

BASICS OF WEB DEVELOPMENT WITH HTML, CSS AND JAVASCRIPT

A hand is holding a smartphone, with its screen visible in the bottom left corner. The screen displays a presentation slide with a dark blue background and white text. The text on the slide reads "BASICS OF WEB DEVELOPMENT WITH HTML, CSS AND JAVASCRIPT". In the background of the slide, there is a faint, semi-transparent watermark of a person's face. The overall composition suggests a person is presenting or demonstrating web development concepts.

types.Operator):
 X mirror to the selected
 object.mirror_mirror_x"
 for X"

FUNDAMENTALS OF WEB DEVELOPMENT WITH HTML

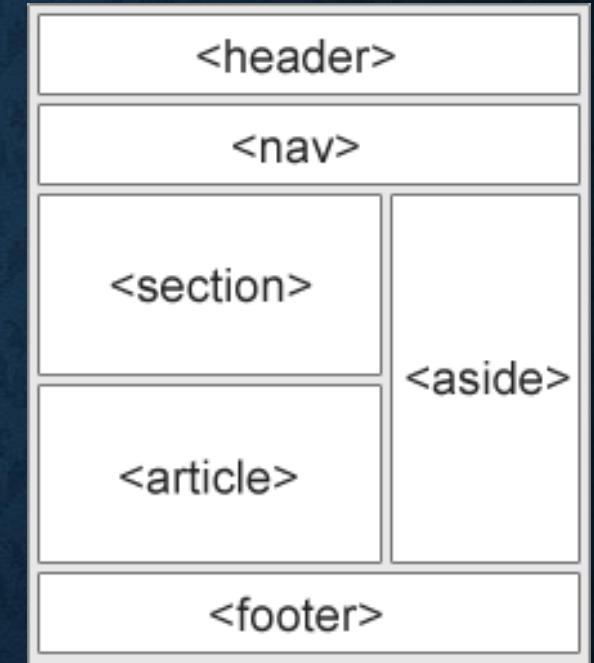
Topics to be covered:

- HTML Layout Elements and Techniques
- HTML class and ID Attribute
- HTML Forms
- HTML Media
- HTML Graphics

HTML LAYOUT ELEMENTS AND TECHNIQUES

HTML Layout Elements:

1. <header> - Defines a header for a document or a section
2. <nav> - Defines a set of navigation links
3. <section> - Defines a section in a document
4. <article> - Defines an independent, self-contained content
5. <aside> - Defines content aside from the content (like a sidebar)
6. <footer> - Defines a footer for a document or a section
7. <details> - Defines additional details that the user can open and close on demand
8. <summary> - Defines a heading for the <details> element



HTML LAYOUT TECHNIQUES

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

1. **CSS framework:** If you want to create your layout fast, you can use a CSS framework, like Bootstrap, Material UI etc.
2. **CSS float property:** It is common to do entire web layouts using the CSS float property. Float is easy to learn - you just need to remember how the float and clear properties work.
3. **CSS flexbox:** Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.
4. **CSS grid:** The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

CLASS ASSIGNMENT 1

Float one

In this task, you need to float the two elements with a class of `float1` and `float2` left and right, respectively. The text should then appear between the two boxes, as in the image below.

One

The two
boxes should
float to either
side of this
text.

Two

Float two

In this example, the element with a class of `float` should be floated left. Then we want the first line of text to display next to that element, but the following line of text (which has a class of `.below`) to display underneath it. You can see the desired result in this image.

