

Lecture Three:

HTML Forms & introduction to CSS

COURSE TITLE: WEB PROGRAMMING 1

TOPIC: HTML FORMS & INTRODUCTION TO CSS

SEMESTER: 1 2025/2026

DURATION: 10 WEEKS

HTML Forms

HTML form is used to **collect** user input and **sent** to a server for processing

example

First name:

John

Last name:

Doe

Submit

The <form> Element

The HTML <form> element is used to **create** an HTML form for user input:



```
<form>  
.  
form elements  
.  
</form>
```

The <input> Element

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the **type** attribute.

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

The `<label>` Element

The `<label>` tag defines a label for many form elements.

Text Fields

This is how the HTML code will be displayed in a browser:

First name:

Last name:

Text Fields

The `<input type="text">` defines a single-line input field for text input.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

The **for** attribute of the `<label>` tag should be **equal to** the **id** attribute of the `<input>` element.

Radio Buttons

This is how the HTML code above will be displayed in a browser:

Choose your favorite Web language:

- ☐ HTML
- ☐ CSS
- ☐ JavaScript

Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select **ONE** of a **limited** number of choices.



`<p>Choose your favorite Web language:</p>`

`<form>`

`<input type="radio" id="html" name="fav_language" value="HTML">`

`<label for="html">HTML</label><br>`

`<input type="radio" id="css" name="fav_language" value="CSS">`

`<label for="css">CSS</label><br>`

`<input type="radio" id="javascript" name="fav_language" value="JavaScript">`

`<label for="javascript">JavaScript</label>`

`</form>`

Checkboxes

This is how the HTML code above will be displayed in a browser:

- ☐ I have a bike
- ☐ I have a car
- ☐ I have a boat

Checkboxes

The `<input type="checkbox">` defines a checkbox.

Checkboxes let a user select **ZERO** or **MORE** options of a limited number of choices.



```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

The Submit Button

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

The Name Attribute for <input>

Notice that each input field must have a name attribute to be submitted.

If the **name** attribute is **omitted**, the value of the input field will **not be sent at all. IMPORTANT!**



```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" value="John"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

This example will not submit the value of the "First name" input field:

The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window

```
<form action="/action_page.php" target="_blank">
```

Here, the submitted result will open in a new browser tab:

The <select> Element

The **<select>** element defines a drop-down list:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

`<option value="fiat" selected>Fiat</option>`

To define a pre-selected option, add the selected attribute to the option:

The select Element

The select element defines a drop-down list:

Choose a car:

Saab
Fiat
Audi

The <textarea> Element

The `<textarea>` element defines a multi-line input field (a text area):



```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```

Textarea

The textarea element defines a multi-line input field.

Introduction to CSS

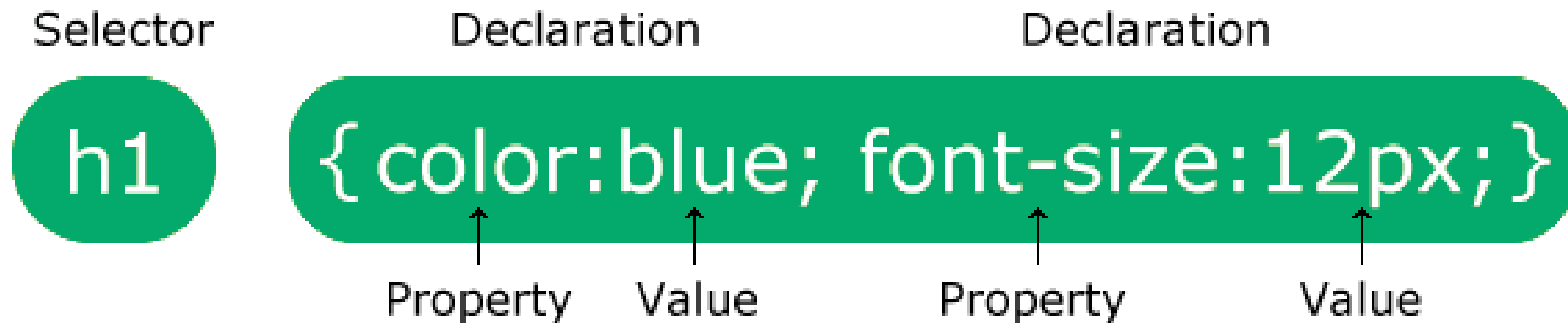
CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.



CSS Syntax

A CSS rule consists of a **selector** and a **declaration** block.



Example



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

- **p** is a selector in CSS
(It points to the HTML element you want to style: <p>)
- **color** is a property, and red is the **property value**
- **text-align** is a property, and center is the **property value**

The CSS element Selector



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

The Simple Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>
|
</body>
</html>
```

Hello World!

