JavaScript Basics: A Beginner's Guide

Introduction

JavaScript is one of the most important programming languages for web development. It helps in making web pages interactive and dynamic. This guide will help beginners understand the basics of JavaScript.

---

1. Printing 'Hello, World!' in JavaScript

In any programming language, the first step is to print "Hello, World!" to understand basic syntax.

Code Example:

```
console.log("Hello, World!");
```

Explanation:

- `console.log()` is used to print messages in the browser console.
- `"Hello, World!"` is enclosed in double quotes (or single quotes) as a string.

---

2. Comments in JavaScript

Comments help in explaining code and making it more readable.

Types of Comments:

- **Single-line comment**: Use `//` before the text.
- **Multi-line comment**: Enclose the text within `/* */`.

Examples:

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

/*
This is a multi-line comment
which can span multiple lines.
*/
```

---

3. Variables in JavaScript

Variables store data values. In JavaScript, we use var, let, and const to declare variables.

Key Differences:

- var is function-scoped, can be redeclared and reassigned, but has issues with hoisting.
- let is block-scoped, can be reassigned but not redeclared.
- const is block-scoped and cannot be reassigned or redeclared.

Examples:

```
var name = "John"; // Can be redeclared and updated
let age = 25; // Can be updated but not redeclared
const PI = 3.14; // Cannot be updated or redeclared
```

---

4. Scope in JavaScript

Scope determines the accessibility of variables.

Types of Scope:

1. **Global Scope** - Accessible anywhere in the code.
2. **Function Scope** - Accessible only inside a function.
3. **Block Scope** - Variables declared with let and const are confined within {}.

Example:

```
function example() {
    var x = 10; // Function scope
    let y = 20; // Block scope
    const z = 30; // Block scope
    console.log(x, y, z); // Works fine
}
console.log(x); // Error: x is not defined
```

---

5. Variable Hoisting

Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope.

Example:

```
console.log(a); // Undefined
var a = 10;
```

Explanation:

- The variable a is hoisted to the top, but only its declaration, not its value.

6. JavaScript Data Types

JavaScript has 8 basic data types.

Primitive Data Types:

- **Number** – Example: let num = 10;
- **String** – Example: let str = "Hello";
- **Boolean** – Example: let isValid = true;
- **Undefined** – Example: let x;
- **Null** – Example: let y = null;
- **BigInt** – Example: let big = 9007199254740991n;

Non-Primitive Data Type:

- **Object** – Example: { name: "John", age: 25 }

---

7. Operators in JavaScript

Operators perform operations on variables and values.

Arithmetic Operators:

- + Addition → Example: 5 + 3 = 8
- - Subtraction → Example: 5 - 3 = 2
- * Multiplication → Example: 5 * 3 = 15
- / Division → Example: 5 / 2 = 2.5
- % Modulus (Remainder) → Example: 5 % 2 = 1
- ** Exponentiation → Example: 2 ** 3 = 8

Comparison Operators:

- == Equal to → Example: 5 == "5" (true)

- === Strictly Equal → Example: 5 === "5" (false)
- != Not Equal → Example: 5 != 3 (true)
- !== Strictly Not Equal → Example: 5 !== "5" (true)
- > Greater than → Example: 5 > 3 (true)
- < Less than → Example: 5 < 3 (false)

Logical Operators:

- && AND → Example: true && false (false)
- || OR → Example: true || false (true)
- ! NOT → Example: !true (false)

Assignment Operators:

- = Assigns a value → Example: a = 10
- += Adds and assigns → Example: a += 5 // a = a + 5
- -= Subtracts and assigns → Example: a -= 5 // a = a - 5
- *= Multiplies and assigns → Example: a *= 5 // a = a * 5

Know More

FAQs

1. **Why use let instead of var?**
    - let has block scope, making it safer to use than var, which can be overwritten anywhere in the function.
2. **Can const be changed later?**
    - No, const values cannot be reassigned after initialization.
3. **What is the difference between == and ===?**

- == checks value equality but ignores type, while === checks both value and type.

4. **What is JavaScript used for?**
    - JavaScript is used to create interactive web applications, add dynamic effects, and manage back-end logic.
5. **Is JavaScript case-sensitive?**
    - Yes, var name and var Name are treated as different variables.

This guide gives you a strong foundation in JavaScript. Keep practicing and exploring!
