Cascading Style Sheets (CSS)

Why CSS?

- Prior to CSS all formatting was embedded in the HTML document
  - font colors, background styles, element alignments, borders and sizes
- What was the problem?
  - Web site development became a long process as the formatting was repeated for same tag everywhere in the code
  - Addition of <font> and color attributes in HTML 3.2

What is CSS?

- Standard style sheet language used for describing the presentation (layout and formatting) of the web pages
- CSS was introduced in 1996 by the World Wide Web Consortium (W3C) which maintains its standard
- Designed to separate presentation and content
  - Content – HTML
  - Presentation – CSS
- What is the advantage?
  - Simpler HTML
  - Better maintainability
- Latest version is CSS3 with new styling features

What all can be done with CSS?

- apply same style rules on multiple elements
- control the presentation of multiple pages of a website with a single style sheet
- style dynamic states of elements such as hover, focus, etc. which is not possible otherwise
- change the position of an element on a web page without changing the markup
- create animations and transitions effects without using any JavaScript

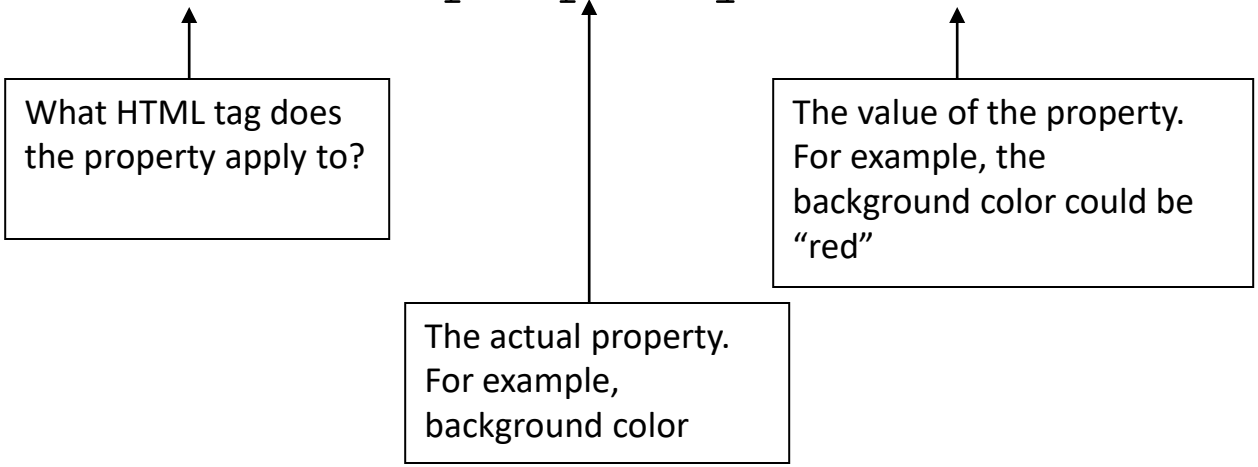
Advantages of CSS

- CSS save lots of time - same code to the groups of HTML elements and on multiple pages too
- Easy maintenance – easy update possible, consistency
- Pages load faster – reduces file transfer size; share formatting information
- Superior styles to html – better layout of web pages
- Multiple device compatibility - same HTML document can be presented in different viewing styles for different devices

CSS Syntax

- The following example shows the fundamental CSS rule syntax:

```
selector { property : value; }
```



- For example,

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

CSS Syntax

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

What does this mean?

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

Whole structure - ruleset

Note:

1. **Curly braces { }**
2. **Colon :**
3. **Semi-colon ;**

Which is

a. Selector
b. Property
c. Value

CSS Syntax

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

```
/* comment in CSS */
h1{
        color:blue;
        text-align: left;
}
```

Multiple selectors

```
p,
li,
h1 {
  color: red;
}
```

```
p{
        font-size: 18px;
        text-transform:uppercase;
}
```

```
<head>
<style>
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
```

```
<body>
<h1>CSS Example</h1>
<p>This is a paragraph.</p>
</body>
```

What about case sensitivity?

- property names and many values are not case-sensitive
- CSS selectors are usually case-sensitive

CSS Selectors

- selects the HTML element(s) that is to be styled
- a pattern to match the elements on a web page

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

Common types of Selector

- Element Type Selectors
- Id Selector
- Universal Selector
- Class Selector
- Grouping Selectors

Element Type Selectors

- Selects the HTML element by name

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

```
h3{
        color: cyan;
}
```

Id Selector

- used to define style rules for a single or unique element
- uses the id attribute of an HTML element to select a specific element.

