

# CSS

Cascading Style Sheets

# Contents

1. Introduction of CSS?
2. Cascading Order
3. Examples of Properties
4. Limitations
5. CSS variations

# CSS

CSS stands for Cascading Style Sheets

CSS is a style sheet language used to determine the formatting of an HTML document.

Styles define how to display (X)HTML elements

Styles are normally stored in Style Sheets

Multiple style definitions will **cascade** into one

# Why to use Styles?

- Documents written with CSS are
  - more flexible
  - short
  - clear
- CSS can control the fonts, text, colors, backgrounds, margins, and layout
- Redesign your site more quickly
- Build more diverse websites

# Style Sheet Languages

Style sheet is a document that contains style information written in a specified format about one or more documents written in markup language.



# CSS Introduction

- A style sheet language that specifies how to incorporate style information in a style sheet.
- 
- Separates the presentation from the content.
- 
- Reduces the development time.
- 
- Speeds up overall response time.

# How to Add CSS

- There are 3 ways to specify the style information in a document.
  - External
    - Written in a separate file and referenced from an HTML document.
    - Useful when same style is applied on different documents.
  - Internal
    - Style information is placed under one style tag in the head section of an HTML page.  
`<style>Write style rules here</style>`
  - Inline
    - Style information is incorporated directly into HTML tags using style attribute.  
`<p style="color: red;">Hello World</p>`

# Cascading Rule

- multiple external style sheets can be referenced inside a single HTML document.

## Cascading Order:

- Browser default (**Lowest Priority**)
- External style sheet
- Internal style sheet (in the head section)
- Inline style (inside an HTML element) (**highest Priority**)



# CSS SYNTAX

# Basic Syntax

- Made up of three parts:



**selector { property: value }**

- The selector is normally the HTML element/tag we want to define
- The property is the attribute we wish to change
- Every property has the value

# Syntax

- If the value is multiple words, put quotes around the value

```
p {font-family: "Times new roman"}
```

# Example

```
p
{
  text-align: center;
  color: black;
  font-family: arial
}
```

# Grouping multiple styles

```
h1,h2,h3,h4,h5,h6  
{  
  color: green  
}
```

- By above command all header elements will be displayed in green text color

This is header h1

This is header h2

This is header h3

This is header h4

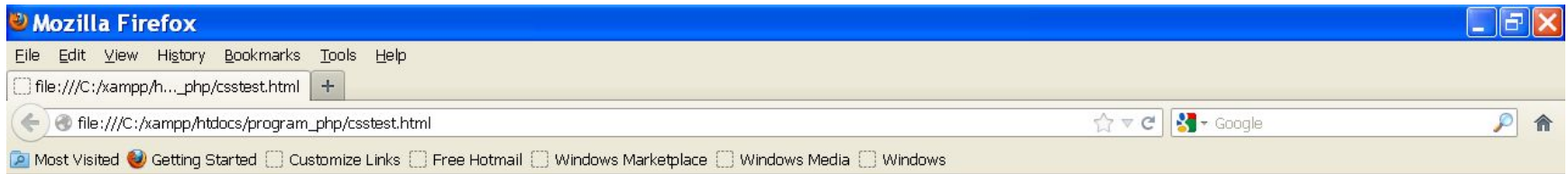
# The class Selector

- The class Selector
- The class selector is used to specify a style for a group of elements.
- It is used to set a particular style for many HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."

# Example

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p.center`
- `{`
- `text-align:center;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<h1 class="center">This heading will not be affected</h1>`
- `<p class="center">This paragraph will be center-aligned.</p>`
- `</body>`
- `</html>`

# Output



**This heading will not be affected**

This paragraph will be center-aligned.



{ Text color

```
<html><head>
<style type="text/css">
  h1 {color: green}
  h2 {color: #dda0dd}
  p {color: rgb(0,0,255)}
</style>
</head>
<body>
  <h1>This is header 1</h1>
  <h2>This is header 2</h2>
  <p>This is a
  paragraph</p>
</body>
</html>
```

**This is header 1**

**This is header 2**

This is a paragraph

CSS

# INSERTING OF STYLE SHEET

# Inserting style sheet

## Three Ways to Insert CSS

- External style sheet
- Internal style sheet
- Inline style
-

## External style sheet

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

# Mystyle.css

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

# External Style Sheet

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

# Internal Style Sheet

- Should be used when a single document has a unique style.
- Defined in the head section by using the `<style>` tag

