Day 2: Operators & Loops (JavaScript) [Colored & Attractive]

1. Operators

Operators are **symbols used to perform operations** on values or variables. In JavaScript, we have several types of operators:

1.1 Arithmetic Operators

Used for **mathematical calculations**.

Operator	Description	Example	Result
`+`	Addition	`5 + 3`	8
`-`	Subtraction	`5 - 3`	2
`*`	Multiplication	`5 * 3`	15
`/`	Division	`15 / 3`	5
`%`	Modulus (remainder)	`10 % 3`	1
`**`	Exponentiation (power)	`2 ** 3`	8

Example in code:

```
let a = 10;
let b = 3;

console.log(a + b);  // <span style="color:green;">13</span>
console.log(a - b);  // <span style="color:green;">7</span>
console.log(a * b);  // <span style="color:green;">30</span>
console.log(a / b);  // <span style="color:green;">3.333</span>
console.log(a % b);  // <span style="color:green;">1</span>
console.log(a ** b); // <span style="color:green;">1000</span>
```

1.2 Comparison Operators

Used to **compare two values**. Returns `true` or `false`.

Example:

```
let x = 10;
let y = "10";

console.log(x == y);   // <span style="color:green;">true</span>  (value same)
console.log(x === y);  // <span style="color:red;">false</span> (type different)
console.log(x != 5);   // <span style="color:green;">true</span>
console.log(x > 5);    // <span style="color:green;">true</span>
console.log(x <= 10);  // <span style="color:green;">true</span>
```

1.3 Logical Operators

Used to **combine conditions**.

Example:

```
let a = 10;
let b = 5;

console.log(a > 5 && b < 10);  // <span style="color:green;">true</span> (both
true)
console.log(a > 5 || b > 10);  // <span style="color:green;">true</span> (one
true)
console.log(!(a >
5));         // <span style="color:red;">false</span> (negates true)
```

1.4 Assignment Operators

Used to **assign or update values** of variables.

Example:

```
let x = 10;

x += 5;  // <span style="color:green;">x = 15</span>
x -= 3;  // <span style="color:green;">x = 12</span>
x *= 2;  // <span style="color:green;">x = 24</span>
x /= 4;  // <span style="color:green;">x = 6</span>
x %= 4;  // <span style="color:green;">x = 2</span>
```

```
console.log(x);
```

2.1 For Loop

```
for (let i = 1; i <= 10; i++) {
    console.log(i); // <span style="color:blue;">1 to 10</span>
}
```

Example: Loop numbers 1–10

```
for (let i = 1; i <= 10; i++) {
    console.log(i);
}
```

Explanation:

1. `let i = 1` → start from 1
2. `i <= 10` → loop until i is 10
3. `i++` → increase i by 1 each iteration

2.2 While Loop

```
let i = 1;
while (i <= 10) {
    console.log(i); // <span style="color:blue;">1 to 10</span>
    i++;
}
```

Example: Loop numbers 1–10

```
let i = 1;
while (i <= 10) {
    console.log(i);
    i++;
}
```

Explanation:

- The loop continues **as long as the condition is true**.
- Increment `i++` is necessary, otherwise it becomes an **infinite loop**

2.3 Do...While Loop

```
let i = 1;
do {
    console.log(i); // <span style="color:blue;">1 to 10</span>
    i++;
} while (i <= 10);
```

Example: Loop numbers 1–10

```
let i = 1;
do {
    console.log(i);
    i++;
} while (i <= 10);
```

Difference from while:

Executes at least once even if the condition is false initially.

Tasks/Practice

Task 1: Simple Arithmetic

```
let a = 12;
let b = 5;
console.log("Addition:", a + b);
console.log("Subtraction:", a - b);
console.log("Multiplication:", a * b);
console.log("Division:", a / b);
console.log("Remainder:", a % b);
console.log("Power:", a ** b);
```

Task 2: Compare numbers and strings

```
let x = 10;
let y = "10";
console.log(x == y);   // <span style="color:green;">true</span>
console.log(x === y);  // <span style="color:red;">false</span>
console.log(x != y);   // <span style="color:green;">false</span>
console.log(x !== y);  // <span style="color:green;">true</span>
```

Task 3: Loop 1–10 using for

```
for (let i = 1; i <= 10; i++) {
    console.log(i); // <span style="color:blue;">1 to 10</span>
}
```

Task 4: Loop 1–10 using while

```
let i = 1;
while (i <= 10) {
    console.log(i); // <span style="color:blue;">1 to 10</span>
    i++;
}
```

Task 5: Loop 1–10 using do...while

```
let i = 1;
do {
    console.log(i); // <span style="color:blue;">1 to 10</span>
    i++;
} while (i <= 10);
```

✅**Key Notes:**

- Use `===` instead of `==` to avoid type coercion problems.
- Always make sure **while loops have proper increment/decrement**, or they'll run infinitely.
- `for` loops are usually used when the **number of iterations is known**.
- `while` and `do...while` are used when **number of iterations depends on a condition**.