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

```
<p id="para1">This is first paragraph!</p>

<p>No style applied here </p>
```

Universal Selector

- Applies to every element on the page

```
*{
        margin: 0; padding: 0;
}
```

```
<h1>first Heading!!</h1>
<p id="para1">This is first paragraph!</p>

<p>No style applied here </p>
```

Class Selector

- used to select any HTML element that has a class attribute
- defined with a period sign (.) immediately followed by the class value.

```
<html lang="en">
<head>
<title>Example of CSS class selector</title>
  <style>
    .blue {
        color: #0000ff; font-size: 14px;
    }
  </style>
</head>
<body>
  <h1 class="blue">This is a heading</h1>
  <p class="blue">This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Example of CSS class selector</title>
 <style>
   p.Color1 {
      color: blue;
   }
 </style>
</head>
<body>
 <h1 class=“Color1">This is a heading</h1>
 <p class=“Color1">This is a paragraph.</p>
 <p>This is another paragraph.</p>
</body>
</html>
```

The style rule inside the selector p.Color1 renders the text in blue of only those <p> elements that has class attribute set to Color1, and has no effect on other paragraphs.

```
.highlight {
    background-color: yellow;
}
```

```
<p class = “highlight”>
This is  a paragraph.
</p>
```

```
<h2 class = “highlight”>
Types of Books
</p>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Example of CSS Selectors without Grouping</title>
<style>
        h1 {
                font-size: 36px;
                font-weight: normal;
        }
        h2 {
                font-size: 28px;
                font-weight: normal;
        }
        h3 {
                font-size: 22px;
                font-weight: normal;
        }
</style>
</head>
<body>
        <h1>This is a heading of level 1</h1>
        <h2>This is a heading of level 2</h2>
        <h3>This is a heading of level 3</h3>
</body>
</html>
```

Without grouping

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of CSS Grouping Selectors</title>
<style>
        h1, h2, h3 {
                font-weight: normal;
        }
        h1 {
                font-size: 36px;
                text-transform: uppercase;
        }
        h2 {
                font-size: 28px;
        }
        h3 {
                font-size: 22px;
        }
</style>
</head>
<body>
        <h1>This is a heading of level 1</h1>
        <h2>This is a heading of level 2</h2>
        <h3>This is a heading of level 3</h3>
</body>
```

Types of Selector: **Grouping Selectors**

Grouping Selectors

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

Who wins?

```
h1 {
    color: red;
}
h1 {
    color: black;
}
```

Cascade

```
.first-heading {
    color: red;
}

h1 {
    color: blue;
}
```

```
<h1 class="first-heading">This is my heading.</h1>
```

Specificity

Inheritance

```
body {
    color: blue;
}

span {
    color: black;
}
```

```
<p>This will have body color</p>
<p>Even this will have body color except <span>span text</span>
</p>
```

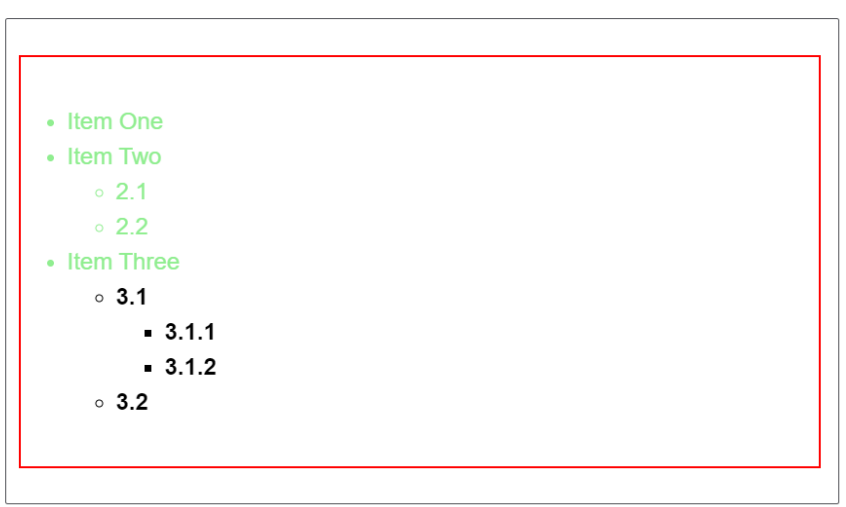
This will have body color

Even this will have body color except span text.

```
.main {
    color: lightgreen;
    border: 2px solid red;
    padding: 2em;
}

.special {
    color: black;
    font-weight: bold;
}
```

```
<ul class="main">
    <li>Item One</li>
    <li>Item Two
        <ul>
            <li>2.1</li>
            <li>2.2</li>
        </ul>
    </li>
    <li>Item Three
        <ul class="special">
            <li>3.1
                <ul>
                    <li>3.1.1</li>
                    <li>3.1.2</li>
                </ul>
            </li>
            <li>3.2</li>
        </ul>
    </li>
</ul>
```



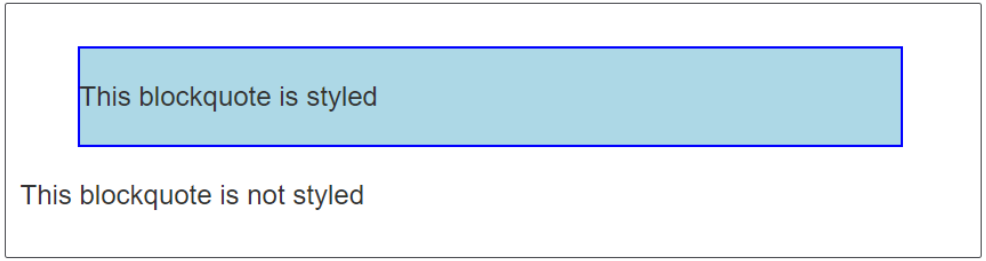
Inheritance

Note

- Properties like width, margin, padding, and border are not inherited

```
blockquote {
    background-color: lightblue;
    border: 2px solid blue;
}

.fix-this {
    all: unset;
}
```



```
<blockquote>
        <p>This blockquote is styled</p>
    </blockquote>

    <blockquote class="fix-this">
        <p>This blockquote is not styled</p>
    </blockquote>
```

Overriding behaviour