# Example (internal style sheet)

```
<html>
<head>
<style type="text/css">

hr      {  color: white  }

p      {color:#FFFFFF;   margin-left:120px      }

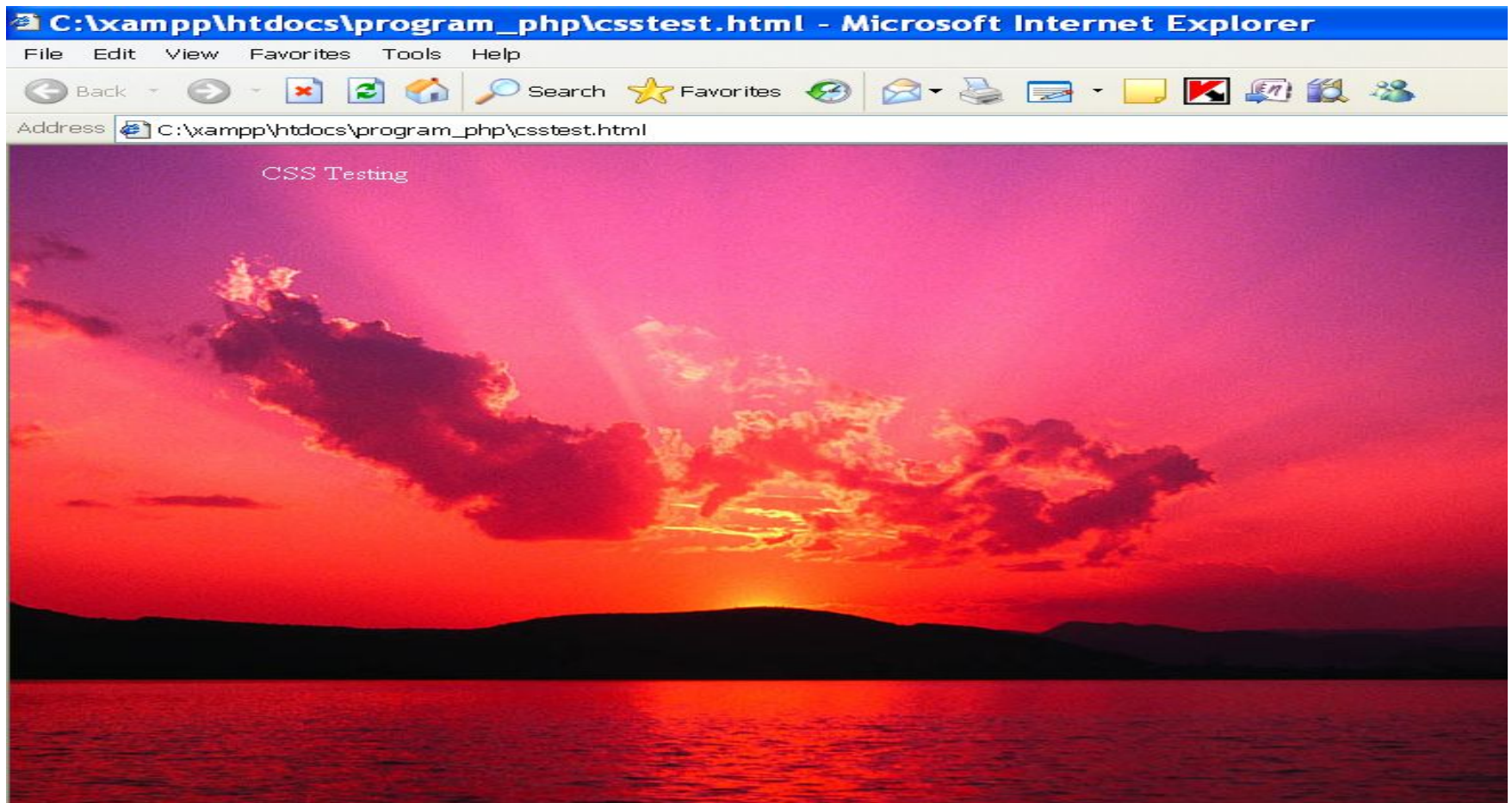
body    { background-image: url("sunset.jpg") }
</style>

</head>

<body>
<p> CSS Testing </p>
</body>
</html>
```



## Output



# Multiple Style Sheets

- Multiple Style Sheets
- If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

# Multiple style sheet

Suppose an external style sheet has these properties for the h3 selector:

```
H3 { color:red; text-align:left; font-size:8pt; }
```

And an internal style sheet has these properties for the h3 selector:

```
H3 { text-align:right; font-size:20pt; }
```

# Multiple style sheet

- So if the page with the internal style sheet also links to the external style sheet then the properties for h3 will be:
  - color:red;  
text-align:right;  
font-size:20pt;

CSS Properties & values

# EXAMPLES

## Background:

- Control over the background color of an element
- set an image as the background,
- Repeat a background image

# CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element

# CSS Background Properties

Property	Description
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated



# Background-properties

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

# Background-color

- With CSS, a color is most often specified by:

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

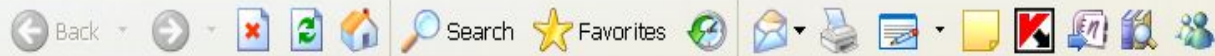
# Example


```
<html>
<head>
<style type="text/css">
  h1 {background-color:#6495ed;}
  p {background-color:#e0ffff;}
  div {background-color:#b0c4de;}
</style> </head>

<body>
<h1> CSS Heading</h1>
<p> CSS paragraph </p>
</body> </html>
```

C:\xampp\htdocs\program\_php\csstest.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help



Address  C:\xampp\htdocs\program\_php\csstest.html

# CSS Heading

CSS paragraph

# background-image

- Example
- ```
body {  
    background-image: url("one.gif");  
}
```
- By default, the image is repeated so it covers the entire element.

# background-repeat

- background-repeat
  - repeat
  - repeat-x
  - repeat-y
  - no-repeat

# Example

- <html>
- <head>
- <style>
- body
- {
- background-image : url('sunset.jpg');
- background-repeat:no-repeat;
- }
- </style>
- </head>
  
- <body>
- <h1>Hello World!</h1>
- </body>
- </html>



**Hello World!**





# Backgroundexample.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Background Properties Example</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="background-example">
    <h2>Welcome to the Background Properties Example</h2>
    <p>This <strong>div</strong> demonstrates different CSS background properties.</p>
  </div>

</body>
</html>
```

# style.css

```
body {  
  margin: 0;  
  padding: 0;  
  font-family: Arial, sans-serif;  
}  
  