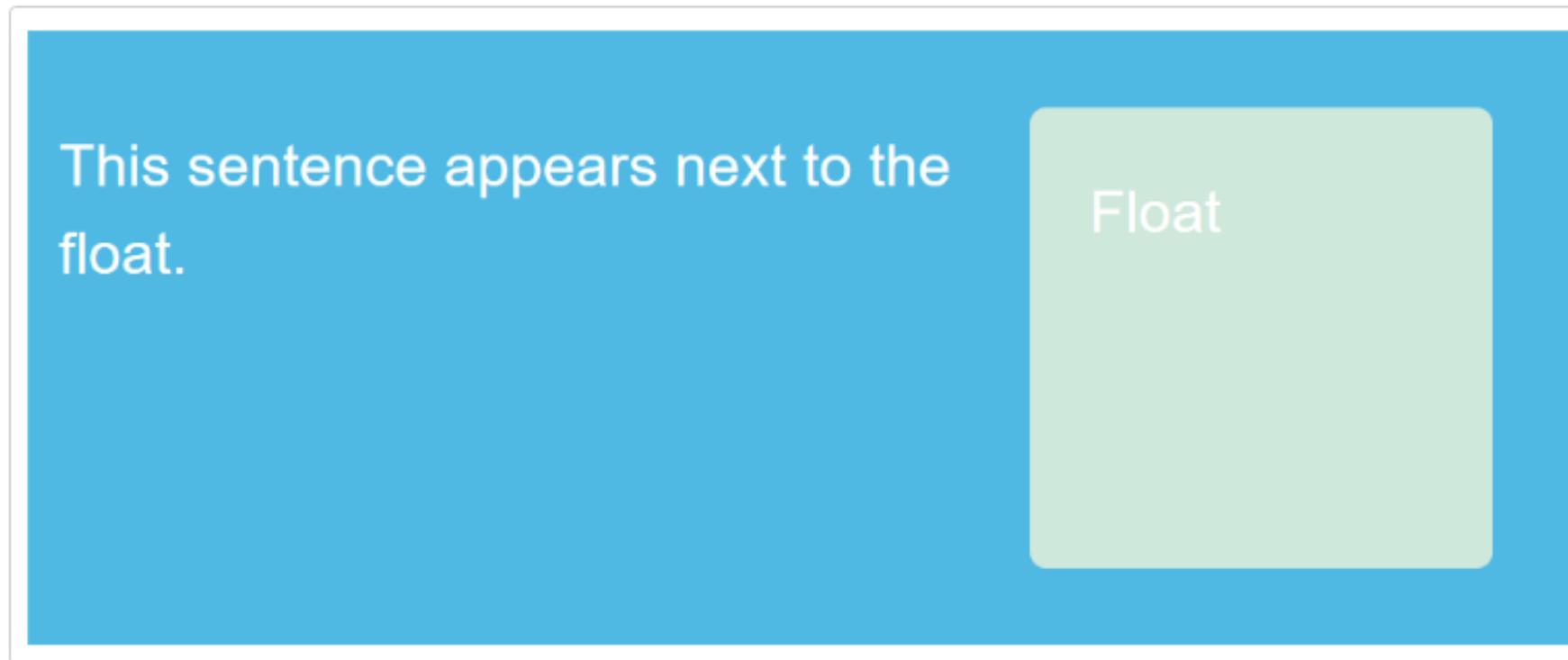
Float

This sentence appears next to the float.

Cause this sentence to appear below the float.

Float three

Finally, we have a floated element in this example. The box wrapping the float and text is displaying behind the float. Use the most up-to-date method available to cause the box background to extend to below the float, as in the image.



HTML CLASS AND ID ATTRIBUTE

- The **HTML class attribute** is used to specify a class for an HTML element.
- Multiple HTML elements can share the same class.
- The **HTML id attribute** is used to specify a unique id for an HTML element.
- You cannot have more than one element with the same id in an HTML document.

Switch to vs code

HTML FORMS

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.
- The HTML <input> element is the most used form element.
- An <input> element can be displayed in many ways, depending on the type attribute.
- The action attribute defines the action to be performed when the form is submitted.