```
h2 {
    font-size: 2em;
    color: #000;
    font-family: Georgia, 'Times New Roman',
Times, serif;
}

.small {
    font-size: 1em;
}

.bright {
    color: purple;
}
```

```
<h2>Heading with no class</h2>
<h2 class="small">Heading with class of
small</h2>
<h2 class="bright">Heading with class of
bright</h2>
```

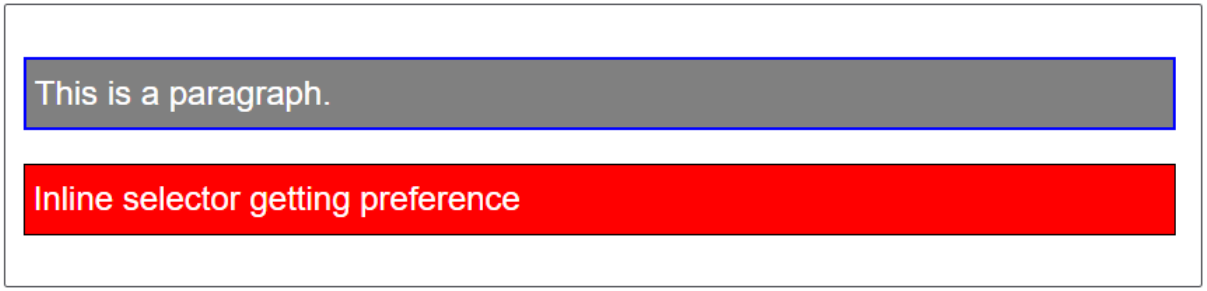
Inline styles

```
#p1{
    background-color: red;
    border: 1px solid black;
}

.old {
    background-color: gray;
    border: 2px solid blue ;
}

p {
    background-color: blue;
    color: white;
    padding: 5px;
}
```

```
<p class="old">This is a paragraph.</p>
<p class="old" id="p1">Inline selector
getting preference</p>
```



How to attach styles?

Style Attachment Types

- There are three ways to attach Styles to a tag
  1. In-Line Styles – using style attribute in the tag

  2. Internal Styles or Embedded styles - <style> element in the head section of a document

  3. External Styles Sheets - <link> element, pointing to an external CSS file

In-Line Style Sheet

- In-line styles are the simplest way to attach a style to the tag
- Includes the style attribute of the relevant tag along with a list of properties and their values
- For example, to change the color and the left margin of a paragraph

```
<p style="color: maroon; margin-left: 10px"> Inline style has been used for this para</p>
```

```
<h1 style="color:b; font-size:25px;">This is a heading</h1>
```

In-Line Style Sheet

- Style rules are embedded directly inside the HTML tag, it causes the presentation to become mixed with the content of the document

Internal Style Sheet

- Internal style sheets are used when a single document has a unique style
- It is defined in **the head section** with the `<style>` tag
- For example,

```
<head>
  <style type="text/css">
    hr {color: sienna}
    p {margin-left: 20px}
    body {background-image: url("images/heart1.gif")}
  </style>
</head>
```

Internal Style Sheet

- The type attribute of the <style> and <link> tag (i.e. type="text/css") defines the language of the style sheet.
- This attribute is purely informative. You can omit this since CSS is the standard and default style sheet language in HTML5.

External Style Sheets

- External style sheet is a file separate from the HTML document
- External style sheet can be reused
- Ideal when style to be applied to different pages in the same web site
- most flexible - change the look of an entire website by changing just one file.

External Style Sheets

- Can be attached in 2 ways

Linked External Style Sheets

- The `<link>` tag is used to create linked external style sheet as follows –

```
body {
   background: lightyellow;
   font: 18px Arial, sans-serif;
}
h1 {
   color: orange;
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>My HTML Document</title>
   <link rel="stylesheet" href=
   “d:/CSS_ExternalLinkStyle.css">
</head>
<body>
   <h1>This is a heading</h1>
   <p>This is a paragraph of text.</p>
</body>
</html>
```

Imported External Style Sheets

- The `<style>` tag is used to create imported external style sheet with the keyword ‘import’

Linked and Imported Styles

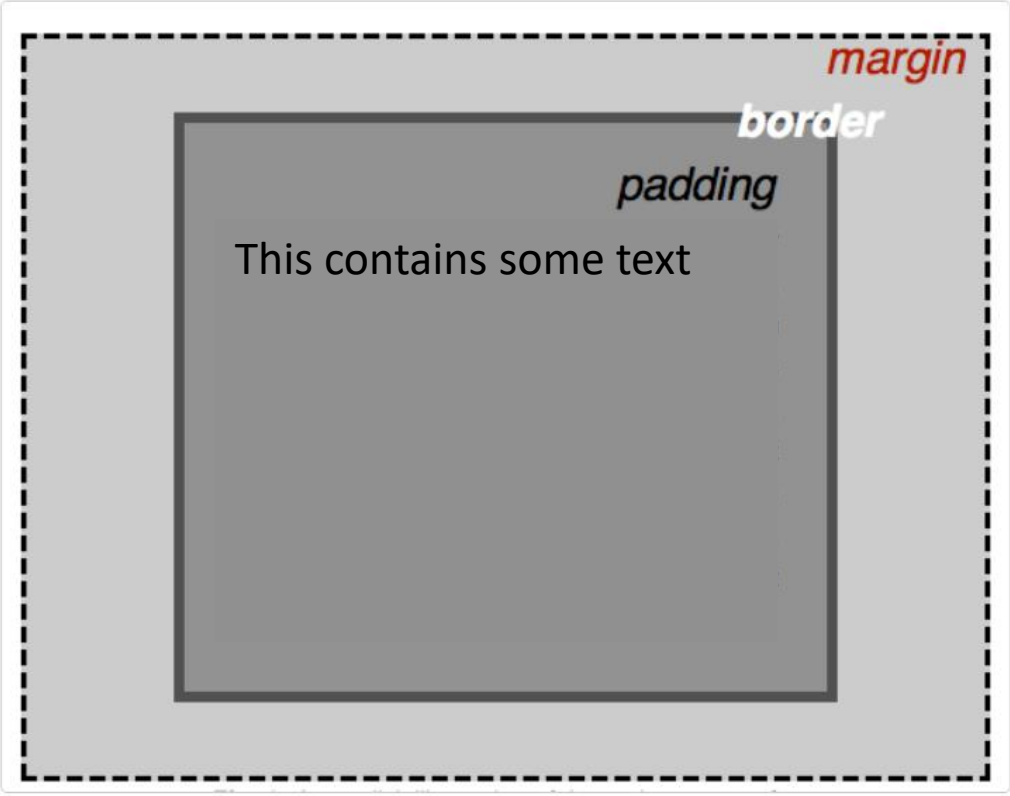
- Imported styles override linked external styles

