Learn JavaScript with me Day-3

1. for Loops

- A `for` loop repeats a block of code a certain number of times.
- Syntax:

```
for (initialization; condition; increment) {

// Code to be executed

}
```

- Example:

```
for (let i = 0; i < 5; i++) {

console.log(i);

}

// Output: 0 1 2 3 4
```

2. Dry Run

- Dry running means simulating a loop or algorithm step by step without actually executing the code.
- Helps in understanding the flow of the program and tracking the variables at each iteration.

3. Print Odd Numbers

- Use a loop to print odd numbers within a range.
- Example:

```
for (let i = 1; i <= 10; i += 2) {

console.log(i);

}

// Output: 1 3 5 7 9
```

4. Print Even Numbers

- Similar to odd numbers, but for even numbers.
- Example:

```
for (let i = 0; i <= 10; i += 2) {
```

```
console.log(i);

}

// Output: 0 2 4 6 8 10
```

5. Infinite Loops

- An infinite loop occurs when the termination condition is never met, causing the loop to run endlessly.
- Example:

```
for (let i = 1; i > 0; i++) {

console.log(i); // This will run forever

}
```

6. Print Multiplication Table

- Use a loop to print the multiplication table for any given number.
- Example (for number 5):

```
let num = 5;

for (let i = 1; i <= 10; i++) {

console.log(`${num} * ${i} = ${num * i}`);

}

// Output: 5 * 1 = 5, ..., 5 * 10 = 50
```

7. Nested for Loop

- A `for` loop inside another `for` loop.
- Commonly used for working with multi-dimensional arrays.
- Example:

```
for (let i = 1; i <= 3; i++) {

for (let j = 1; j <= 3; j++) {

console.log(i, j);

}

}
```

```
// Output:

// 1 1, 1 2, 1 3, 2 1, ..., 3 3
```

8. while Loops

- The `while` loop repeats a block of code as long as a specified condition is true.
- Syntax:

```
while (condition) {

// Code to be executed

}
```

- Example:

```
let i = 0;

while (i < 5) {

console.log(i);

i++;

}
```

9. break Keyword

- `break` is used to exit a loop prematurely.
- Example:

```
for (let i = 0; i < 10; i++) {

if (i === 5) {

break;

}

console.log(i);

}

// Output: 0 1 2 3 4
```

11. Loops with Arrays

- Use loops to iterate over arrays and perform actions on each element.
- Example:

```
let arr = [10, 20, 30, 40];

for (let i = 0; i < arr.length; i++) {

console.log(arr[i]);

}
```

12. Loops with Nested Arrays

- Use nested loops to iterate over multi-dimensional arrays.
- Example:

```
let matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

];

for (let i = 0; i < matrix.length; i++) {

for (let j = 0; j < matrix[i].length; j++) {

console.log(matrix[i][j]);

}

}
```

13. for-of Loops

- A `for-of` loop iterates over iterable objects like arrays.
- Syntax:

```
for (let element of iterable) {

// Code to be executed

}
```

- Example:
  ```
  let arr = ['a', 'b', 'c'];
  for (let char of arr) {
  console.log(char);
  }
  ```

14. Nested for-of Loop

- Similar to a nested `for` loop but using the `for-of` loop syntax.
- Example:

  ```
  let nestedArr = [[1, 2], [3, 4]];

  for (let subArr of nestedArr) {

  for (let num of subArr) {

  console.log(num);

  }

  }
  ```

15. Todo App (only JS)

- A basic todo application where tasks are added, displayed, and removed using JavaScript.
- Tasks can be stored in arrays, and loops can be used to display them.
- Example

  ```
  let todos = ['Task 1', 'Task 2', 'Task 3'];

  for (let task of todos) {

  console.log(task);

  }
  ```