Day - 3: Input / Output in JavaScript

`alert()`

- Displays a dialog box with a message.
- Syntax: `alert("Message");`
- It returns `undefined`.

`confirm()`

- Displays a dialog box with **OK** and **Cancel** options.
- If the user clicks **OK**, it returns `true`.
- If the user clicks **Cancel**, it returns `false`.

`prompt()`

- Displays a dialog box asking the user for input.
- Syntax: `prompt("Enter your name:");`
- Returns the input as a **string**.

Tokens in JavaScript

- A token is the **smallest unit** in a program.

Example:

```
let name = "Avinash";
```

Types of Tokens:

Token Type	Example
Keyword	`let`
Identifier	`name`
Operator	`=`
Literal	`"Avinash"`
Punctuator	`;` , `""`

```
// Single-line comment
/* Multi-line
   comment */
```

Keywords in JavaScript: `var`, `let`, `const`

`var` (Behaves like "awara")

```
var gf = "Simran";
gf = "Anjali";
gf = "Anu";
var gf = "kajal";

console.log(gf); // Output: Kajal
```

- Multiple declarations allowed.
- Multiple initializations allowed.
- Function-scoped.

`let` (Behaves like "paglaa")

```
let gf = "Angel Priya";
gf = "Megha";
gf = "Muskan";
let gf = "sanvi"; //Error

console.log(gf); // Output: Muskan
```

- Declaration allowed only once per scope.
- Reassignment is allowed.
- Block-scoped.

`const` (Behaves like "deewana")

```
const gf = "Khushi";
gf = "Prachi"; //  Error
```

- One-time declaration.
- No reassignment allowed.
- Must be initialized during declaration.
- Block-scoped.

Identifier

- Name used to identify variables, functions, classes, etc.

Examples:

```
let name = 10;          // 'name' is identifier
class Car {}            // 'Car' is identifier
function fun() {}       // 'fun' is identifier
```

Naming Conventions:

- `isMarried` → lower camel case
- `IsMarried` → upper camel case

Rules for Identifiers:

- Cannot start with a number.
- Cannot contain special characters (except `_` and `$`).
- Cannot contain spaces.
- Cannot use reserved keywords (`if`, `let`, `const`, etc.).

Operators in JavaScript

1. Unary Operators:

- Increment / Decrement: `++`, `--`

2. Binary Operators:

Arithmetic:

`+`, `-`, `*`, `/`, `%`

Assignment:

`=`, `+=`, `-=`, `*=`, `/=`, `%=`

Comparison:

`==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`

Logical:

`&&`, `||`, `!`

3. Ternary Operator (Conditional):

```
condition ? value_if_true : value_if_false;
```

Comparison Operators Deep Explanation

`==` (Loose Equality)

- Compares **values only**, ignores type.

```
let a = 10;
let b = "10";

console.log(a == b); // true (Other languages will give 'false')
```

> JS converts the string to number automatically. It will check only value without checking their type.

`===` (Strict Equality)

- Compares **value and type**.

```
let a = 10;
let b = "10";

console.log(a === b); // false
- because it will compare value including type.
```

`!=` (Loose Not Equal)

- Compares values, ignoring type.

```
let a = 10;
let b = "10";

console.log(a != b); // false (Other language give true)
- Here implicit type conversion will be happen automatically and value of b
will treated as number.
```

`!==` (Strict Not Equal)

- Compares both value and type.

```
let a = 10;
let b = "10";

console.log(a !== b); // true
```

Script.js Code with Explanation

```
console.log("object"); // Prints 'object' to console

alert("Alert message"); // Shows alert box (returns undefined)

const x = alert("Alert Message -2"); // Shows alert, x is undefined
console.log(x);

const y = confirm("Do you want to learn JavaScript?");
console.log(y); // true/false based on user's response

const name = prompt("Enter Your Name: "); // Takes input from user
document.writeln("<h1>Hello </h1>" + name); // Displays greeting on page

let a = 10;
++a; // Pre-increment: 11
++a; // 12
++a; // 13
++a; // 14

let b = a++ * 2; // b = 14*2 = 28, then a becomes 15

console.log("a = " + a); // 15
console.log("b = " + b); // 28

// JavaScript coercion examples
console.log(10 + '10'); // "1010" - number + string = string
console.log(10 + 10 + "10"); // "2010" - 20 + "10"
console.log(10 - 50); // -40
console.log(10 * 5); // 50

console.log('10' + 3); // "103" - string + number = string
console.log('10' + 10 + '10'); // "101010"

console.log(10 - 'a'); // NaN - 'a' can't be converted to number
```