

# INTRODUCTION TO JAVASCRIPT

FULL STACK SKILLS BOOTCAMP

# WHAT IS JAVASCRIPT?

- **Lesson Overview:**

- In this lesson, we will be introduced to:

1. JavaScript installation
2. The Browser Console
3. Variables and concatenation
4. Expressions and Conditions
5. Functions and Arguments

# INTRODUCTION TO JAVASCRIPT

## ■ What is JavaScript?

- JavaScript is a programming language that allows you to create interactive and dynamic content on web pages.
- It's used for handling user interactions, manipulating content, and building complex applications.

## ■ Why Learn JavaScript?

- JavaScript is essential for front-end web development.
- Works alongside HTML and CSS.

# INCLUDING JAVASCRIPT IN A WEB PAGE

- **Where to Place JavaScript**

Inside `<script>` tags in the HTML file.

You can also link to an external .js file using the `<script src= "... "></script>` tag.

- **Example:**

```
<html>
<head>
  <title>Simple JS Example</title>
</head>
<body>
  <h1>Welcome to JavaScript</h1>
  <script>
    console.log("Hello, JavaScript!");
  </script>
</body>
</html>
```

# VARIABLES AND DATA TYPES

- **What is a Variable?**

A way to store and manage data in JavaScript.

Variables can be declared using var, let, or const.

- **Data Types in JavaScript:**

**Number:** Represents numeric values (e.g., 10, 3.14).

**String:** Represents text enclosed in quotes (e.g., "Hello", 'World').

**Boolean:** Represents true/false values (e.g., true, false).

```
let name = "Alice"; // String
let age = 25; // Number
let isStudent = true; // Boolean
```

# CONSOLE AND CONSOLE.LOG

## ■ What is the Console?

A tool in web browsers for debugging and testing JavaScript code.

## ■ Using console.log()

Outputs information to the browser console for debugging.

Use it to display variable values, results, or messages during development.

## ■ How to Access the Console:

Right-click on the webpage → Inspect → Console tab (in browsers like Chrome or Firefox).

```
let message = "Hello, World!";  
console.log(message); // Outputs: Hello, World!
```

# BASIC CONCATENATION

## ■ String Concatenation:

- Combining two or more strings using the + operator.

## ■ Example:

demo...

```
let firstName = "John";  
let lastName = "Doe";  
console.log("Full Name: " + firstName + " " + lastName);
```

# EXPRESSIONS AND CONDITIONS

## ■ Expressions:

- Any valid unit of code that resolves to a value.
- Examples: `5 + 3`, `name === "Alice"`

## ■ Comparison Operators:

- `>`, `<`, `==`, `===`, `!=`, etc.

## ■ Conditions with if/else:

- Use conditions to execute code based on different situations.

```
let age = 20;

if (age >= 18) {
  console.log("You are an adult.");
} else {
  console.log("You are not an adult.");
}
```



# HANDLING MULTIPLE CONDITIONS WITH IF/ELSE

## ■ Multiple Conditions:

- You can use multiple if/else if statements to handle different cases.

## ■ Example:

- If the first condition is met, the first block runs.
- If not, the code checks the next else if.
- If no conditions are true, the else block runs by default.

Demo...

```
var day = "Monday";

if (day === "Monday") {
  console.log("Start of the week");
} else if (day === "Friday") {
  console.log("Almost weekend");
} else {
  console.log("Just another day");
}
```

# SIMPLE FUNCTIONS

- **What is a Function?**

A block of code designed to perform a specific task.

- **Defining and Calling a Function:**

A function is defined using the function keyword, followed by a name, parameters, and code block.

```
function greet(name) {  
  console.log("Hello, " + name);  
}  
  
greet("Alice"); // Outputs: Hello, Alice
```

# PASSING ARGUMENTS TO FUNCTIONS

- **What are Arguments?**

Inputs passed into a function when calling it.

- **Function Parameters:**

A function can take one or more parameters to operate on.

```
function add(a, b) {  
  return a + b;  
}  
  
var result = add(5, 10);  
console.log(result);
```

# PASSING ARGUMENTS TO FUNCTIONS

- **What are returns?**

A function can optionally return a value

Demo...

```
function add(a, b) {  
  return a + b;  
}  
  
var result = add(5, 10);  
console.log(result);
```

# CONCLUSION

- **The Console:**  
Allows us to see output from the JavaScript routine
  - **Variables:**  
Store data in memory
  - **Expressions and conditions:**  
Can be used to test and determine the program flow
  - **Functions**  
Can be used to break down functionality in to smaller reusable routines
- 1. JavaScript Variables (MDN Web Docs):  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar\\_and\\_types#declarations](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types#declarations)
  - 2. JavaScript Concatenation (w3schools):  
[https://www.w3schools.com/js/js\\_string\\_methods.asp](https://www.w3schools.com/js/js_string_methods.asp)
  - 3. JavaScript Expressions and Conditions:  
<https://javascript.info/ifelse>
  - 4. JavaScript Functions (MDN Web Docs):  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>
  -

QUESTIONS?