the width of a box.
- The **line-height** property is used to set the height of a line of text.
- The **max-height** property is used to set a maximum height that a box can be.
- The **min-height** property is used to set the minimum height that a box can be.
- The **max-width** property is used to set the maximum width that a box can be.
- The **min-width** property is used to set the minimum width that a box can be.

CSS – Dimension
(CO 3)

The Height and Width Properties

The *height* and *width* properties allow you to set the height and width for boxes. They can take values of a length, a percentage, or the keyword auto.

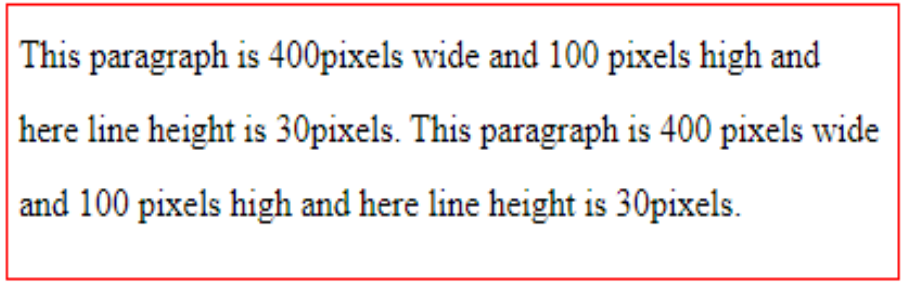
```
<html>
  <head>
  </head>

  <body>
    <p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;">
      This paragraph is 400pixels wide and 100 pixels high
    </p>
  </body>
</html>
```



CSS – Dimension
(CO 3)

The line-height Property

The *line-height* property allows you to increase the space between lines of text. The value of the line-height property can be a number, a length, or a percentage.



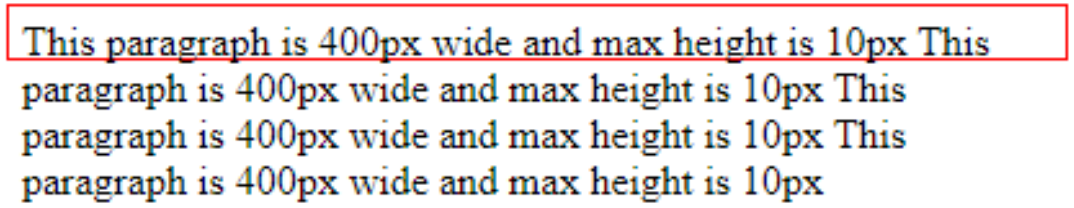

```
<html>
  <head>
  </head>

  <body>
    <p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;
    line-height:30px;">
      This paragraph is 400pixels wide and 100 pixels high and here line height is 30pixels.
      This paragraph is 400 pixels wide and 100 pixels high and here line height is 30pixels.
    </p>
  </body>
</html>
```

CSS – Dimension
(CO 3)

The Height and Width Properties

- The *height* and *width* properties allow you to set the height and width for boxes. They can take values of a length, a percentage, or the keyword auto.

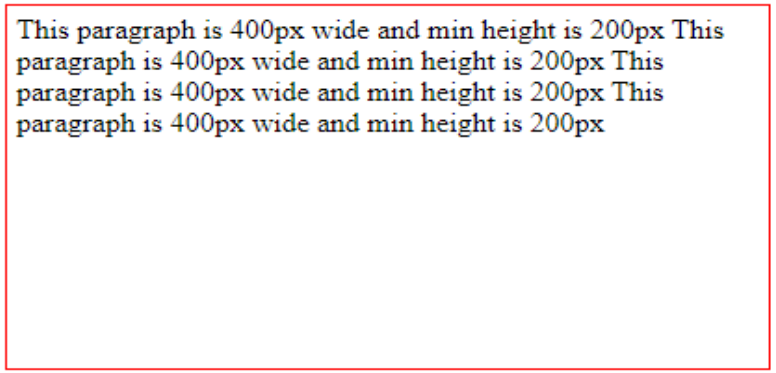
```
<html>
  <head>
  </head>
  <body>
    <p style = "width:400px; max-height:10px; border:1px solid red; padding:5px;
     margin:10px;">
      This paragraph is 400px wide and max height is 10px
      This paragraph is 400px wide and max height is 10px
      This paragraph is 400px wide and max height is 10px
      This paragraph is 400px wide and max height is 10px
    </p>
    </body>
</html>
```



CSS – Dimension
(CO 3)

The min-height Property

The *min-height* property allows you to specify minimum height of a box. The value of the min-height property can be a number, a length, or a percentage.



