SRM Institute of Science and Technology
College of Engineering and Technology
School of Computing

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2024-25 (EVEN)

18AIC302J

Web Programming for Artificial Intelligence

UNIT 1

Cascading Style Sheets
(CSS)

CSS

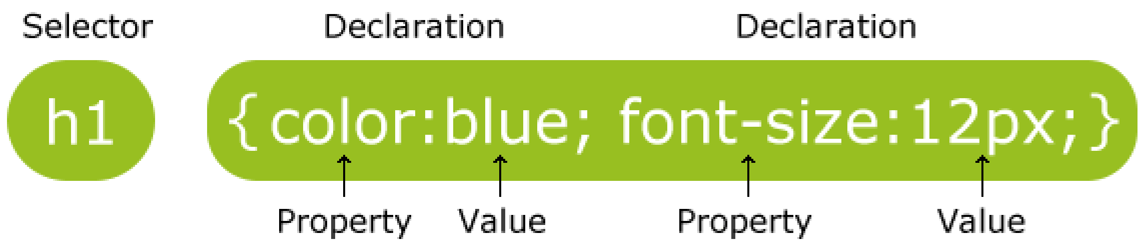
- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files**

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax

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



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.
A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
In the following example all <p> elements will be center-aligned, with a red text color:
Example

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

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector

The element selector selects elements based on the element name.
You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

Example

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

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":
Example
```
#para1 {
    text-align: center;
    color: red;
 }
```
The class Selector

The class selector selects elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the name of the class.
In the example below, all HTML elements with class="center" will be red and center-aligned:
Example
```
.center {
    text-align: center;
    color: red;
 }
```
You can also specify that only specific HTML elements should be affected by a class.
In the example below, only <p> elements with class="center" will be center-aligned:
Example
```
p.center {
    text-align: center;
    color: red;
 }
```
Grouping Selectors

If you have elements with the same style definitions, like this: h1
```
{
    text-align: center;
    color: red;
 }

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

p {
    text-align: center;
    color: red;
 }
```
It will be better to group the selectors, to minimize the code.
To group selectors, separate each selector with a comma.
In the example below we have grouped the selectors from the code above:
Example
```
h1, h2, p {
    text-align: center;
    color: red;
 }
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

Example

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
 }

/* This is
a multi-line
comment */
```

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

With an external style sheet, you can change the look of an entire website by changing just one file! Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

Example

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

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

</body>
</html>
```

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

Here is how the "mystyle.css" looks:

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

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

Note: Do not add a space between the property value and the unit (such as margin-left: 20 px;). The correct way is: margin-left: 20px;

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.
Internal styles are defined within the <style> element, inside the <head> section of an HTML page:
Example

```
<head>
<style>
body {
    background-color: linen;
 }

h1 {
    color: maroon;
    margin-left: 40px;
 }
</style>
</head>
```

Inline Styles

An inline style may be used to apply a unique style for a single element.
To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
The example below shows how to change the color and the left margin of a <h1> element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

Tip: An inline style loses many of the advantages of a style sheet (by mixing content with presentation).

Cascading Order

What style will be used when there is more than one style specified for an HTML element?
Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

Color Names

In HTML, a color can be specified by using a color name:

- Tomato
- Orange
- DodgerBlue
- MediumSeaGreen
- Gray
- SlateBlue

- Violet
- LightGray

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Text Color

Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Border Color

You can set the color of borders:

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

Same as color name "Tomato":

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

RGB Value

In HTML, a color can be specified as an RGB value, using this formula:

rgb(*red, green, blue*)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display the color black, all color parameters must be set to 0, like this: rgb(0, 0, 0).
To display the color white, all color parameters must be set to 255, like this: rgb(255, 255, 255).

HEX Value

In HTML, a color can be specified using a hexadecimal value in the form:

#rrggbb

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

Example

#ff0000
#0000ff
#3cb371
#ee82ee
#ffa500
#6a5acd

HSL Value

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

hsl(*hue, saturation, lightness*)

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage, 0% is black, 50% is neither light or dark, 100% is white

Example

hsl(0, 100%, 50%)

hsl(240, 100%, 50%)

hsl(147, 50%, 47%)

hsl(300, 76%, 72%)

hsl(39, 100%, 50%)

hsl(248, 53%, 58%)

<u>Saturation</u>

Saturation can be describe as the intensity of a color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

0% is completely gray, you can no longer see the color.

Example

hsl(0, 100%, 50%)

hsl(0, 80%, 50%)

hsl(0, 60%, 50%)

hsl(0, 40%, 50%)

hsl(0, 20%, 50%)

hsl(0, 0%, 50%)

RGBA Value

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

rgba(*red, green, blue, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

rgba(255, 99, 71, 0)

rgba(255, 99, 71, 0.2)

rgba(255, 99, 71, 0.4)

rgba(255, 99, 71, 0.6)

rgba(255, 99, 71, 0.8)

rgba(255, 99, 71, 1)

HSLA Value

HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

hsla(*hue, saturation, lightness, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

hsla(9, 100%, 64%, 0)

hsla(9, 100%, 64%, 0.2)

hsla(9, 100%, 64%, 0.4)

hsla(9, 100%, 64%, 0.6)
hsla(9, 100%, 64%, 0.8)
hsla(9, 100%, 64%, 1)

CSS Backgrounds

The CSS background properties are used to define the background effects for elements.
CSS background properties:

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

Background Color

The background-color property specifies the background color of an element.
The background color of a page is set like this:
Example

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

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

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:
Example

