Day :9 JavaScript Notes.

Loop Control Statements

break

The break statement **exits the loop immediately**, stopping any further iterations.

Example:

```
for (let i = 1; i <= 10; i++) {
  if (i % 3 == 0) break;
  console.log(i);
}
```

// Output: 1, 2

The loop stops once i is divisible by 3.

---

continue

The continue statement **skips the current iteration** and moves to the next loop cycle.

Example:

```
for (let i = 1; i <= 10; i++) {
  if (i == 3) continue;
  console.log(i);
}
```

// Output: 1, 2, 4, 5, 6, 7, 8, 9, 10

When i is 3, the console.log(i) is skipped, so 3 is not printed.

Functions in JavaScript

What is a Function?

Functions are reusable blocks of code designed to perform a specific task. They help make your code modular, organized, and easier to manage and maintain.

Function Declaration

A function is declared using the function keyword, followed by a name and parentheses. The code to execute goes inside curly braces {}.

Example:

```
function greet() {
  console.log('Welcome, user!');
}

greet();  // Calls the function and prints: Welcome, user!
```

Variables Inside Functions

- **Local Variables:** Variables declared inside a function exist only within that function and cannot be accessed from outside.

```
function showMessage() {
  let message = "Hello";
  console.log(message);  // Prints "Hello"
}

showMessage();
```

```
console.log(message); // Error: message is not defined outside the function
```

- **Outer (Global) Variables:** Functions can access variables declared outside of them.

```
let user = "Alice";

function sayHello() {
  console.log(`Hello, ${user}!`);
}

sayHello();  // Prints: Hello, Alice!
```

Function Parameters

Functions can accept inputs called **parameters** to work with dynamic data.

```
function greet(name) {
  console.log(`Hello, ${name}!`);
}

greet('Ravi');  // Prints: Hello, Ravi!
```

Default Parameters

If a function is called without passing an argument, the parameter defaults to undefined. You can assign a **default value** in the function declaration to handle such cases.

```
function greet(name = "Guest") {
  console.log(`Hello, ${name}!`);
}

greet();        // Prints: Hello, Guest!
greet("Ravi");  // Prints: Hello, Ravi!
```

Returning Values

Functions can send values back to the caller using the return statement. Using return without a value immediately exits the function.

Example:

```
function add(a, b) {
  return a + b;
}

console.log(add(5, 3));  // Prints: 8
```

Naming Functions

Since functions represent actions, their names should usually be **verbs** or verb phrases.

- Choose names that are brief, clear, and descriptive.
- Examples of **poor names:** doStuff(), stuffHandler(), x()
- Examples of **good names:** calculateTotal(), getUserData(), sendEmail()