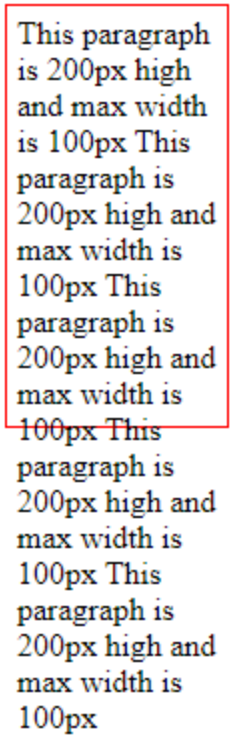

```
<html>
  <head>
  </head>

  <body>
    <p style = "width:400px; min-height:200px; border:1px solid red; padding:5px; margin:10px;">
      This paragraph is 400px wide and min height is 200px
      This paragraph is 400px wide and min height is 200px
      This paragraph is 400px wide and min height is 200px
      This paragraph is 400px wide and min height is 200px
    </p>

  </body>
</html>
```

CSS – Dimension
(CO 3)

The max-width Property

The *max-width* property allows you to specify maximum width of a box. The value of the max-width property can be a number, a length, or a percentage.

```
<html>
  <head>
  </head>
  <body>
    <p style = "max-width:100px; height:200px; border:1px solid red; padding:5px;
     margin:10px;">
       This paragraph is 200px high and max width is 100px
       This paragraph is 200px high and max width is 100px
       This paragraph is 200px high and max width is 100px
       This paragraph is 200px high and max width is 100px
       This paragraph is 200px high and max width is 100px
    </p>
  </body>
```



CSS – Dimension
(CO 3)

The min-width Property

The *min-width* property allows you to specify minimum width of a box. The value of the min-width property can be a number, a length, or a percentage.

```
<html>
  <head>
  </head>

  <body>
    <p style = "min-width:400px; height:100px; border:1px solid red; padding:5px;
     margin:10px;">
       This paragraph is 100px high and min width is 400px
       This paragraph is 100px high and min width is 400px
    </p>
   </body>
</html>
```

CSS – Display
(CO 3)

- The display property specifies if/how an element is displayed.
- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

Examples of block-level elements:

-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>

CSS – Display
(CO 3)

Inline Elements

- An inline element does not start on a new line and only takes up as much width as necessary.
- This is an inline element inside a paragraph.

Examples of inline elements:

-
- <a>
- <img>

CSS – Display
(CO 3)

Override The Default Display Value

- As mentioned, every element has a default display value. However, you can override this.
- Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

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

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

</body>
</html>
```

CSS - Positioning
(CO 3)

CSS helps you to position your HTML element. You can put any HTML element at whatever location you like. You can specify whether you want the element positioned relative to its natural position in the page or absolute.

Relative Positioning

- Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position.

```
<html>
  <head>
  </head>

  <body>
    <div style = "position:relative; left:80px; top:2px; background-color:yellow;">
      This div has relative positioning.
    </div>
</body>
```

This div has relative positioning.

CSS - Positioning
(CO 3)

Absolute Positioning

- An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

```
<html>
  <head>
  </head>

  <body>
    <div style = "position:absolute; left:80px; top:20px; background-color:yellow;">
      This div has absolute positioning.
    </div>
  </body>
</html>
```

CSS - Positioning
(CO 3)

Fixed Positioning

- Fixed positioning allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.

```
<html>
  <head>
  </head>

  <body>
    <div style = "position:fixed; left:80px; top:20px; background-color:yellow;">
      This div has fixed positioning.
    </div>
  </body>
</html>
```

CSS - Pseudo Classes
(CO 3)

- CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows −

  selector:pseudo-class {property: value}

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

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

CSS - Pseudo Classes
(CO 3)

The most commonly used pseudo-classes are as follows –

Sr.No.	Value & Description
1	**:link** Use this class to add special style to an unvisited link.
2	**:visited** Use this class to add special style to a visited link.
3	**:hover** Use this class to add special style to an element when you mouse over it.
4	**:active** Use this class to add special style to an active element.
5	**:focus** Use this class to add special style to an element while the element has focus.
6	**:first-child** Use this class to add special style to an element that is the first child of some other element.
7	**:lang** Use this class to specify a language to use in a specified element.

CSS - Pseudo Classes
(CO 3)

While defining pseudo-classes in a <style>...</style> block, following points should be noted −

- a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
- a:active MUST come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.

CSS - Pseudo Classes
(CO 3)

The :link pseudo-class

- how to use the ***:link*** class **to set the link color**. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:link {color:#000000}
    </style>
  </head>
  <body>
    <a href = "">Black Link</a>
  </body>
</html>
```