HTML INPUT TYPES

<input
type="button">

<input
type="checkbox">

<input
type="color">

<input
type="date">

<input
type="datetime-
local">

<input
type="email">

<input type="file">

<input
type="hidden">

<input
type="image">

<input
type="month">

<input
type="number">

<input
type="password">

<input
type="radio">

<input
type="range">

<input
type="reset">

<input
type="search">

<input
type="submit">

<input type="tel">

<input
type="text">

<input
type="time">

<input type="url">

<input
type="week">

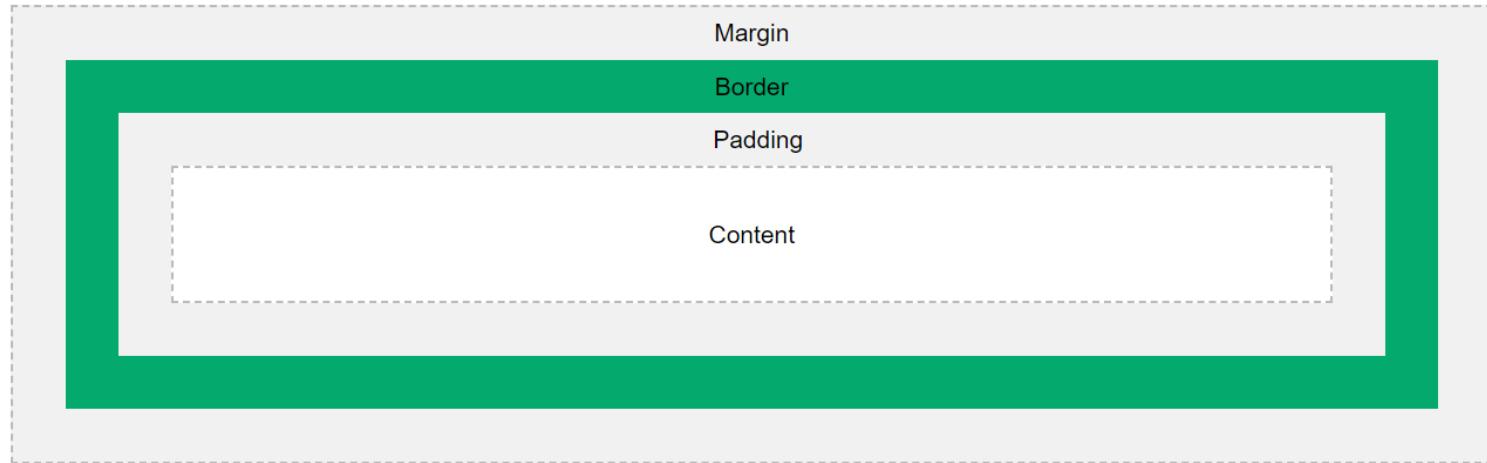
BASICS OF WEB DEVELOPMENT WITH CSS

Topics to be covered:

- The CSS Box Model
- CSS Website Layout - Position Property, z-index Property, Overflow, float and clear
- CSS Navigation Bar
- CSS Pseudo-classes
- CSS Transitions and Animations

THE CSS BOX MODEL

- 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.

CSS WEBSITE LAYOUT - POSITION PROPERTY, Z-INDEX PROPERTY, OVERFLOW, FLOAT AND CLEAR

- https://www.w3schools.com/css/css_positioning.asp
- https://www.w3schools.com/css/css_z-index.asp
- https://www.w3schools.com/css/css_overflow.asp
- https://www.w3schools.com/css/css_float.asp

CSS NAVIGATION BAR

- Vs code

CSS PSEUDO-CLASSES

- A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

```
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

CSS TRANSITIONS AND ANIMATIONS

- CSS transitions allows you to change property values smoothly, over a given duration.
- **Properties:**
 1. transition
 2. transition-delay
 3. transition-duration
 4. transition-property
 5. transition-timing-function

CSS ANIMATIONS

- CSS allows animation of HTML elements without using JavaScript or Flash!

- **Properties:**

1. @keyframes
2. animation-name
3. animation-duration
4. animation-delay
5. animation-iteration-count
6. animation-direction
7. animation-timing-function
8. animation-fill-mode
9. animation

BASICS OF WEB DEVELOPMENT WITH JAVASCRIPT

Topics to be covered:

- JavaScript Variables - Using var, Using let, Using const
- JavaScript Functions, JavaScript Objects
- JavaScript Events
- JavaScript Array Methods
- JavaScript Regular Expressions
- DOM Manipulation
- JavaScript Best Practices

JAVASCRIPT VARIABLES - USING VAR, USING LET, USING CONST

- Most of the time, a JavaScript application needs to work with information. Here are two examples:
 1. An online shop – the information might include goods being sold and a shopping cart.
 2. A chat application – the information might include users, messages, and much more.
- A variable is a “named storage” for data. We can use variables to store goodies, visitors, and other data.

VAR



Variables are containers for storing data (storing data values).



In this example, x, y, and z, are variables, declared with the var keyword:

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

<h2>JavaScript Variables</h2>

<p>In this example, x, y, and z are variables.</p>

<p id="demo"></p>

<script>
let x = 5;
let y = 6;
let z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>

</body>
</html>
```

LET

To create a variable in JavaScript, use the let keyword.

Variables defined with let cannot be Redeclared.

Variables defined with let must be Declared before use.

Variables defined with let have Block Scope.

```
1 let message = 'Hello!'; // define the variable and assign the value  
2  
3 alert(message); // Hello!
```

BLOCK SCOPE

Before ES6 (2015), JavaScript had only Global Scope and Function Scope.

ES6 introduced two important new JavaScript keywords: let and const.

These two keywords provide Block Scope in JavaScript.