```
body {
    background-image: url("gradient_bg.png");
 }
```

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

```
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
 }
```

Tip: To repeat an image vertically, set background-repeat: repeat-y;

Background Image - Set position and no-repeat

Showing the background image only once is also specified by the background-repeat property:
Example

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
 }
```
in the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.
The position of the image is specified by the background-position property:
Example
```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
 }
```
Background Image - Fixed position
To specify that the background image should be fixed (will not scroll with the rest of the page), use the background-attachment property:
Example
```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: fixed;
 }
```
Border Style
The border-style property specifies what kind of border to display.
The following values are allowed:

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).
Example
```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

A dotted border.

A dashed border

Result:

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border. A mixed border.

Border Width

The border-width property specifies the width of the four borders.

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

5px border-width

Example

```
p.one {
    border-style: solid;
    border-width: 5px;
 }

p.two {
    border-style: solid;
    border-width: medium;
 }

p.three {
    border-style: solid;
    border-width: 2px 10px 4px 20px;
 }
```

Border Color

The border-color property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- Hex - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- transparent

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

If border-color is not set, it inherits the color of the element.

Red border

Example

```
p.one {
    border-style: solid;
    border-color: red;
  }

p.two {
    border-style: solid;
    border-color: green;
  }

p.three {
    border-style: solid;
    border-color: red green blue yellow;
  }
```

Border - Individual Sides

From the examples above you have seen that it is possible to specify a different border for each side.

In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):

Different Border Styles

Example

```
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
  }
```

Border - Shorthand Property

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The border property is a shorthand property for the following individual border properties:

- border-width
- border-style (required)
- border-color

Example

```
p {
    border: 5px solid red;
  }
```

Rounded Borders

The border-radius property is used to add rounded borders to an element:

Example

```
p {
    border: 2px solid red;
    border-radius: 5px;
  }
```

CSS Margins

This element has a margin of 70px.
The CSS margin properties are used to create space around elements, outside of any defined borders.
With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- margin-top
- margin-right
- margin-bottom
- margin-left

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- *%* - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.
The following example sets different margins for all four sides of a <p> element:
Example

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
 }
```

Margin - Shorthand Property

- **margin: 25px 50px 75px 100px;**
    - top margin is 25px
    - right margin is 50px
    - bottom margin is 75px
    - left margin is 100px

Example

```
p {
    margin: 25px 50px 75px 100px;
 }
```

The auto Value

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:
Example

```
div {
    width: 300px;
    margin: auto;
    border: 1px solid red;
 }
```

The inherit Value

This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

Example

```
div {
    border: 1px solid red;
    margin-left: 100px;
 }

p.ex1 {
    margin-left: inherit;
 }
```

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- padding-top
- padding-right
- padding-bottom
- padding-left

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

The following example sets different padding for all four sides of a <div> element:

Example

```
div {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
 }
```

Padding - Shorthand Property

- **padding: 25px 50px 75px 100px;**
    - top padding is 25px
    - right padding is 50px
    - bottom padding is 75px
    - left padding is 100px

Example

```
div {
    padding: 25px 50px 75px 100px;
 }
```

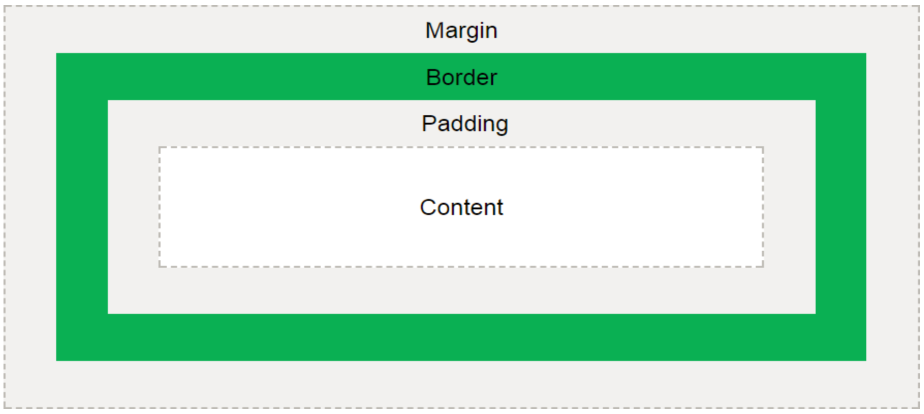
Setting height and width

The height and width properties are used to set the height and width of an element.
The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block.
This element has a height of 200 pixels and a width of 50%
Example

```
div {
     height: 200px;
     width: 50%;
     background-color: powderblue;
 }
```