```
<head>
<link rel=stylesheet href=style1.css
  type=text/css>
<style>
  @import
  url=(http://www.mywebsite.com/style1.css
  )
</style>
</head>
```

Understanding CSS boxes

CSS

- Styling with CSS is with boxes
  - setting size, color, and position
  - Most HTML elements on the web page can be thought of as boxes sitting on top of other boxes.



padding - the space around the content

border - the solid line that is just outside the padding

margin - the space around the outside of the border

```
body {
  width: 900px;
  margin: 0 auto;
  background-color: #ff9500;
  padding: 0 20px 20px 20px;
  border: 2px solid black;
}
```

```
h1 {
  margin: 0;
  padding: 20px 0;
  color: #00539f;
  text-shadow: 3px 3px 1px black;
}
```

CSS Color

CSS Color

- defines the text color (foreground color in general) of an element

Color	Color Name	Hex Value	RGB Value	Shades
	aqua	#00FFFF	rgb(0, 255, 255)	Shades
	black	#000000	rgb(0, 0, 0)	Shades
	blue	#0000FF	rgb(0, 0, 255)	Shades
	fuchsia	#FF00FF	rgb(255, 0, 255)	Shades
	gray	#808080	rgb(128, 128, 128)	Shades
	green	#008000	rgb(0, 128, 0)	Shades
	lime	#00FF00	rgb(0, 255, 0)	Shades
	maroon	#800000	rgb(128, 0, 0)	Shades
	navy	#000080	rgb(0, 0, 128)	Shades
	olive	#808000	rgb(128, 128, 0)	Shades
	purple	#800080	rgb(128, 0, 128)	Shades
	red	#FF0000	rgb(255, 0, 0)	Shades

Extended color keywords

The following table lists the extended color keywords defined in CSS3 specification.

Color	Color Name	Hex Value	RGB Value	Shades
	aliceblue	#F0F8FF	rgb(240, 248, 255)	Shades
	antiquewhite	#FAEBD7	rgb(250, 235, 215)	Shades
	aqua	#00FFFF	rgb(0, 255, 255)	Shades
	aquamarine	#7FFFD4	rgb(127, 255, 212)	Shades
	azure	#F0FFFF	rgb(1240, 255, 255)	Shades
	beige	#F5F5DC	rgb(245, 245, 220)	Shades
	bisque	#FFE4C4	rgb(255, 228, 196)	Shades
	black	#000000	rgb(0, 0, 0)	Shades

Pre-defined


Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
LightGray

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Learning CSS</p>
```

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Learning CSSp>
<p style="color:MediumSeaGreen;">Learning JS</p>
```

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Color Keywords</title>
<style>
   h1 {
      color: red;
   }
   #para1 {
      background-color: yellow;
   }
</style>
</head>
<body>
   <h1>This is a heading</h1>
   <p id =“para1”>This is a paragraph.</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>CSS transparent Color Keywords</title>
<style>
   h1 {
      color: transparent;
   }
   p {
      background-color: transparent;
   }
</style>
</head>
<body>
   <h1>This is a heading</h1>
   <p>This is a paragraph.</p>
</body>
</html>
```

RGB Color values

```
<style>
   h1 {
      color: #f80;
   }
   p {
      background-color: #ff8800;
   }
</style>
```

```
<style>
   h1 {
      color: rgb(0,255,255);
   }
   p {
      background-color: rgb(0%,100%,100%);
   }
</style>
```

CSS Background

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

Properties

value of opacity from 0.0 - 1.0

```
body {
  background-color: lightblue;
}
h1 {
  background-color: green;
}
```

```
div {
  background-color: green;
  opacity: 0.3;
}
```

```
div {
  background: rgba(0, 128, 128, 0.3)
}
```

