

Thursday, 18 April 2024

# Basics of Javascript

---

## Variables

Variables in JavaScript are containers used to store data values. They provide a way to label and refer to data in a program. JavaScript variables can hold various types of data, including numbers, strings, booleans, objects, arrays, functions, and more.

Variables are declared using the `var`, `let`, or `const` keywords.

### **var Keyword:**

**Declaration:** Declares a variable globally or locally to a function, regardless of block scope.

**Hoisting:** Variables declared with `var` are hoisted to the top of their scope and can be accessed before they are declared (but their value will be undefined).

### **let Keyword:**

**Declaration:** Declares a block-scoped variable, limiting its scope to the block (`{}`) in which it is defined.

**Hoisting:** Variables declared with `let` are hoisted to the top of their block but are not initialized until their declaration is evaluated.

### **const Keyword:**

**Declaration:** Declares a block-scoped constant variable, whose value cannot be reassigned after initialization.

**Hoisting:** Variables declared with `const` are hoisted to the top of their block but, like `let`, are not initialized until their declaration is evaluated.

### **Naming Conventions:**

Variable names can consist of letters, digits, underscores (`_`), and dollar signs (`$`).

They cannot start with a digit.

Use meaningful and descriptive names to improve code readability.

# Operators

Operators in JavaScript are symbols that perform operations on operands (values or variables). They allow you to perform arithmetic, comparison, logical, assignment, and other operations.

## Arithmetic Operators:

Addition (+): Adds two operands.

Subtraction (-): Subtracts the second operand from the first.

Multiplication (\*): Multiplies two operands.

Division (/): Divides the first operand by the second.

Modulus (%): Returns the remainder of the division of the first operand by the second.

Increment (++): Increases the value of a variable by 1.

Decrement (--): Decreases the value of a variable by 1.

## Comparison Operators:

Equal to (==): Checks if two operands are equal.

Not equal to (!=): Checks if two operands are not equal.

Strict equal to (===): Checks if two operands are equal in value and type.

Strict not equal to (!==): Checks if two operands are not equal in value or type.

Greater than (>): Checks if the first operand is greater than the second.

Less than (<): Checks if the first operand is less than the second.

Greater than or equal to (>=): Checks if the first operand is greater than or equal to the second.

Less than or equal to (<=): Checks if the first operand is less than or equal to the second.

## Logical Operators:

Logical AND (&&): Returns true if both operands are true.

Logical OR (||): Returns true if at least one operand is true.

Logical NOT (!): Returns true if the operand is false.

## Assignment Operators:

Assignment (=): Assigns a value to a variable.

Addition assignment (+=): Adds the value of the right operand to the variable and assigns the result to the variable.

Subtraction assignment (-=): Subtracts the value of the right operand from the variable and assigns the result to the variable.

Multiplication assignment (\*=): Multiplies the variable by the value of the right operand and assigns the result to the variable.

Division assignment (/=): Divides the variable by the value of the right operand and assigns the result to the variable.

Modulus assignment (%=): Computes the modulus of the variable and the value of the right operand and assigns the result to the variable.

## Conditionals

Conditionals in JavaScript are used to execute different code based on specified conditions. They allow you to control the flow of your program by evaluating expressions and deciding which block of code to execute.

### if Statement:

- **Syntax:**

```
javascript

if (condition) {
    // Code to execute if condition is true
}
```

- Executes a block of code if the specified condition evaluates to true.

## if...else Statement:

- Syntax:

```
javascript

if (condition) {
    // Code to execute if condition is true
} else {
    // Code to execute if condition is false
}
```

- Executes one block of code if the condition is true, and a different block if it is false.

## else if Statement:

- Syntax:

```
javascript Copy code

if (condition1) {
    // Code to execute if condition1 is true
} else if (condition2) {
    // Code to execute if condition2 is true
} else {
    // Code to execute if all conditions are false
}
```

- Allows you to specify multiple conditions to test, executing the block of code associated with the first condition that evaluates to true.

## switch Statement:

- Syntax:

```
javascript Copy code

switch (expression) {
    case value1:
        // Code to execute if expression equals value1
        break;
    case value2:
        // Code to execute if expression equals value2
        break;
    default:
        // Code to execute if expression does not match any case
}
```

- Evaluates an expression and executes code based on a matching case value. The `break`