JavaScript Data Types, Conditional Statements, and Loops

Anthoniraj Amalanathan

1. Data Types

JavaScript has several built-in data types:

- **Primitive Data Types:**
  - i. **Number:** Represents numeric values. Examples: `5`, `3.14`.
  - ii. **String:** Represents textual data enclosed in single `'` or double `"` quotes. Examples: `'Hello'`, `"World"`.
  - iii. **Boolean:** Represents a logical value `true` or `false`.
  - iv. **Undefined:** Represents a variable that has been declared but not assigned a value.
  - v. **Null:** Represents the intentional absence of any object value.

- **Non-primitive Data Types:**
    1. **Object:** Represents a collection of key-value pairs (properties and methods).
    2. **Array:** Represents a collection of elements, which can be of any data type, accessed by their index.

2. Conditional Statements

JavaScript provides conditional statements to execute different code based on different conditions:

- **if Statement:**

```
let num = 10;
if (num > 0) {
    console.log("Positive");
} else if (num < 0) {
    console.log("Negative");
} else {
    console.log("Zero");
}
```

- **switch Statement:**

```
let day = 2;
switch (day) {
    case 1:
        console.log("Monday");
        break;
    case 2:
        console.log("Tuesday");
        break;
    default:
        console.log("Other day");
}
```

3. Loops

JavaScript provides different types of loops to iterate over arrays, objects, or execute code a certain number of times:

- **for Loop:**

  ```
  for (let i = 0; i < 5; i++) {
      console.log(i);
  }
  ```

- **while Loop:**

```javascript
let i = 0;
while (i < 5) {
    console.log(i);
    i++;
}
```

- **do-while Loop:**

```javascript
let i = 0;
do {
    console.log(i);
    i++;
} while (i < 5);
```

- **for...in Loop (for objects):**

```
const person = {
    name: 'John',
    age: 30,
    city: 'New York'
};
for (let key in person) {
    console.log(key + ': ' + person[key]);
}
```

- **for...of Loop (for arrays):**

```
const fruits = ['apple', 'banana', 'orange'];
for (let fruit of fruits) {
    console.log(fruit);
}
```