Properties

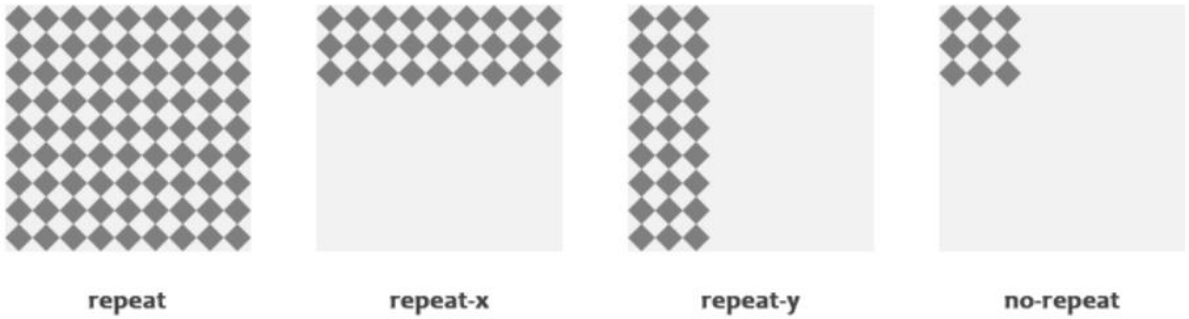
```
body {
  background-image: url(“sea.png");
}
```

Can be applied also for elements

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

Values:

- repeat-x
- repeat-y
- norepeat



```
background-position: right top;
```

```
background-attachment: fixed;
```

Values:

- fixed
- scroll

Font selection

- Most important
  - Decides how user feel
  - What is more important
  - Make something distinct
  - Choosing right color for text

CSS Fonts

- font-family
- font-style
- font-weight
- font-size and
- font-variant

5 Generic Font Families

1. Serif fonts
2. Sans-serif fonts
3. Monospace fonts
4. Cursive fonts
5. Fantasy fonts

Courier New

Lucida Console

Monaco

Brush Script MT

Lucida Handwriting

Copperplate

Papyrus

```
body {
font-family: Arial, Helvetica, sans-serif;
}
```

Last should be generic family

Serif, sans-serif, monospace, cursive and fantasy

“Times New Roman”

Fallback system to ensure maximum compatibility between browsers/operating systems

```
h1 {
 font-size: 24px;
}
p {
font-size: 14px;
}
```

```
body {
font-size: large;
}
h1
{
font-size: larger;
}
p {
font-size: smaller;
}
```

Default size is 16

```
body {
  font-size: 100%;
}

h1 {
  font-size: 2.5em;
}

h2 {
  font-size: 1.875em;
}

p {
  font-size: 0.875em;
}
```

Responsive Font Size viewport width

```
<h1 style="font-size:10vw;">Responsive Text</h1>

<p style="font-size:5vw;">Resize the browser window to see how the text size scales.</p>
```

text size will follow the size of the browser window

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

```
h1 {
 font-size: 24px;
}
p {
font-size: 14px;
}
```

```
body {
font-size: large;
}
h1
{
font-size: larger;
}
p {
font-size: smaller;
}
```

Default size is 16

```
p {
   font-weight: bold;
}
```

Other Values:

normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

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

Other Values normal

CSS Text

- text-align
  - left, right, centre or justify
- text-decoration
  - overline, line-through, underline, none
- text-decoration-style
  - Solid, double, dotted, dashed
- text-transform
  - Uppercase, capitalize, lowercase
- text-indent
  - text-indent: 100px;

CSS Text

- line-height
  - line-height: 1.2;
- letter-spacing
  - letter-spacing: 10px;
- word-spacing
  - word-spacing: 20px;

CSS links

- What are default styles for links?
  - Underlined
  - Unvisited links – blue
  - Visited links – purple
  - Hovering – hand icon
  - Active links - red

CSS Links

```
<head>
<style>
a:link {
  color: red;
}

a:visited {
  color: green;
}
a:hover {
  color: hotpink;
}

a:active {
  color: blue;
}
    }
</style>
</head>
<body>
<p><a href="https://www.google.com/"
target="_top">Google...</a></p>
</body>
```

Unvisited link - underlined blue
Visited link – underlined purple
Active link – underlined red

Hover state - remains blue, purple or red depending on which state it is in

CSS Links

```
a:link, a:visited {
text-decoration: none;
}

a:hover, a:active {
text-decoration: underline;
}
```

CSS Lists

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

CSS Lists

```
ul {
list-style-type: square;
}

ol {
list-style-type: upper-roman;
}
```

CSS Lists

```
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;
}
```

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

```
ul.a {
  list-style-position: outside;
}

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

All in one…

Maintain the order

```
ul {
  list-style: square inside url(“myimage.gif");
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Setting the Position of List
Marker</title>
<style>
```

```
</style>
<body>
   <h2>List Marker Position - Inside</h2>
   <ol class="mystle1 ">
      <li>CSS Tables</li>
      <li>CSS Links</li>
      <li>CSS Color</li>
   </ol>
   <h2>List Marker Position - Outside</h2>
   <ol class="mystle2">
      <li>CSS Tables</li>
      <li>CSS Links</li>
      <li>CSS Color</li>
   </ol>
</body>
</html>
```

```
<style>
   body{
      font-size: 14px;
      font-family: Arial,sans-serif;
   }
   ol li {
      background: #ddd;
      padding: 5px;
      margin: 5px;
   }
   ol.mystle1 li {
      list-style-position: inside;
   }
   ol. mystle2 li {
      list-style-position: outside;
   }
</style>
```

CSS tables