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.
Example

```
div {
     width: 300px;
     border: 25px solid green;
     padding: 25px;
     margin: 25px;
 }
```

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.
It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:
Example p.uppercase

```
{
     text-transform: uppercase;
 }
```

```
p.lowercase {
    text-transform: lowercase;
 }

p.capitalize {
    text-transform: capitalize;
 }
```

Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text.
The following example demonstrates how to increase or decrease the space between characters:
Example

```
h1 {
    letter-spacing: 3px;
 }
h2 {
    letter-spacing: -3px;
 }
```

Line Height

The line-height property is used to specify the space between lines:
Example

```
p.small {
    line-height: 0.8;
 }

p.big {
    line-height: 1.8;
 }
```

Word Spacing

The word-spacing property is used to specify the space between the words in a text.
The following example demonstrates how to increase or decrease the space between words:
Example

```
h1 {
    word-spacing: 10px;
 }

h2 {
    word-spacing: -5px;
 }
```

Text Shadow

The text-shadow property adds shadow to text.
The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red):
Example

```
h1 {
    text-shadow: 3px 2px red;
 }
```

Font Family

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

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.
Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.
Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".
More than one font family is specified in a comma-separated list:
Example

```
p {
    font-family: "Times New Roman", Times, serif;
 }
```

Font Style

The font-style property is mostly used to specify italic text.
This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Example

```
p.normal {
    font-style: normal;
 }

p.italic {
    font-style: italic;
 }

p.oblique {
    font-style: oblique;
 }
```

Set Font Size With Pixels

Setting the text size with pixels gives you full control over the text size:
Example

```
h1 {
    font-size: 40px;
 }

h2 {
    font-size: 30px;
 }

p {
    font-size: 14px;
 }
```

CSS Icons

How To Add Icons

The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome.

Add the name of the specified icon class to any inline HTML element (like <i> or <span>).
All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

Font Awesome Icons

To use the Font Awesome icons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Note: No downloading or installation is required!

Example

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>

<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>

</body>
</html>
```

Bootstrap Icons

To use the Bootstrap glyphicons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

Note: No downloading or installation is required!

Example

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
 >
</head>
<body>

<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>

</body>
```

```
</html>
```

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links 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

Example

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

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

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

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

Background Color

The background-color property can be used to specify a background color for links:
Example

```
a:link {
    background-color: yellow;
  }

a:visited {
    background-color: cyan;
  }

a:hover {
    background-color: lightgreen;
  }

a:active {
    background-color: hotpink;
  }
```

CSS List Properties

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

Different List Item Markers

The list-style-type property specifies the type of list item marker.

The following example shows some of the available list item markers:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a {
    list-style-type: circle;
  }

ul.b {
    list-style-type: square;
  }

ol.c {
    list-style-type: upper-roman;
  }

ol.d {
    list-style-type: lower-alpha;
  }
</style>
</head>
<body>

<p>Example of unordered lists:</p>
<ul class="a">
   <li>Coffee</li>
   <li>Tea</li>
   <li>Coca Cola</li>
</ul>

<ul class="b">
   <li>Coffee</li>
   <li>Tea</li>
   <li>Coca Cola</li>
</ul>

<p>Example of ordered lists:</p>
<ol class="c">
   <li>Coffee</li>
   <li>Tea</li>
   <li>Coca Cola</li>
```

```
</ol>
<ol class="d">
   <li>Coffee</li>
   <li>Tea</li>
   <li>Coca Cola</li>
</ol>

</body>
</html>
```

An Image as The List Item Marker

The list-style-image property specifies an image as the list item marker:

Example

```
ul {
    list-style-image: url('sqpurple.gif');
  }
```

Remove Default Settings

The list-style-type:none property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add margin:0 and padding:0 to <ul> or <ol>:

Example

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
  }
```

Styling List With Colors

We can also style lists with colors, to make them look a little more interesting.
Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items:

Example

```
ol {
    background: #ff9999;
    padding: 20px;
  }

ul {
    background: #3399ff;
    padding: 20px;
  }

ol li {
    background: #ffe5e5;
    padding: 5px;
    margin-left: 35px;
  }

ul li {
    background: #cce5ff;
```

```
      margin: 5px;
  }
```

Table Borders

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

The example below specifies a black border for <table>, <th>, and <td> elements:

Example table, th, td {

```
     border: 1px solid black;
  }
```

Collapse Table Borders

The border-collapse property sets whether the table borders should be collapsed into a single border:

Example table {
border-collapse: collapse;
}

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