CSS - Pseudo Classes
(CO 3)

The :visited pseudo-class

The following is the example which demonstrates how to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:visited {color:blue}
    </style>
  </head>
  <body>
    <a href = "">Click this link</a>
  </body>
</html>
```

Click this link

CSS - Pseudo Classes
(CO 3)

The :hover pseudo-class

The *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:hover {color: #FFCC00}
    </style>
  </head>
  <body>
    <a href = "">Bring Mouse Here</a>
  </body>
</html>
```

Bring Mouse Here

CSS - Pseudo Classes
(CO 3)

The :visited pseudo-class

How to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:visited {color: blue}
    </style>
  </head>
  <body>
    <a href = "">Click this link</a>
  </body>
</html>
```

CSS - Pseudo Classes
(CO 3)

The :hover pseudo-class

- How to use the *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:hover {color: Red}
    </style>
  </head>
  <body>
    <a href = "">Bring Mouse Here</a>
  </body>
</html>
```

Bring Mouse Here

CSS - Pseudo Classes
(CO 3)

The :active pseudo-class

How to use the *:active* class to change the color of active links.
Possible values could be any color name in any valid format.

```
<html>
  <head>
    <style type = "text/css">
      a:active {color: pink}
    </style>
  </head>
  <body>
    <a href = "">Click This Link</a>
  </body>
</html>
```

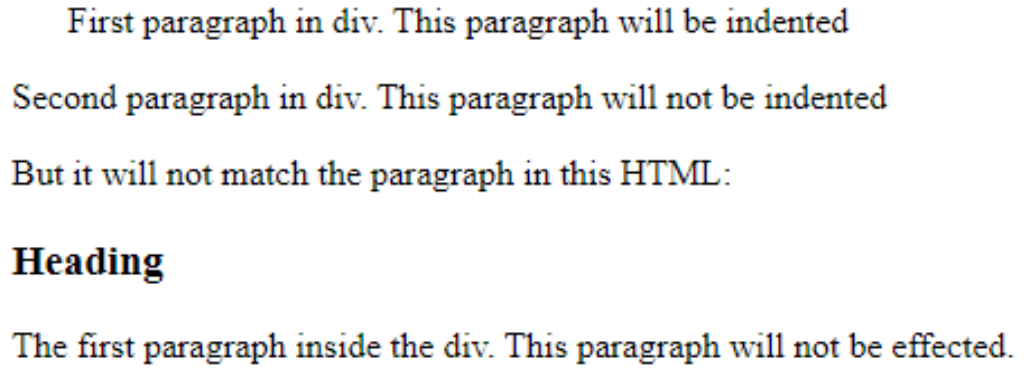
It will produce the link. When a user clicks it, the color changes to pink.

CSS - Pseudo Classes
(CO 3)

The :first-child pseudo-class

- The *:first-child* pseudo-class matches a specified element that is the first child of another element and adds special style to that element that is the first child of some other element.

```
<html>
  <head>
    <style type = "text/css">
      div > p:first-child {
        text-indent: 25px;
      }
    </style>
  </head>

  <body>

    <div>
      <p>First paragraph in div. This paragraph will be indented</p>
      <p>Second paragraph in div. This paragraph will not be indented</p>
    </div>
```



Pseudo Elements
(CO 3)

Pseudo Elements

CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows:

selector:pseudo-element {property: value}

CSS classes can also be used with pseudo-elements

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

Pseudo Elements
(CO 3)

The most commonly used pseudo-elements are as follows −

Sr.No.	Value & Description
1	**:first-line** Use this element to add special styles to the first line of the text in a selector.
2	**:first-letter** Use this element to add special style to the first letter of the text in a selector.
3	**:before** Use this element to insert some content before an element.
4	**:after** Use this element to insert some content after an element.