```
table, th, td {
 border: 1px solid black;
width: 100%;
height: 20px;
Border-spacing: 15px;
}
```

```
table {
border-collapse: collapse;
}
th, td {
 border: 1px solid black;
padding : 15px;
text-align: left;
 }
```

CSS Tables

```
table {
width: 300px; table-layout: fixed;
}
caption
{ caption-side: bottom;
}
```

```
table {
border-collapse: separate;
empty-cells: hide;
}
```

Properties

- text-align– left, center, right
- vertical-align – top, bottom, middle
- border-bottom: 1px solid red;
- background-color: #04AA6D;

```
tr:hover {background-color: coral;}
```

```
color: white;
```

```
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid red;
}
```

CSS shadow effects

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

```
h1 {
  color: white;
  text-shadow: 2px 2px 4px #000000;
}
```

```
h1 {
  color: white;
  text-shadow: 1px 1px 2px black, 0 0 25px
blue, 0 0 5px darkblue;
}
```

white text with black, blue, and darkblue shadow

Box shadow

```
div {
  box-shadow: 10px 10px;
}
```

```
div {
  box-shadow: 10px 10px lightblue;
}
```

```
box-shadow: 10px 10px 5px lightblue inset;
```

```
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: coral;
  box-shadow: 10px 10px 5px lightblue;
}
```

CSS texts

- Text overflow
- Word-wrap