These paragraphs are styled with CSS.

The CSS id Selector



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

The CSS id Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>
```

Hello World!

This paragraph is not affected by the style.

The CSS class Selector



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



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

You can also specify that only specific HTML elements should be affected by a class.

The CSS class Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

Red and center-aligned heading

Red and center-aligned paragraph.

The CSS Grouping Selector

Option 1

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

Option 2

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

The CSS Grouping Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>

</body>
</html>
```

Hello World!

Smaller heading!

This is a paragraph.

CSS Colors & CSS Backgrounds & border

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

```
<h1 style="color:Tomato;">Hello World</h1>
```

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

CSS Colors & CSS Backgrounds & border

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

<h1 style="background-color:DodgerBlue;">Hello World</h1>
<h1 style="color:Tomato;">Hello World</h1>
<h1 style="border: 2px solid Tomato;">Hello World</h1>

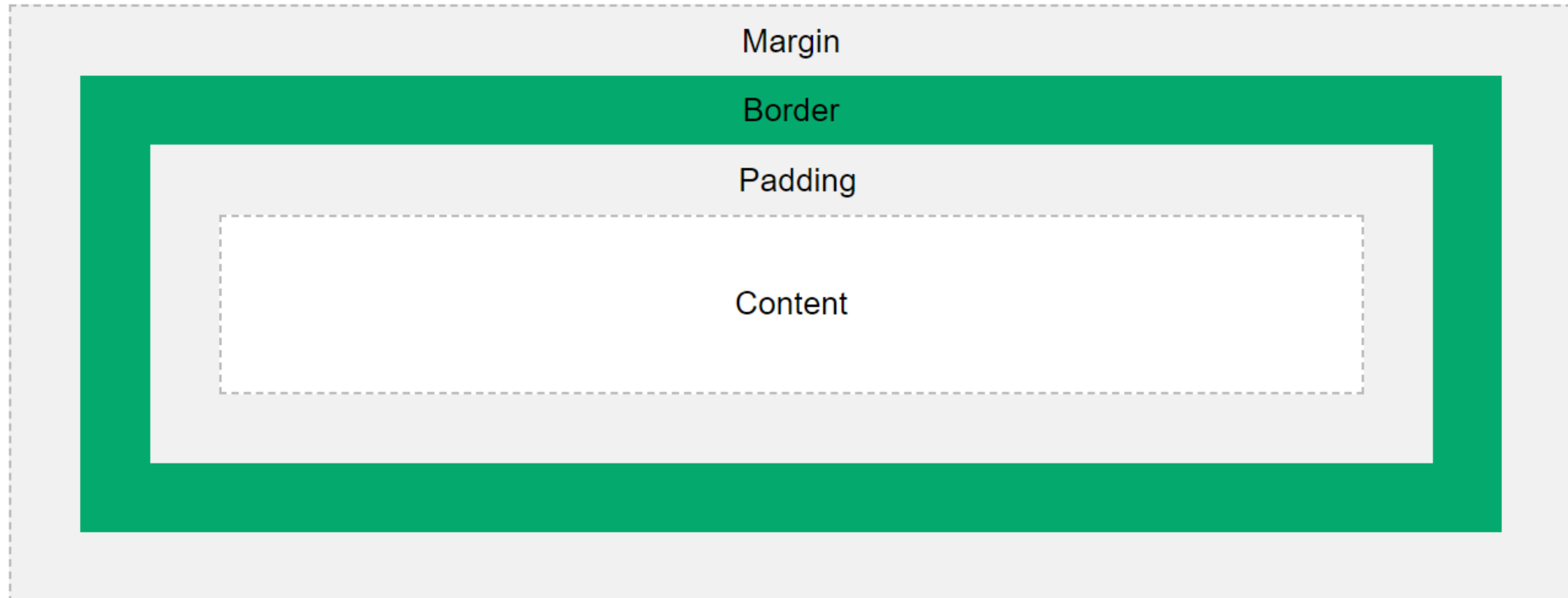
</p>
</body>
</html>
```

Hello World

Hello World

Hello World

CSS Margins



CSS Margins

The CSS margin properties are used to create space around elements, outside of any defined borders.

margin-top

margin-right

margin-bottom

margin-left

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

CSS Margins

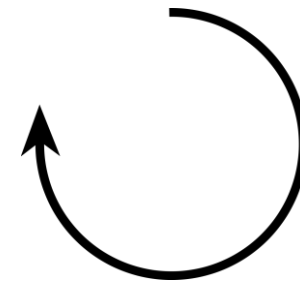
Use the margin shorthand property with **four** values:



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

• **margin: 25px 50px 75px 100px;**

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



CSS Margins

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>Using individual margin properties</h2>

<div>This div element has a top margin of 100px, a right margin of 150px, a bottom
margin of 100px, and a left margin of 80px.</div>

</body>
</html>
```