Pseudo Elements
(CO 3)

The :first-line pseudo-element

```
<html>
  <head>
    <style type = "text/css">
      p:first-line { text-decoration: underline; }
      p.noline:first-line { text-decoration: none; }
    </style>
  </head>

  <body>
    <p class = "noline">
      This line would not have any underline because this belongs to nline class.
    </p>

    <p>
      The first line of this paragraph will be underlined as defined in the
      CSS rule above. Rest of the lines in this paragraph will remain normal.
      This example shows how to use :first-line pseduo element to give effect
      to the first line of any HTML element.
    </p>
  </body>
</html>
```

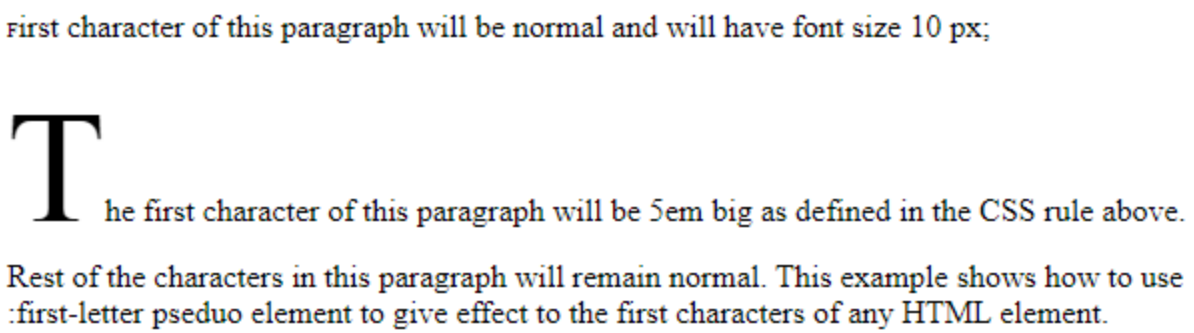
This line would not have any underline because this belongs to nline class.

The first line of this paragraph will be underlined as defined in the CSS rule above. Rest of the lines in this paragraph will remain normal. This example shows how to use :first-line pseduo element to give effect to the first line of any HTML element.

Pseudo Elements
(CO 3)

The :first-letter pseudo-element

```
<html>
  <head>
    <style type = "text/css">
      p:first-letter { font-size: 5em; }
      p.normal:first-letter { font-size: 10px; }
    </style>
  </head>

  <body>
    <p class = "normal">
      First character of this paragraph will be normal and will have font size 10 px;
    </p>

    <p>
      The first character of this paragraph will be 5em big as defined in the
      CSS rule above. Rest of the characters in this paragraph will remain
      normal. This example shows how to use :first-letter pseduo element
      to give effect to the first characters  of any HTML element.
    </p>
  </body>
</html>
```



Pseudo Elements
(CO 3)

The :before pseudo-element

```
<html>
<head>
<style>
h1::before {
  content: url(smiley.gif);
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The ::before pseudo-element inserts content before the content of an element.</p>

<h1>This is a heading</h1>
<p><b>Note:</b> IE8 supports the content property only if a !DOCTYPE is specified.</p>

</body>
</html>
```

This is a heading

The ::before pseudo-element inserts content before the content of an element.

This is a heading

Note: IE8 supports the content property only if a !DOCTYPE is specified.

Pseudo Elements
(CO 3)

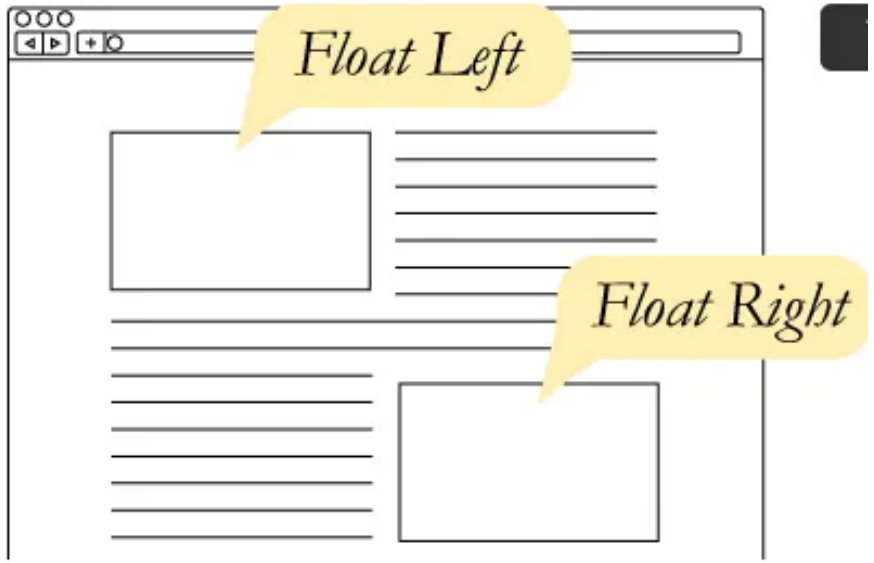
The :after pseudo-element

```
<html>
  <head>
    <style type = "text/css">
      p:after {
        content: url(smiley.gif)
      }
    </style>
  </head>

  <body>
    <p> This line will be succeeded by a bullet.</p>
    <p> This line will be succeeded by a bullet.</p>
    <p> This line will be succeeded by a bullet.</p>
  </body>
</html>
```