```
p {
  word-wrap: break-word;
}
```

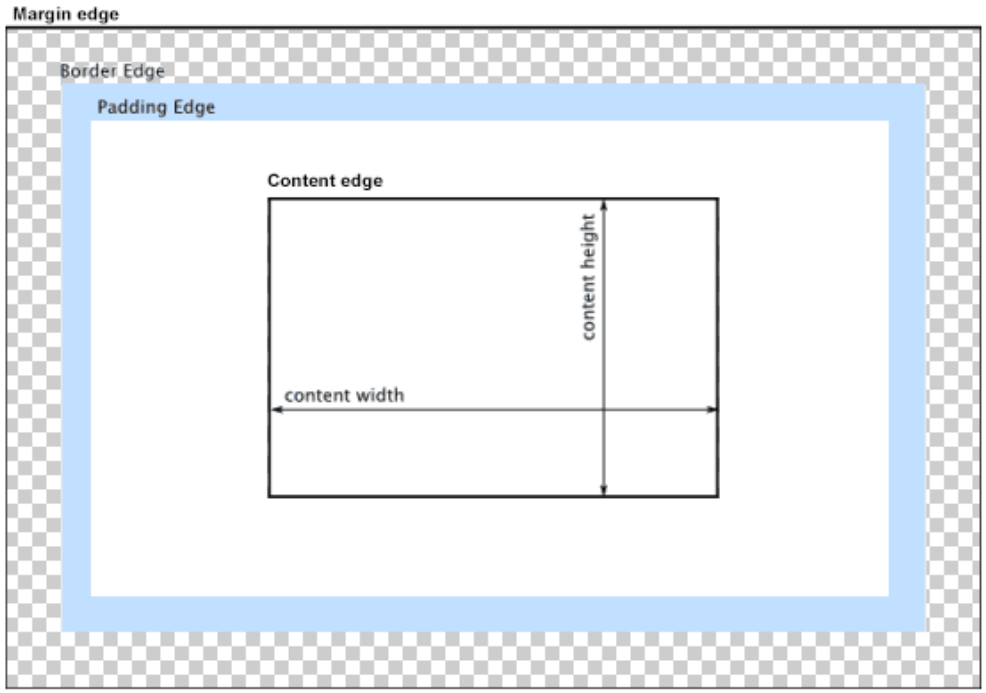
text-underline-position: auto;

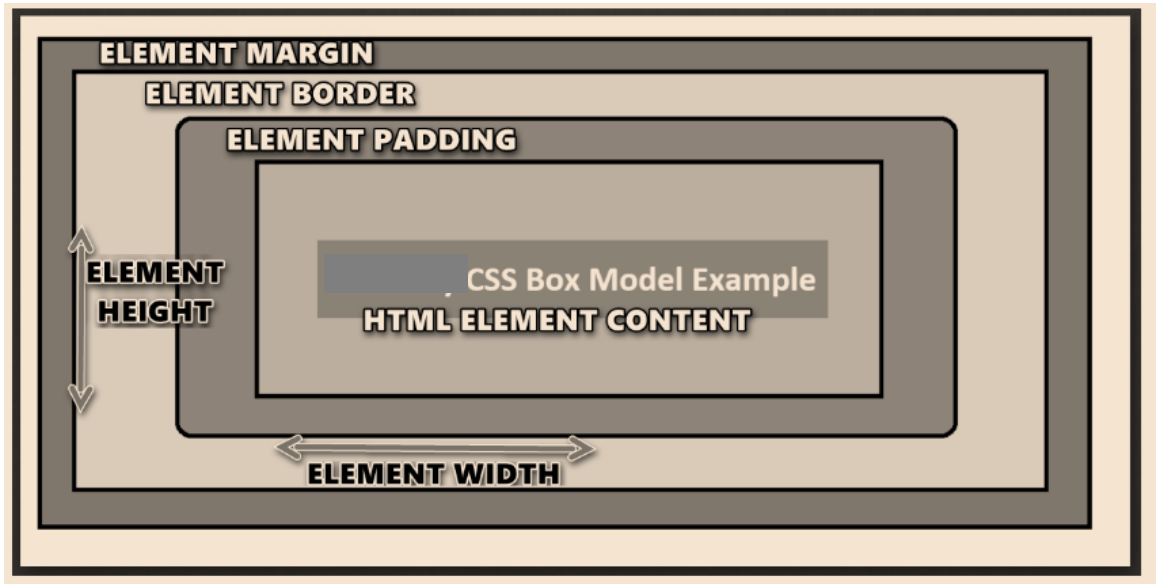
text-underline-position: under;

CSS Box Model

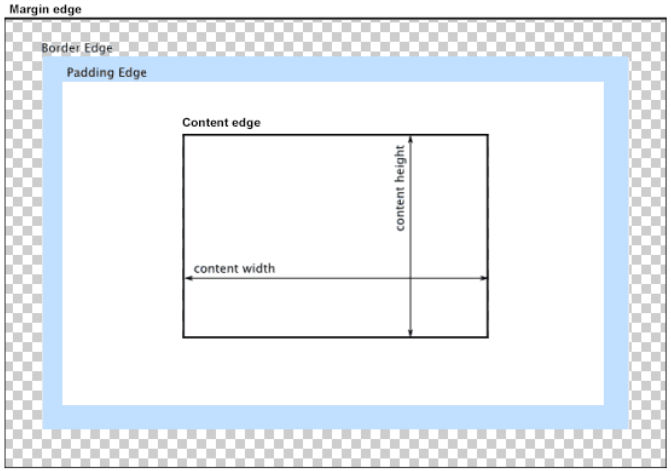
CSS Box Model

- For layout of the document
  - how HTML elements are organized on the screen
- Each element is represented as a Rectangular Box by browser's rendering engine
- Determines the size, position, and properties (color, background, border size, etc.) of these boxes


Margin edge
Border Edge
Padding Edge
Content edge
content height
content width

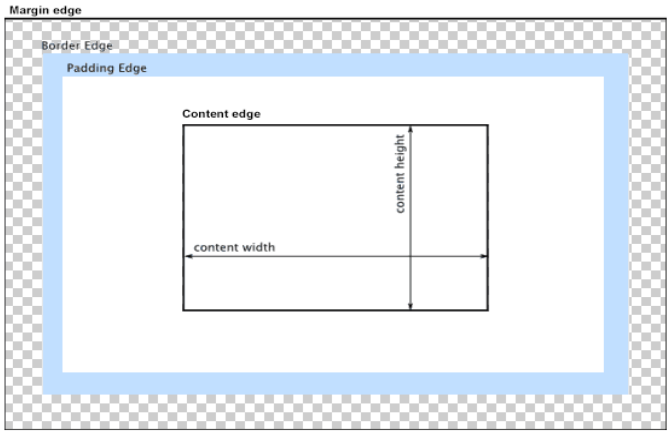

ELEMENT MARGIN
ELEMENT BORDER
ELEMENT PADDING
ELEMENT HEIGHT
CSS Box Model Example
HTML ELEMENT CONTENT
ELEMENT WIDTH

Content area



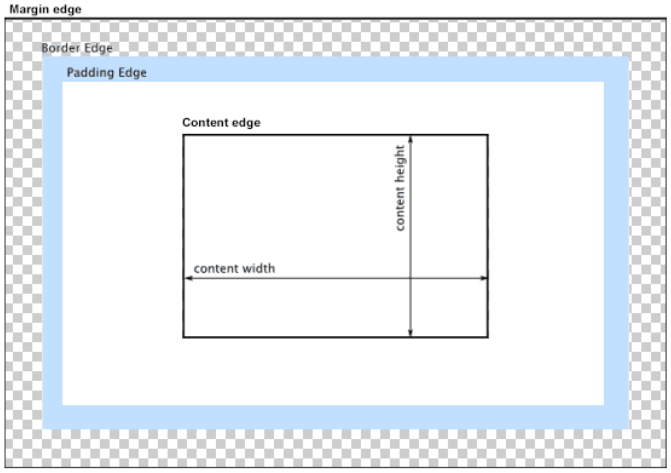
- bounded by the content edge
- contains the "real" content of the element, such as text, an image, or a video player
- often has a background color or background image
- box-sizing property is set to content-box (default) and if the element is a block element
  - the content area's size can be explicitly defined with the width, min-width, max-width, height, min-height, and max-height properties

Padding area



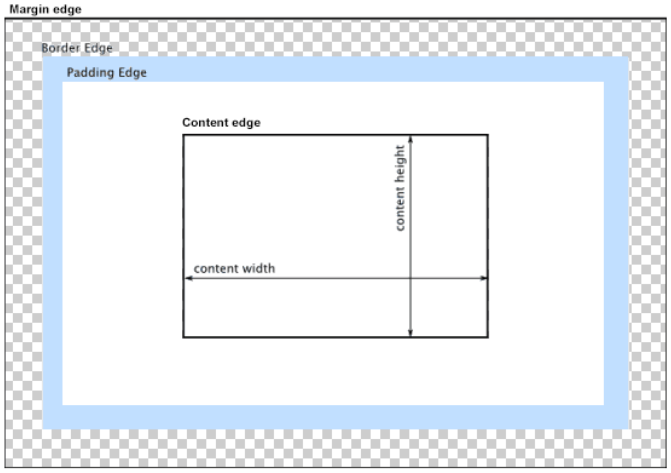
- bounded by the padding edge
- extends the content area to include the element's padding
  - dimensions are the padding-box width and the padding-box height.
- Thickness of padding is determined by
  - padding-top, padding-right, padding-bottom, padding-left, and shorthand padding properties

Border area



- bounded by the border edge, extends the padding area to include the element's borders
  - dimensions are the border-box width and the border-box height
- The thickness of the borders are determined by the border-width and shorthand border properties

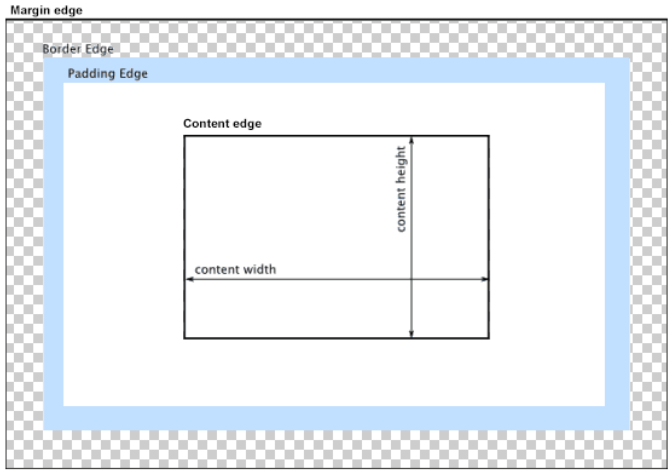