.background-example {  
  width: 100%;  
  height: 400px;  
  padding: 20px;  
  text-align: center;  
  color: white;  
  font-size: 18px;  
  font-weight: bold;  
  
  background-color: rgba(0, 0, 0, 0.5); /* semi-transparent black */  
  background-image: url('https://via.placeholder.com/800x600'); /* Image URL */  
  background-repeat: no-repeat; /* Don't repeat the background image */  
  background-attachment: fixed; /* Keeps the background fixed during scrolling */  
  background-position: center center; /* Centers the background image */  
}
```



CSS Text

1. **font-family:**
  - Specifies the font type. For example, the heading uses 'Times New Roman', and the paragraph uses 'Verdana'.
2. **font-size:**
  - Specifies the size of the text. For example, the heading is larger (36px), and the paragraph is smaller (18px).
3. **font-weight:**
  - Sets the weight of the font (bold, normal, etc.). The heading is bold, while the paragraph is normal weight.
4. **text-transform:**
  - Controls the capitalization of the text. In the heading, uppercase is used to make the text all capital letters, and in the paragraph, capitalize is used to capitalize the first letter of each word.
5. **letter-spacing:**
  - Adjusts the spacing between letters. In the heading, it's set to 2px, and in the paragraph, it's 1px.
6. **text-shadow:**
  - Adds a shadow effect to the text. The heading has a shadow with a 2px horizontal and vertical offset and a blur of 4px, using rgba(0, 0, 0, 0.3) for the shadow color.
7. **color:**
  - Defines the color of the text. The heading is a dark blue (#2c3e50), and the paragraph is a lighter grey-blue (#34495e).
8. **line-height:**
  - Adjusts the line spacing in the paragraph to make the text more readable (set to 1.8 here).
9. **text-align:**
  - Controls the alignment of the text. The heading is centered by default, and the paragraph is justified to create even margins.
10. **text-decoration:**
  - Adds a decoration to the text, such as underline. The paragraph will be underlined when hovered over (text-decoration: underline;).
11. **text-align: justify:**
  - Justifies the text in the paragraph so it aligns evenly on both the left and right margins.

# CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text

# CSS Text Properties

Property	Description
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text

# CSS Text Properties

Property	Description
vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in a text

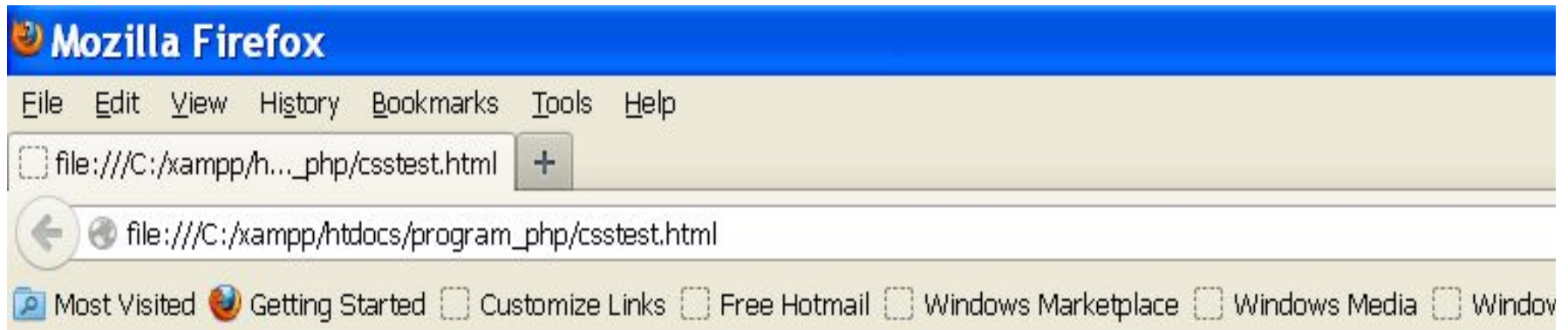


# Example

```
<html>
<head>
<style>

a {
  text-decoration:blink;
  text-align:center;
  text-transform:capitalize;
  text-shadow:#993366;
}
</style>
</head>
<body>
  <a href="http://www.cdacnoida.com">cdacnoida.com</a>
</body>
</html>
```

# Output



[Cdacnoida.com](http://Cdacnoida.com)

# Text.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Text Properties Example</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="text-example">
    <h1>Styled Text Example</h1>
    <p>This paragraph demonstrates various CSS text properties.</p>
  </div>

</body>
</html>
```

Text.css

```
/* Basic body styling */
```

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
}
```

```
/* Text example div */
```

```
.text-example {  
    width: 80%;  
    margin: 50px auto;  
    padding: 20px;  
    background-color: #fff;  
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);  
    text-align: center;  
}
```

```
/* Heading styling */
```

```
.text-example h1 {  
    font-family: 'Times New Roman', serif;  
    font-size: 36px;          /* Font size */  
    font-weight: bold;        /* Font weight */  
    text-transform: uppercase; /* Uppercase text */  
    color: #2c3e50;           /* Text color */  
    letter-spacing: 2px;       /* Letter spacing */  
    margin-bottom: 20px;  
}
```

```
/* Hover effect for paragraph */
```

```
.text-example p:hover {  
    color: #e74c3c;           /* Change text color on hover */  
    text-decoration: underline; /* Underline text on hover */  
}
```

CSS Font:

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font



# Example



The image shows a web browser window with a single tab titled "Design". The address bar is empty. The browser's toolbar includes icons for back, forward, home, search, and other standard navigation functions. The main content area displays the following HTML and CSS code:

```
<html>
<head>
<style>
h1    {
    font:Arial, Helvetica, sans-serif;
    font-style:italic;
    font-weight:500;
    font-stretch:narrower;
    font-size:40px;
    }
</style>
</head>
<body>
    <h1>CSS font-family</h1>