Variables declared inside a { } block cannot be accessed from outside the block:

Example

```
{  
  let x = 2;  
}  
// x can NOT be used here
```

Variables declared with the `var` keyword can NOT have block scope.

Variables declared inside a `{ }` block can be accessed from outside the block.

Example

```
{  
  var x = 2;  
}  
// x CAN be used here
```

REDECLARING VARIABLES

Redeclaring a variable using the var keyword can impose problems.



Redeclaring a variable inside a block will also redeclare the variable outside the block.

Example

```
var x = 10;
// Here x is 10

{
var x = 2;
// Here x is 2
}

// Here x is 2
```

Redeclaring a variable using the `let` keyword can solve this problem.

Redeclaring a variable inside a block will not redeclare the variable outside the block:

Example

```
let x = 10;
// Here x is 10

{
let x = 2;
// Here x is 2
}

// Here x is 10
```

JAVASCRIPT CONST

Variables defined with const cannot be Redeclared.

Variables defined with const cannot be Reassigned.

Variables defined with const have Block Scope.

CANNOT BE REASSIGNED AND MUST BE ASSIGNED

A const variable cannot be reassigned:

Example

```
const PI = 3.141592653589793;
PI = 3.14;      // This will give an error
PI = PI + 10;   // This will also give an error
```

JavaScript const variables must be assigned a value when they are declared:

Correct

```
const PI = 3.14159265359;
```

Incorrect

```
const PI;
PI = 3.14159265359;
```

When to use JavaScript const?

As a general rule, always declare a variable with `const` unless you know that the value will change.

Use `const` when you declare:

- A new Array
- A new Object
- A new Function
- A new RegExp

CONSTANT OBJECTS AND ARRAYS

- It does not define a constant value. It defines a constant reference to a value.

Because of this you can NOT:

- Reassign a constant value
- Reassign a constant array
- Reassign a constant object

But you CAN:

- Change the elements of constant array
- Change the properties of constant object

CONSTANT ARRAYS AND CONSTANT OBJECTS

- You can change the elements of a constant array/object, but you can NOT reassign the array/object:

```
// You can create a constant array:  
const cars = ["Saab", "Volvo", "BMW"];  
  
// You can change an element:  
cars[0] = "Toyota";  
  
// You can add an element:  
cars.push("Audi");
```

```
// You can create a const object:  
const car = {type:"Fiat", model:"500", color:"white"};  
  
// You can change a property:  
car.color = "red";  
  
// You can add a property:  
car.owner = "Johnson";
```

```
const cars = ["Saab", "Volvo", "BMW"];  
  
cars = ["Toyota", "Volvo", "Audi"];    // ERROR
```

```
const car = {type:"Fiat", model:"500", color:"white"};  
  
car = {type:"Volvo", model:"EX60", color:"red"};    // ERROR
```

JAVASCRIPT FUNCTIONS, JAVASCRIPT OBJECTS

- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

JAVASCRIPT OBJECTS

- In real life, a car is an object.
- A car has properties like weight and color, and methods like start and stop:

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

JAVASCRIPT EVENTS

- In html, there are various events which represents that some activity is performed by the user or by the browser.
- When JavaScript code is included in HTML, js react over these events and allow the execution. This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.
- For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

Mouse events:

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Window/Document events

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

Click Event

```
<html>
<head> Javascript Events </head>
<body>
<script language="Javascript" type="text/Javascript">
    <!--
        function clickevent()
        {
            document.write("This is JavaTpoint");
        }
        //-->
    </script>
<form>
<input type="button" onclick="clickevent()" value="Who's this?" />
</form>
</body>
</html>
```

MouseOver Event

```
<html>
<head>
<h1> Javascript Events </h1>
</head>
<body>
<script language="Javascript" type="text/Javascript">
    <!--
        function mouseoverevent()
        {
            alert("This is JavaTpoint");
        }
        //-->
    </script>
<p onmouseover="mouseoverevent()"> Keep cursor over me </p>
</body>
</html>
```

JAVASCRIPT ARRAY METHODS

- <https://www.programiz.com/javascript/library/array>
- <https://javascript.info/array-methods>

JAVASCRIPT REGEXP

- A regular expression is a pattern of characters.
- The pattern is used to do pattern-matching "search-and-replace" functions on text.
- In JavaScript, a RegExp Object is a pattern with Properties and Methods.
- **Syntax:** /pattern/modifier(s);
- <https://www.programiz.com/javascript/regex>