Border area



- If the box-sizing property is set to border-box, the border area's size can be explicitly defined with the width, min-width, max-width, height, min-height, and max-height properties

- When there is a background (background-color or background-image) set on a box, it extends to the outer edge of the border (i.e. extends underneath the border in z-ordering). This default behavior can be altered with the background-clip css property.

Margin area



- bounded by the margin edge, extends the border area to include an empty area used to separate the element from its neighbors
  - dimensions are the margin-box width and the margin-box height
- The size of the margin area is determined by the margin-top, margin-right, margin-bottom, margin-left, and shorthand margin properties.
- When margin collapsing occurs, the margin area is not clearly defined since margins are shared between boxes.

Box Model: examples

- Open file BoxModel1.html
- Open file BoxModel2.html

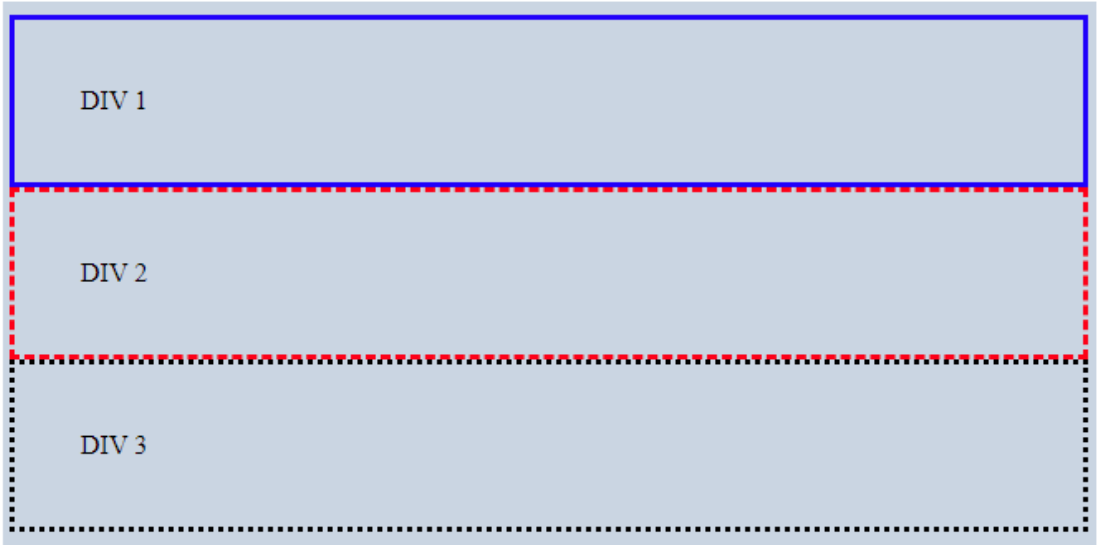
- 320px (width)
  + 20px (left + right padding)
  + 10px (left + right border)
  + 0px (left + right margin)
  = 350px
- Total element width = width + left padding + right padding + left border + right border + left margin + right margin
- Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Box Model

Change padding value and note the difference

```
#div1 {
padding: 40px;
border-style: solid;
border-color: blue;
 }
#div2 {
padding: 40px;
border-style: dashed;
border-color: red;
 }

 #div3 {
padding: 40px;
border-style: dotted;
 border-color: black;
}
```



```
#div1 {
padding: 15px;
border-style: solid;
border-color: blue;
margin: 20px;
width: 100px;
height: 100px;
 box-sizing: content-box;
}
```

```
#div1 {
padding: 15px;
border-style: solid;
border-color: red;
margin: 20px;
width: 100px;
height: 100px;
 box-sizing: border-box;
}
```