</body>
</html>
```

# Output



*CSS font-family*

# CSS Lists

- 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

# An Image as The List Item Marker

- `<style>`
- `ul`
- `{ list-style-image:url('bullet1.gif'); }`
- `</style> </head>`
- `<body>`
- `<ul>`
- `<li>one</li>`
- `<li>two</li>`
- `</ul>`

# listing.html

```
<!DOCTYPE html>
<html lang="en">
<body>
  <div class="course-list-container">
    <h1>Available Courses</h1>
    <!-- Unordered list of courses -->
    <ul class="course-list">
      <li>Web Development</li>
      <li>Data Science</li>
      <li>Machine Learning</li>
      <li>Cloud Computing</li>
      <li>Cyber Security</li>
    </ul>
    <!-- Ordered list of courses -->
    <h2>Popular Courses (Ordered)</h2>
    <ol class="course-list-ordered">
      <li>Artificial Intelligence</li>
      <li>Blockchain Technology</li>
      <li>Python Programming</li>
      <li>Digital Marketing</li>
      <li>UX/UI Design</li>
    </ol>
  </div>

</body>
</html>
```

listing.css

```
/* General body styling */
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f9;
    margin: 0;
    padding: 0;
}

/* Container for course list */
.course-list-container {
    width: 80%;
    margin: 50px auto;
    padding: 20px;
    background-color: #ffffff;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
}

/* Styling for course list heading */
h1 {
    text-align: center;
    color: #2c3e50;
    font-size: 2rem;
}

/* Unordered list styling */
.course-list {
    list-style-type: none; /* Remove default bullet points */
    padding-left: 0;
}
```

```
.course-list li {
    background-color: #3498db; /* Blue background for each item */
    color: #fff; /* White text color */
    padding: 10px;
    margin-bottom: 10px; /* Space between list items */
    border-radius: 5px; /* Rounded corners for each list item */
    font-size: 1.2rem;
    transition: all 0.3s ease; /* Smooth transition on hover */
}

.course-list li:hover {
    background-color: #2980b9; /* Darker blue when hovering */
    transform: translateY(-5px); /* Slight lift effect on hover */
}

/* Styling for ordered list */
.course-list-ordered {
    list-style-type: decimal; /* Default numbering style */
    padding-left: 20px; /* Indentation for ordered list */
}

.course-list-ordered li {
    background-color: #2ecc71; /* Green background for each item */
    color: #fff; /* White text color */
    padding: 10px;
    margin-bottom: 10px; /* Space between list items */
    border-radius: 5px; /* Rounded corners for each list item */
    font-size: 1.2rem;
    transition: all 0.3s ease; /* Smooth transition on hover */
}

