**Introduction to JavaScript Fundamentals**

**Created by: Snehal Meshram SY Comp**

**Date: 14/10/24**

**Overview:**

JavaScript is one of the core technologies used in web development to create interactive web pages. Let’s explore the basics: variables, data types, operators, functions, loops, conditionals, and simple examples of form validation or DOM manipulation.

**Objective:**

The objective of this guide is to provide a comprehensive introduction to JavaScript fundamentals, equipping readers with essential knowledge and skills in the following areas:

* Understanding variables, data types, and operators in JavaScript.
* Mastering control structures, including conditional statements and loops.
* Developing and using functions to create reusable code.
* Learning how to manipulate the Document Object Model (DOM) for dynamic web content.
* Implementing form validation techniques to enhance user experience and data integrity.

Table of Content

1. JavaScript Fundamentals

* [Variables](#Variables)
* [Data Types](#dt)
* [Operators](#op)

2. Control Structures

* [Conditional Statements](#cs)
* [Loops](#loops)

3.[Functions](#function)

4.[Form Validation](#fv)

5.[DOM Manipulation](#dm)

6.[Conclusion](#cc)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **Variables**

Variables are used to store data values. JavaScript Variables can be declared in 4 ways:

|  |
| --- |
| * Automatically * Using var * Using let * Using const |

**Example:**

|  |
| --- |
| let productName = "Laptop";  const price = 999.99;  var stockAvailable = 50; |

Correct JavaScript variables:

Incorrect JavaScript variables:

|  |
| --- |
| var  123=30;  var \*aa=320; |

1. **Data Types**

JavaScript has 8 Datatypes

* **String**: Stores text data. Example: let greeting = "Welcome to JavaScript";
* **Number**: Represents numeric values. Example: let temperature = 36.6;
* **Boolean**: Holds true or false values. Example: let isLoggedIn = false;
* **Object**: Stores key-value pairs. Example: let car = { brand: "Tesla", model: "Model X" };
* **Array**: Holds a list of values. Example: let colors = ["red", "green", "blue"];
* **Null**: Represents no value. Example: let selectedProduct = null;
* **Undefined**: Variable declared but not initialized. Example: let discount;

1. **Operators**

Javascript operators are used to perform different types of mathematical and logical computations.

Assignment Operator (=): Assigns values to variables.

Example: let score = 100;

Addition Operator (+): Adds values together.

Example: let total = 20 + 30; // Result: 50

Multiplication Operator (\*): Multiplies values.

Example: let area = 5 \* 6; // Result: 30

Comparison Operator (>): Compares if one value is greater than another.

Example: let isOlder = 25 > 18; // Result: true

1. **Functions**

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 calculateTotal(price, quantity) {      return price \* quantity;    }    console.log(calculateTotal(50, 3));  // Output: 150 |

1. **Loops**

The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. **for loop**
2. **while loop**
3. **do-while loop**
4. **for-in loop**

  // For Loop

console.log("For Loop:");

for (let i = 0; i < 3; i++) {

  console.log("Iteration:", i);

}

// While Loop

console.log("\nWhile Loop:");

let j = 0;

while (j < 3) {

  console.log("Iteration:", j);

  j++;

}

// Do-While Loop

console.log("\nDo-While Loop:");

let k = 0;

do {

  console.log("Iteration:", k);

  k++;

} while (k < 3);

// For-Of Loop (used for arrays)

console.log("\nFor-Of Loop:");

let array = ["apple", "banana", "cherry"];

for (let fruit of array) {

  console.log("Fruit:", fruit);

}

1. **Conditionals**

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

In JavaScript we have the following conditional statements:

* Use if to specify a block of code to be executed, if a specified condition is true
* Use else to specify a block of code to be executed, if the same condition is false
* Use else if to specify a new condition to test, if the first condition is false
* Use switch to specify many alternative blocks of code to be executed

**Example:**

let age = 20;

let day = "Monday";

// If-Else-If

if (age < 18) {

  console.log("You are a minor.");

} else if (age >= 18 && age < 60) {

  console.log("You are an adult.");

} else {

  console.log("You are a senior.");

}

// Switch Statement

switch (day) {

  case "Monday":

    console.log("It's the start of the week!");

    break;

  case "Friday":

    console.log("Weekend is near!");

    break;

  case "Sunday":

    console.log("It's a rest day.");

    break;

  default:

    console.log("It's a regular day.");

}

1. **Form Validation Example**

* HTML form validation can be done by JavaScript.
* If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Form Validation Example</title>

  <script src="formValidation.js" defer></script> <!-- Link to external JavaScript file -->

</head>

<body>

  <form id="myForm">

    First Name: <input type="text" id="fname" name="fname">

    <input type="submit" value="Submit">

  </form>

  <div id="errorMessage" style="color: red;"></div>

</body>

</html>

**JavaScript (formValidation.js):**

document.getElementById("myForm").addEventListener("submit", function(event) {

  let firstName = document.getElementById("fname").value;

  // Check if the first name field is empty

  if (firstName === "") {

    event.preventDefault(); // Prevent form submission

    alert("First name must be filled out.");

    document.getElementById("errorMessage").textContent = "Please enter your first name.";

    return false;

  }

});

1. **DOM Manipulation Example**

Modify the content of HTML elements dynamically with JavaScript:

<p id="text">Hello, World!</p>

<button onclick="changeText()">Click me</button>

<script>

function changeText() {

  document.getElementById("text").innerHTML = "Text changed!";

}

</script>

**Conclusion**

This guide introduced the fundamental concepts of JavaScript, including variables, data types, operators, functions, loops, and conditionals. Additionally, it provided practical examples of form validation and DOM manipulation. Understanding these basics equips you to create interactive web applications and enhances your overall coding skills.