Using individual margin properties

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

CSS Margins

If the margin property has **three** values:

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

- **margin: 25px 50px 75px;**
 - top margin is 25px
 - right and left margins are 50px
 - bottom margin is 75px

CSS Margins

If the margin property has **two** values:



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

- **margin: 25px 50px;**
 - top and bottom margins are 25px
 - right and left margins are 50px

CSS Margins

If the margin property has **one** value:



```
p {  
  margin: 25px;  
}
```

- **margin: 25px;**
 - all four margins are 25px

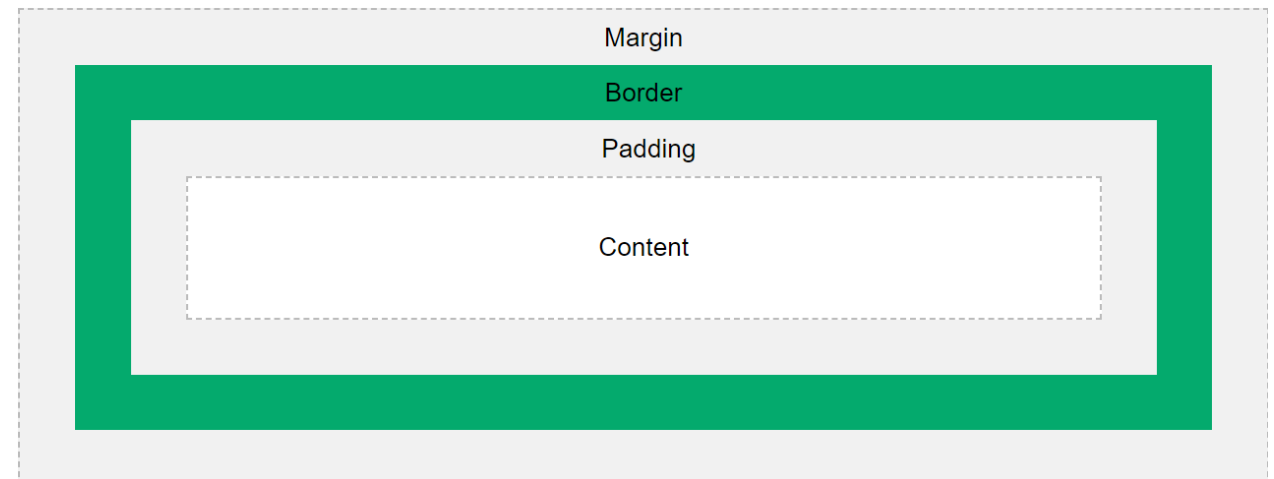
CSS Padding

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

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

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

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



CSS Padding

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>

<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a bottom
padding of 50px, and a left padding of 80px.</div>

</body>
</html>
```

Using individual padding properties

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

CSS Height, Width and Max-width

The CSS height and width properties are used to set the height and width of an element.

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

```
div {  
  height: 100px;  
  width: 500px;  
  background-color: powderblue;  
}
```

CSS Height, Width and Max-width

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>

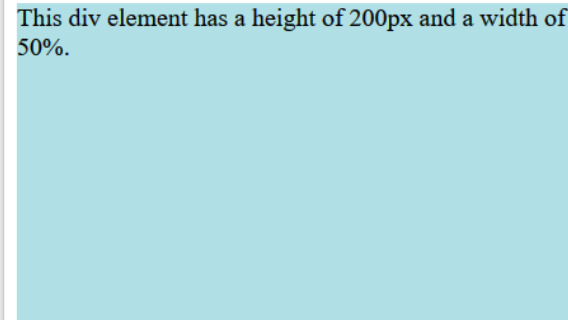
<h2>Set the height and width of an element</h2>

<div>This div element has a height of 200px and a width of 50%.</div>

</body>
</html>
```

Set the height and width of an element

This div element has a height of 200px and a width of 50%.