.course-list-ordered li:hover {
    background-color: #27ae60; /* Darker green when hovering */
    transform: translateY(-5px); /* Slight lift effect on hover */
}
```

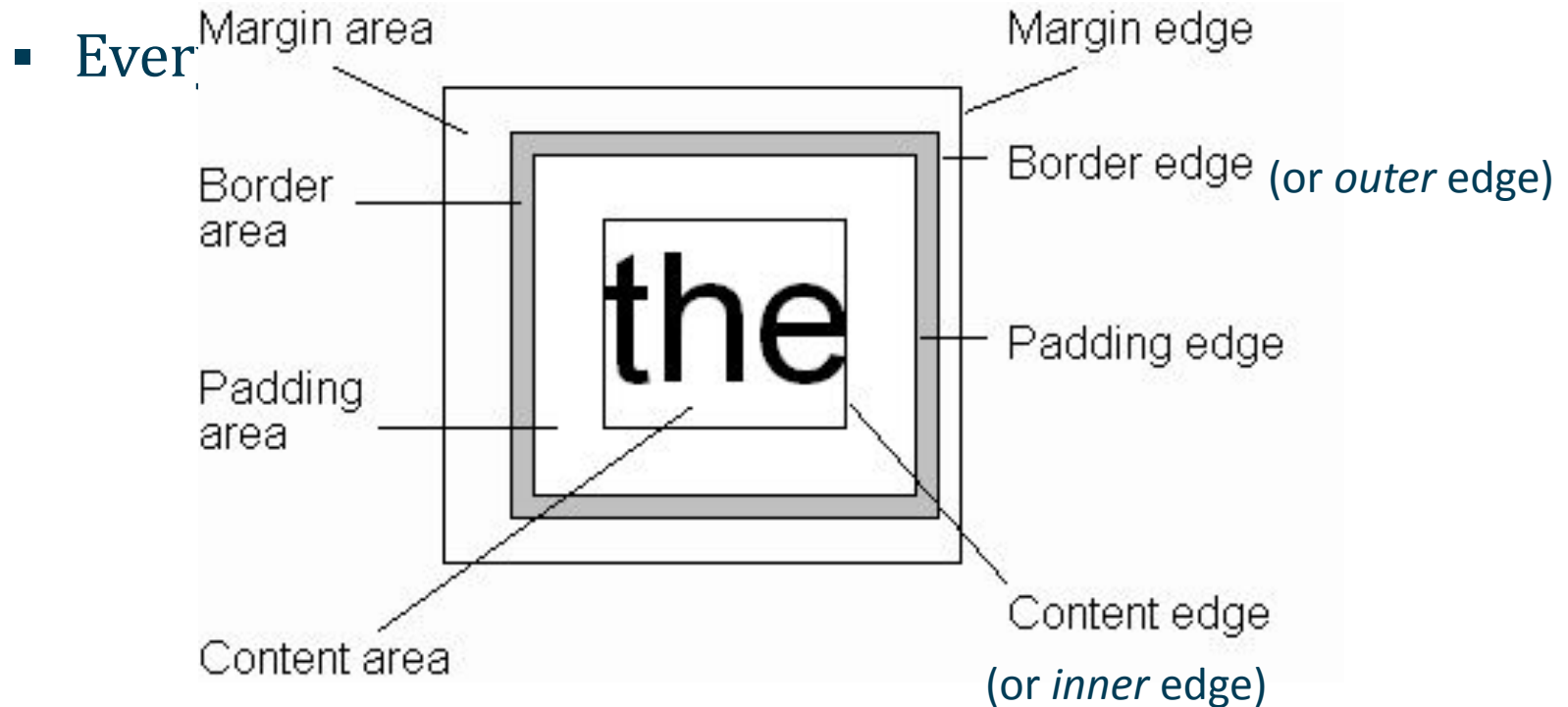


# CSS Box Model

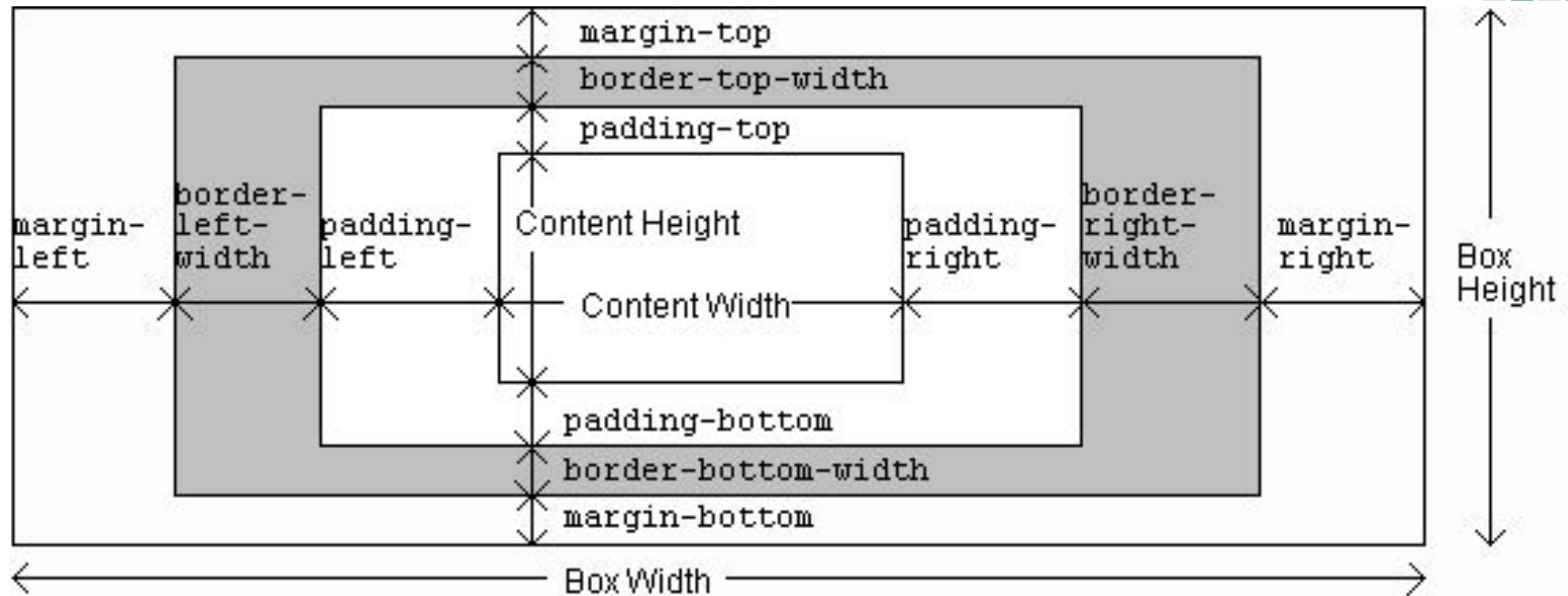
# The CSS Box Model

- All HTML elements can be considered as boxes.