CSS float Property
(CO 3)

CSS float Property

The float property specifies how an element should float.



CSS float Property
(CO 3)

```
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>
<h1>The float Property</h1>

<p>In this example, the image will float to the right in the text, and the text in the paragraph will wrap around the image.</p>

<p><img src="pic.jpeg" alt="smiely" style="width:170px;height:170px;margin-left:15px;">
This is an example of CSS float property. This is an example of CSS float property.
This is an example of CSS float property.
This is an example of CSS float property.
This is an example of CSS float property.</p>

</body>
</html>
```

Text Alignment
(CO 3)

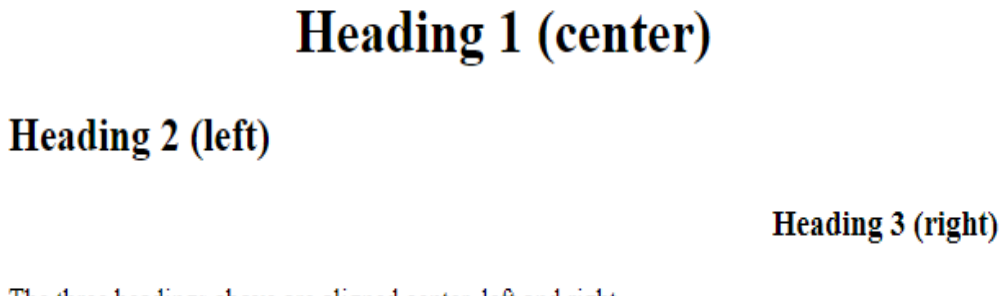
The text-align property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers)

Text Alignment
(CO 3)

```
<html>
<head>
<style>
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
</style>
</head>
<body>

<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>

<p>The three headings above are aligned center, left and right.</p>

</body>
</html>
```



Text Alignment
(CO 3)