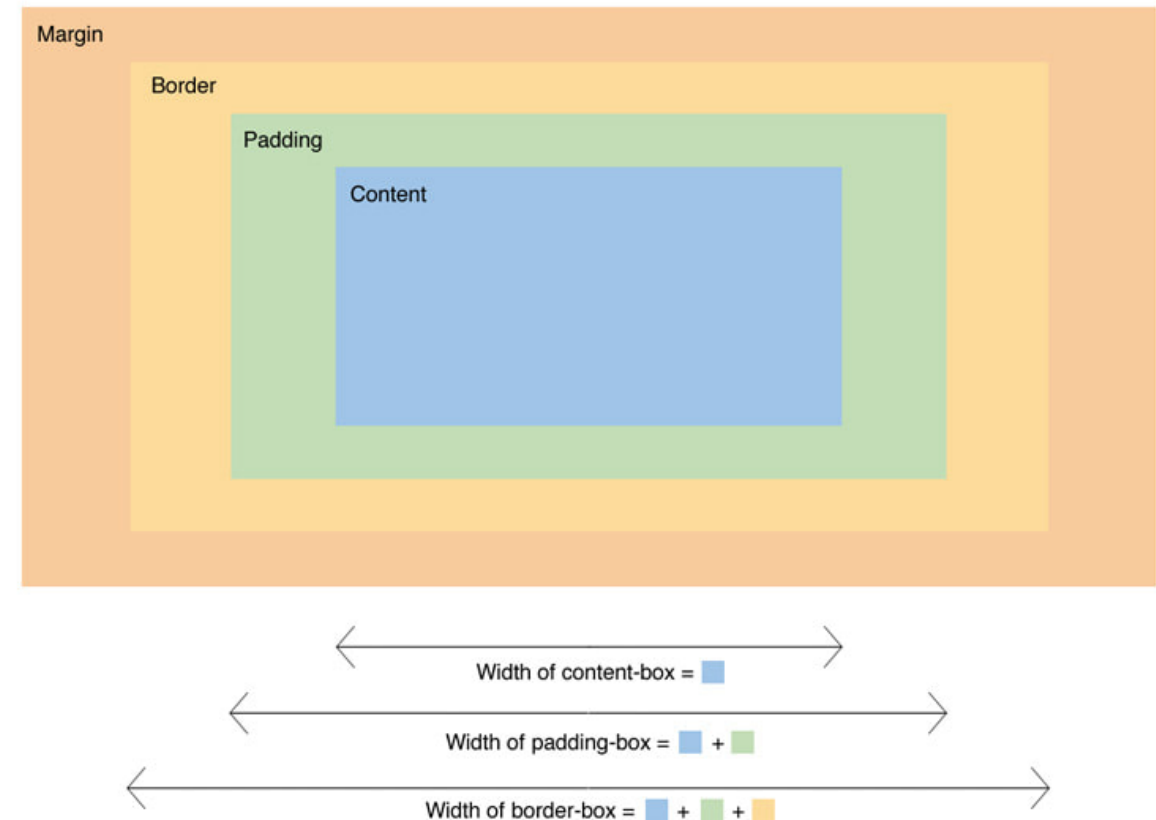


Box Model

- Box-sizing property for controlling how the width and height are calculated.

```
div {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px



Box Model (**Very Important**)

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Box Model

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

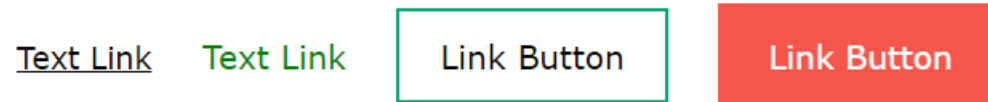
Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CSS Links

With CSS, links can be styled in many different ways.



```
a {  
  color: hotpink;  
}
```

When setting the style for several link states, there are some order rules:

a:hover MUST come after a:link and a:visited
a:active MUST come after a:hover

```
/* 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;  
}
```

References

- Mozilla Developer Network (MDN) - developer.mozilla.org
- W3Schools - www.w3schools.com

THANK
YOU