- "box model" is used when talking about design and layout.
- It allows us to put a border around elements and space elements in relation to other elements.

# CSS Box Model

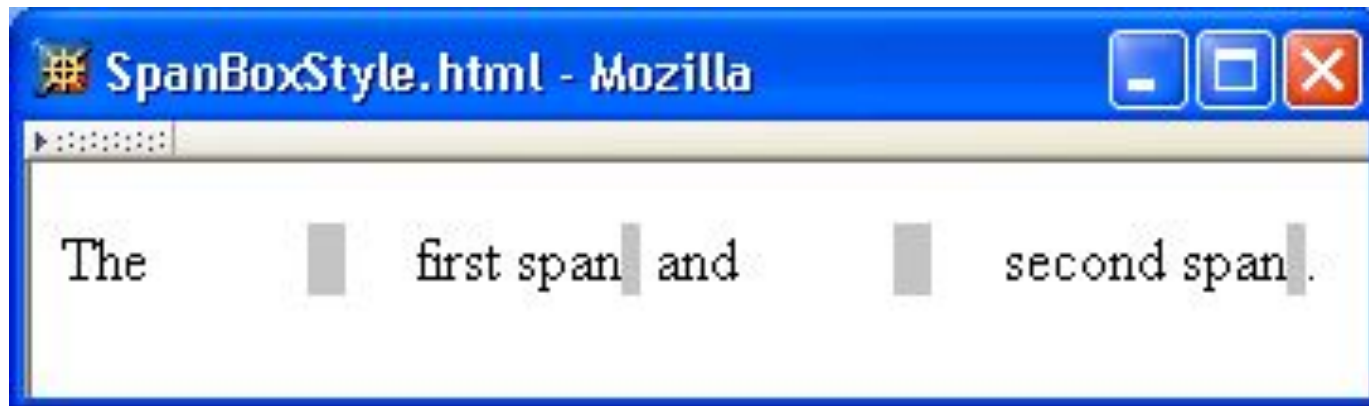


# CSS Box Model



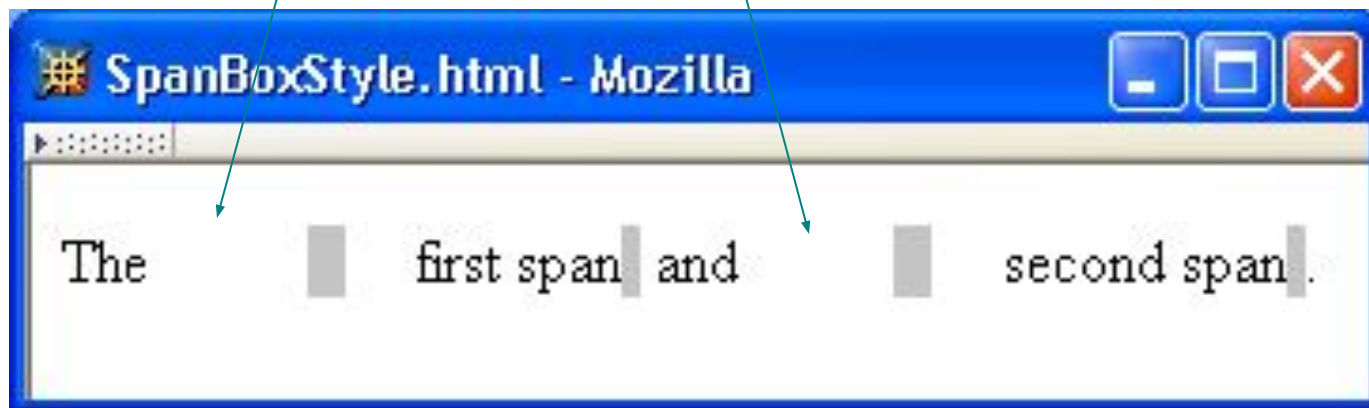
# CSS Box Model

