Orange Digital Center – Codding School

Cuttington University Campus, Suakoko
Bong County, Liberia

Introduction to Web Development

Week-3 Lesson

Introduction to JavaScript

JavaScript Syntax and Basic Operations

JavaScript is a versatile programming language that is primarily used for adding interactivity to web pages. It is an essential part of web development, along with HTML and CSS.

Basic Syntax

A JavaScript program is a series of statements that can include variables, functions, expressions, and operators.

```
// This is a single-line comment


/

This is a

multi-line comment

/


console.log("Hello, World!"); // Prints "Hello, World!" to the console
```

Basic Operations

JavaScript supports basic arithmetic operations:

```
let sum = 5 + 3;       // Addition
let difference = 5 - 3; // Subtraction
let product = 5 * 3;    // Multiplication
let quotient = 5 / 3;   // Division
let remainder = 5 % 3;  // Modulus
```

Variables, Data Types, and Operators

Variables

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

```
var x = 10; // Declares a variable that can be re-assigned and re-declared
let y = 20; // Declares a block-scoped variable that can be re-assigned but not re-declared
const z = 30; // Declares a block-scoped variable that cannot be re-assigned or re-declared
```

Data Types

JavaScript has several data types:

- Primitive Types: number, string, boolean, null, undefined, symbol
- Objects: Collections of properties

```
let number = 42;

let string = "Hello";

let boolean = true;

let empty = null;

let notDefined;

let symbol = Symbol('sym');


let object = {  name: "John" ,  age : 30  };
```

Operators

JavaScript supports various operators for performing operations on variables and values.

Arithmetic Operators

```
let a = 5;

let b = 2;


let sum = a + b;    // Addition

let diff = a - b;   // Subtraction

let prod = a  b;   // Multiplication

let quot = a / b;   // Division

let rem = a % b;    // Modulus
```

Comparison Operators

```
let equal = (a == b);       // Equal to

let notEqual = (a != b);    // Not equal to

let strictEqual = (a === b); // Strict equal to

let strictNotEqual = (a !== b);  // Strict not equal to

let greater = (a > b);      // Greater than

let less = (a < b);         // Less than

let greaterOrEqual = (a >= b); // Greater than or equal to

let lessOrEqual = (a <= b); // Less than or equal to
```

Logical Operators

```
let and = (a > 0 && b > 0); // Logical AND
let or = (a > 0 || b > 0);  // Logical OR
let not = !(a > 0);        // Logical NOT
```

Conditional Statements (if, else)

Conditional statements are used to perform different actions based on different conditions.

```
let num = 10;

if (num > 0) {
   console.log("Number is positive");
} else if (num < 0) {
   console.log("Number is negative");
} else {
   console.log("Number is zero");
}
```

Selecting and Modifying DOM Elements

Selecting Elements

JavaScript can select and manipulate HTML elements using the DOM (Document Object Model).

```
let element = document.getElementById("myId"); // Selects an element by ID

let elements = document.getElementsByClassName ("myClass"); // Selects elements by class name

let elementsByTag = document.getElementsByTagName("div"); // Selects elements by tag name

let queryElement = document.querySelector(".myClass"); // Selects the first element that matches a CSS selector

let queryElements = document.querySelectorAll( ".myClass" ); // Selects all elements that match a CSS selector
```

Modifying Elements

```
let element = document.getElementById("myId");

element.innerHTML = "New Content"; // Changes the inner HTML of the element

element.style.color = "blue";  // Changes the CSS style of the element

element.className = "newClass";  // Changes the class of the element
```

Event Handling

Event handling in JavaScript allows you to execute code when events are triggered by the user.

Adding Event Listeners

```
let button = document.getElementById("myButton");

button.addEventListener( "click",  function() {
   alert("Button was clicked!");
});
```

Common Events:

- **click**: Triggered when an element is clicked
- **mouseover**: Triggered when the mouse pointer is over an element
- **mouseout**: Triggered when the mouse pointer leaves an element
- **keydown**: Triggered when a key is pressed
- **keyup**: Triggered when a key is released

References:

1. W3Schools: JavaScript Tutorial: https://www.w3schools.com/js/

2. MDN Web Docs: Introduction to the DOM: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

Week3 Task:

Add an event listener to the submit button of your product landing page; whenever that button is click it should alert the text "Thank you for reaching out. We will get back to you soon".