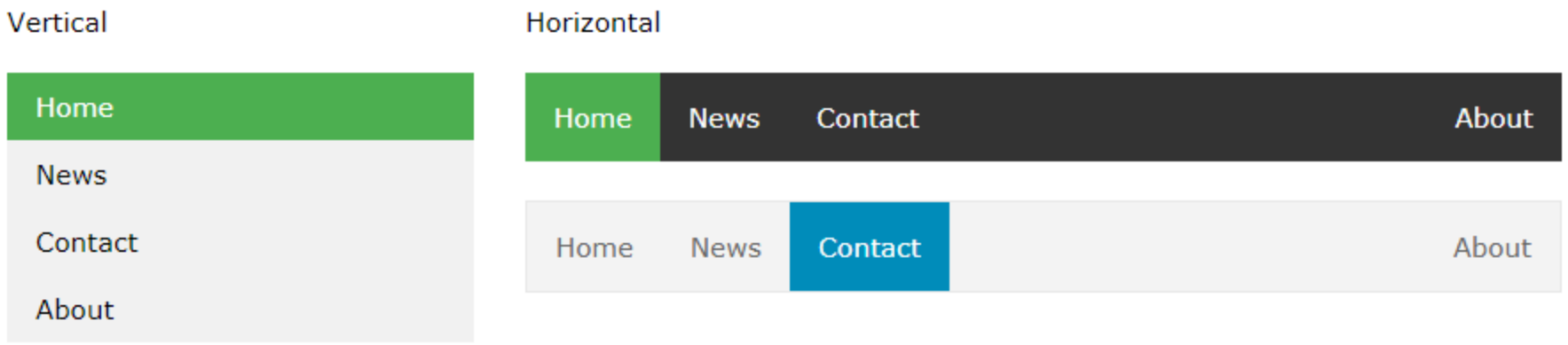
Vertical Alignment

- The vertical-align property sets the vertical alignment of an element.

```
<html>
<head>
<style>
img.top {
  vertical-align: top;
}

img.middle {
  vertical-align: middle;
}

img.bottom {
  vertical-align: bottom;
}
</style>
</head>
<body>

<p>An <img src="smiley.gif" alt="smiley" width="270" height="50"> image with a default alignment.</p><br>

<p>An <img class="top" src="smiley.gif" alt="smiely" width="270" height="50"> image with a top alignment.</p><br>

<p>An <img class="middle" src="smiley.gif" alt="smiely" width="270" height="50"> image with a middle alignment.</p><br>
```

CSS Navigation Bar
(CO 3)

- Having easy-to-use navigation is important for any web site.
- With CSS you can transform boring HTML menus into good-looking navigation bars.



CSS Navigation Bar
(CO 3)

```
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
</style>
</head>
<body>

<p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>
<p>In this example, we remove the bullets from the list, and its default padding and margin.</p>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

Note: We use href="#" for test links. In a real web site this would be URLs.

In this example, we remove the bullets from the list, and its default padding and margin.

Home
News
Contact
About

CSS Website Layout
(CO 3)

Website Layout

A website is often divided into headers, menus, content and a footer:

CSS Website Layout
(CO 3)

Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name.

```
<html >
<head>
<title>CSS Website Layout</title>

<style>
/* Style the header */
.header {
  background-color: yellow;
  padding: 20px;
  text-align: center;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
</div>
```

CSS Website Layout
(CO 3)

Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website.

```
<html>
<head>
<title>CSS Website Layout</title>

<style>

/* Style the header */
.header {
  background-color: pink;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: yellow;
}

/* Style the topnav links */
.topnav a {
  float: left;
```

CSS Website Layout
(CO 3)

Content

The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:

- **1-column** (often used for mobile browsers)
- **2-column** (often used for tablets and laptops)
- **3-column layout** (only used for desktops)

CSS Website Layout
(CO 3)

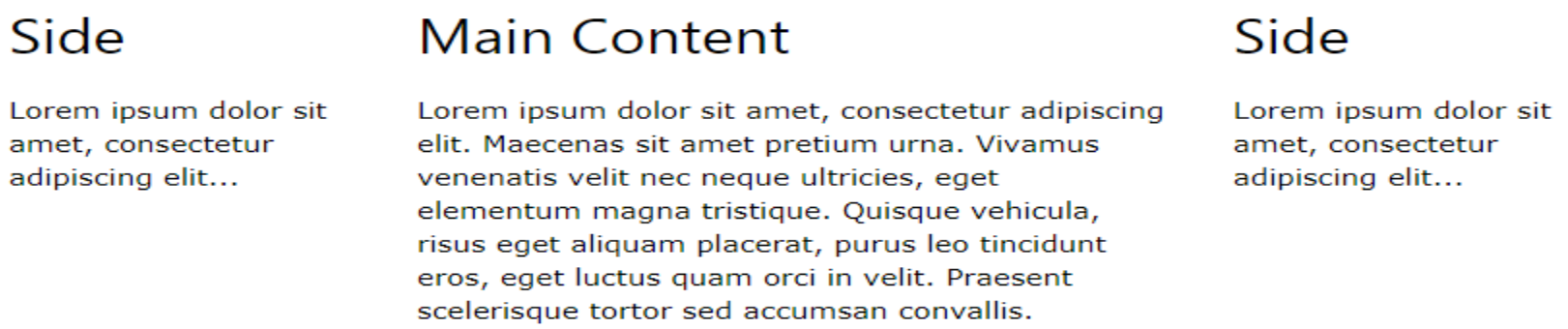
create a 3-column layout, and change it to a 1-column layout on smaller screens

```
<html>
<head>
<title>CSS Website Layout</title>

<style>
 /* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
```

CSS Website Layout
(CO 3)

Unequal Columns

- The main content is the biggest and the most important part of your site.
- It is common with **unequal** column widths, so that most of the space is reserved for the main content. The side content (if any) is often used as an alternative navigation or to specify information relevant to the main content. Change the widths as you like, only remember that it should add up to 100% in total.



CSS Website Layout
(CO 3)

Footer

The footer is placed at the bottom of your page. It often contains information like copyright and contact info.

```
<html>
<head>
<title>CSS Website Layout</title>
<style>

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
```