```
span { margin-left: 1cm;  
        border-left-width: 10px;  
        border-left-color: silver;  
        border-left-style: solid;  
        padding-left: 0.5cm;  
        border-right-width: 5px;  
        border-right-color: silver;  
        border-right-style: solid }
```



# CSS Box Model

```
span { margin-left: 1cm;  
        border-left-width: 10px;  
        border-left-color: silver;  
        border-left-style: solid;  
        padding-left: 0.5cm;  
        border-right-width: 5px;  
        border-right-color: silver;  
        border-right-style: solid }
```



# CSS Box Model

```
span { margin-left: 1cm;  
        border-left-width: 10px;  
        border-left-color: silver;  
        border-left-style: solid;  
        padding-left: 0.5cm;  
        border-right-width: 5px;  
        border-right-color: silver;  
        border-right-style: solid }
```

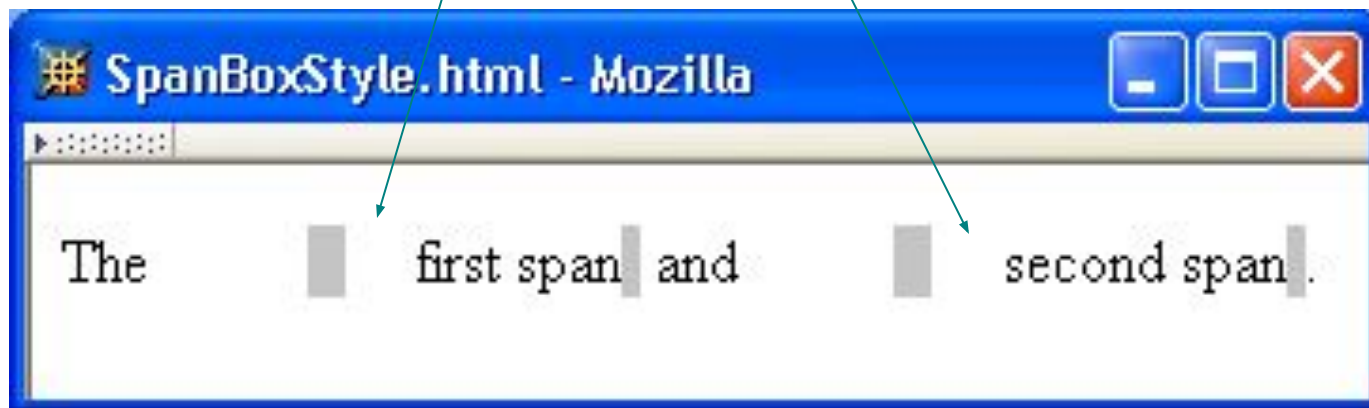


TABLE 3.9: Basic CSS style properties associated with the box model.

Property	Values
<code>padding-{top,right,bottom,left}</code>	CSS length (Sec. 3.6.2).
<code>padding</code>	One to four length values (see text).

TABLE 3.10: Meaning of values for certain shorthand properties that take one to four values.

Number of values	Meaning
One	Assign this value to all four associated properties ( <code>top</code> , <code>right</code> , <code>bottom</code> , and <code>left</code> ).
Two	Assign first value to associated <code>top</code> and <code>bottom</code> properties, second value to associated <code>right</code> and <code>left</code> properties.
Three	Assign first value to associated <code>top</code> property, second value to <code>right</code> and <code>left</code> , and third value to <code>bottom</code> .
Four	Assign first value to associated <code>top</code> property, second to <code>right</code> , third to <code>bottom</code> , and fourth to <code>left</code> .



# CSS Box Model

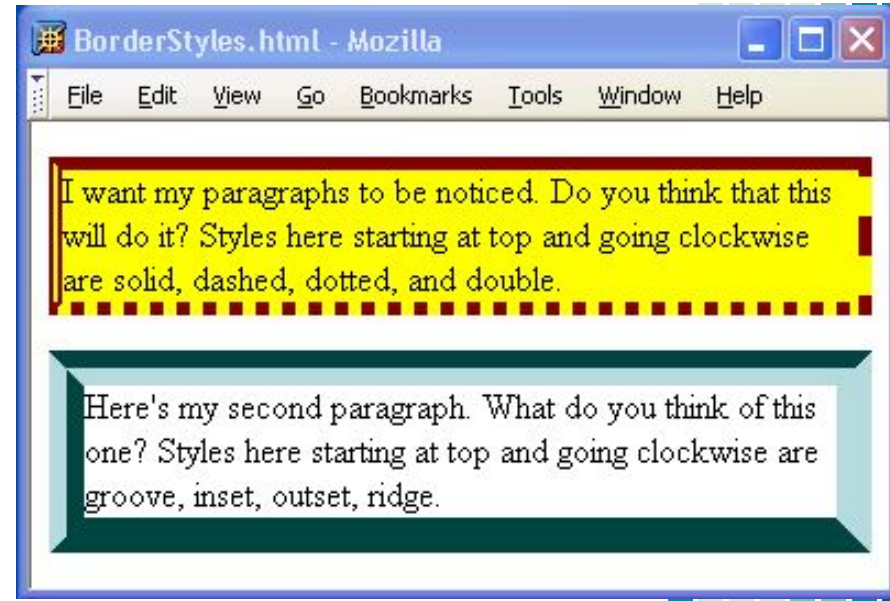
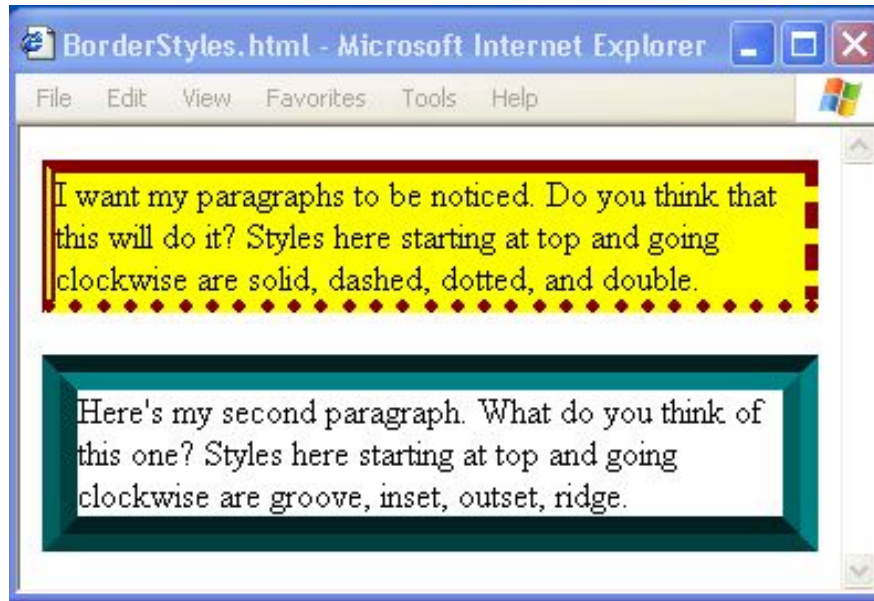
TABLE 3.9: Basic CSS style properties associated with the box model.

<code>border-{top,right,bottom,left}-width</code>	thin, medium (initial value), thick, or a length.
<code>border-width</code>	One to four <code>border-*-width</code> values.
<code>border-{top,right,bottom,left}-color</code>	Color value. Initial value is value of element's color property.
<code>border-color</code>	transparent or one to four <code>border-*-color</code> values.

# CSS Box Model

TABLE 3.9: Basic CSS style properties associated with the box model.

<code>border-{top,right,bottom,left}-style</code>	none (initial value), hidden, dotted, dashed, solid, double, groove, ridge, inset, outset.
<code>border-style</code>	One to four border-*-style values.



# CSS Box Model

TABLE 3.9: Basic CSS style properties associated with the box model.

<code>border-{top,right,bottom,left}</code>	One to three values (in any order) for <code>border-*-width</code> , <code>border-*-color</code> , and <code>border-*-style</code> . Initial values are used for any unspecified values.
<code>border</code>	One to three values; equivalent to specifying given values for each of <code>border-top</code> , <code>border-right</code> , <code>border-bottom</code> , and <code>border-left</code> .
<code>margin-{top,right,bottom,left}</code>	<code>auto</code> (see text) or length.
<code>margin</code>	One to four <code>margin-*</code> values.

# CSS Box Model

- If multiple declarations apply to a property, the last declaration overrides earlier specifications

```
{ border: 15px solid;  
  border-left: 30px inset red;  
  color: blue }
```

Left border is 30px wide,  
inset style, and red

# CSS Dimension

- The CSS dimension properties allow you to control the height and width of an element.

```
<style>  
img  
{  
    width:200px;  
    height:200px;}  
</style>
```

# CSS Positioning

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

There are four different positioning methods.

- **Static Positioning:** HTML elements are positioned static by default.
- **Fixed Positioning:** An element with fixed position is positioned relative to the browser window. It will not move even if the window is scrolled:

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

# Continued...

## **Relative Positioning:**

A relative positioned element is positioned relative to its normal position.

**Example:** `h2 { position: relative; left:-20px; }`

## **Absolute Positioning:**

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

**Example:** `h2 { position: absolute; left:100px; top:150px; }`

## **Overlapping Elements**

The z-index property specifies the stack order of an element.

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

# Attributes

- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border** - A border that goes around the padding and content. The border is affected by the background color of the box



# Attributes

- **Padding** - Clears an area around the content. The padding is affected by the background color of the box
- **Content** - The content of the box, where text and images appear

# Sample css

- body {
  - font-family: "Times New Roman", Times, serif;
  - font-size: 14px;
  - background-image: url("wall1.jpg");
  - color: #CC0000;
- }
- h1 {
  - text-align: center;
  - color: light red;
  - text-shadow: 5px 5px 5px #FF0000;
- }
- h2 {
  - text-align: left;
  - color: red;
- }
- p {
  - text-align: center;
  - color: red;
- }

table, th, td {

- border: 1px solid black;
- background-color: #CCCCCC;
- width: 100%;

▪ }

▪ th{

- text-align: center;
- height: 50px;
- vertical-align: middle;
- }

▪ form {

- position: relative;
- background-image:url(wall1.jpg);
- width : 200px;
- height : 298px;
- border: solid;
- border-color: #FF2E2E;

▪ }

- a {
- display: block;
- width: 60px;
- }
  
- li {
- height: 20px;
  
- width: 200px;
  
- float: left;
- color: #FFFFFF;
- text-shadow: 5px 5px 5px #FF0000;
- background: -moz-linear-gradient(50deg, red, yellow); /\* For Firefox 3.6 to 15 \*/
- padding: 10px;
  
- }

# Css id attribute

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p#p01 {`
- `color: blue;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<p>This is a paragraph.</p>`
- `<p>This is a paragraph.</p>`
- `<p>This is a paragraph.</p>`
- `<p id="p01">I am different.</p>`
- `</body>`
- `</html>`

# Css class attribute

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<style>`
- `p.error {`
- `color:red;`
- `}`
- `</style>`
- `</head>`
- `<body>`
- `<p>This is a paragraph.</p>`
- `<p>This is a paragraph.</p>`
- `<p class="error">I am different.</p>`
- `<p>This is a paragraph.</p>`
- `<p class="error">I am different too.</p>`
- `</body>`
- `</html>`

# css form design

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Styled Form</title>
```

```
<style>
```

```
/* Form container */
```

```
form {
```

```
  max-width: 400px;
```

```
  margin: 50px auto;
```

```
  padding: 20px;
```

```
  background: #f9f9f9;
```

```
  border-radius: 12px;
```

```
  box-shadow: 0px 4px 8px rgba(0,0,0,0.1);
```

```
  font-family: Arial, sans-serif;
```

```
}
```

```
/* Form labels */  
label {  
    display: block;  
    margin-bottom: 8px;  
    font-weight: bold;  
    color: #333;  
}
```

```
/* Input fields */  
input[type="text"],  
input[type="email"],  
input[type="password"],  
textarea,  
select {  
    width: 100%;  
    padding: 10px;  
    margin-bottom: 15px;  
    border: 1px solid #ccc;  
    border-radius: 8px;  
    font-size: 14px;  
    transition: border 0.3s;  
}
```

```
/* Focus effect */  
input:focus,  
textarea:focus,  
select:focus {  
    border-color: #007bff;
```



# Display

## Common Display Types

### 1. **display: block;**

- Takes up the full width of its parent (unless width is set).
- Always starts on a new line.
- You can set **width**, **height**, **margin**, **padding**.
- Examples: `<div>`, `<p>`, `<h1>` ... are block by default.

### **display: inline;**

- Does **not** start on a new line.
- Only takes up as much width as the content.
- You **cannot** set width/height directly.
- Examples: `<span>`, `<a>`, `<strong>` are inline by default.

## **display: inline-block;**

- Behaves like **inline** (sits side by side).
- But allows setting **width**, **height**, **margin**, **padding** (like block).
- Very useful for buttons, menus, grids.

## **display: flex;**

- Turns an element into a **flex container**.
- Its children become **flex items**, which can be aligned, spaced, and ordered easily.
- Great for responsive layouts.


## **display: none;**

- Hides the element completely (it won't take up space).
- Different from **visibility: hidden;** (which hides it but still takes space).

```
/* Submit button */
input[type="submit"], button {
  width: 100%;
  padding: 12px;
  background: #007bff;
  color: white;
  border: none;
  border-radius: 8px;
  font-size: 16px;
  cursor: pointer;
  transition: background 0.3s;
}
```

```
input[type="submit"]:hover, button:hover {
  background: #0056b3;
}
</style>
</head>
<body>
```

### Other useful **cursor** values:

- **default** → normal arrow
- **text** → text selection (I-beam cursor)
- **wait** → loading spinner/hourglass
- **not-allowed** →  symbol (disabled)
- **grab / grabbing** → hand for draggable items

```
<form>
```

```
  <h2>Sign Up</h2>
```

```
  <label for="name">Full Name</label>
```

```
  <input type="text" id="name" name="name" placeholder="Enter your full name">
```

```
  <label for="email">Email Address</label>
```

```
  <input type="email" id="email" name="email" placeholder="Enter your email">
```

```
  <label for="password">Password</label>
```

```
  <input type="password" id="password" name="password" placeholder="Enter a password">
```

```
  <label for="message">Message</label>
```

```
  <textarea id="message" name="message" rows="3" placeholder="Write something..."></textarea>
```

```
  <input type="submit" value="Register">
```

```
</form>
```

```
</body>
```

