

## JS Practical 3 (Part 2)

### JavaScript Looping Constructs Lab Manual

In JavaScript, loops allow you to execute a block of code multiple times based on certain conditions. Loops are very useful in situations where repetitive tasks need to be performed.

This lab will cover the following looping constructs:

1. **for loop**
2. **while loop**
3. **do-while loop**

We will break down each loop type with its definition, syntax, and provide examples of increasing complexity.

### 1. for Loop

#### Definition:

A for loop repeats a block of code for a set number of times. It is typically used when you know beforehand how many times you want to execute the loop.

#### Syntax:

```
for (initialization; condition; update) {  
    // Code to be executed  
}
```

- **Initialization:** Usually used to initialize a counter variable (e.g., let i = 0).
- **Condition:** The loop will continue as long as this condition evaluates to true (e.g., i < 5).
- **Update:** This part executes after each iteration (e.g., i++ to increment i).

#### Examples:

##### Example 1: Basic for Loop

Print numbers 1 through 5.

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
}
```

```
}
```

### Example 2: for Loop with Array

Print each element of an array.

```
let fruits = ["Apple", "Banana", "Cherry"];  
for (let i = 0; i < fruits.length; i++) {  
    console.log(fruits[i]);  
}
```

### Example 3: for Loop with a Step Value

Print numbers from 0 to 10, stepping by 2 each time.

```
for (let i = 0; i <= 10; i += 2) {  
    console.log(i);  
}
```

## 2. while Loop

### Definition:

A while loop repeats a block of code as long as the specified condition evaluates to true. It is commonly used when you don't know in advance how many times the loop should run.

### Syntax:

```
while (condition) {  
    // Code to be executed  
}
```

- **Condition:** This condition is checked before each iteration. The loop will continue as long as it evaluates to true.

### Examples:

#### Example 1: Basic while Loop

Print numbers from 1 to 5.

```
let i = 1;
while (i <= 5) {
  console.log(i);
  i++; // Increment to avoid infinite loop
}
```

### **Example 2: while Loop with Array**

Print each element of an array.

```
let fruits = ["Apple", "Banana", "Cherry"];
let i = 0;
while (i < fruits.length) {
  console.log(fruits[i]);
  i++;
}
```

### **Example 3: while Loop with a Condition**

Print numbers from 10 down to 1.

```
let i = 10;
while (i >= 1) {
  console.log(i);
  i--;
}
```

## **3. do-while Loop**

### **Definition:**

A do-while loop is similar to the while loop, but it guarantees that the loop's block of code is executed at least once, because the condition is checked **after** the code execution.

### **Syntax:**

```
do {
  // Code to be executed
} while (condition);
```

- The code inside the do block is executed **once** before the condition is checked.
- The loop continues running as long as the condition evaluates to true.

### **Examples:**

#### **Example 1: Basic do-while Loop**

Print numbers from 1 to 5.

```
let i = 1;
do {
  console.log(i);
  i++;
} while (i <= 5);
```

#### **Example 2: do-while Loop with Array**

Print each element of an array.

```
let fruits = ["Apple", "Banana", "Cherry"];
let i = 0;
do {
  console.log(fruits[i]);
  i++;
} while (i < fruits.length);
```

#### **Example 3: do-while Loop with a Condition**

Print numbers from 1 to 10.

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

### Differences Between for, while, and do-while Loops

Feature	for Loop	while Loop	do-while Loop
<b>Execution Condition</b>	Condition is checked before each iteration.	Condition is checked before each iteration.	Condition is checked after each iteration.
<b>When to Use</b>	When you know the number of iterations beforehand.	When you don't know the number of iterations in advance.	When you need to execute the loop at least once.
<b>Syntax</b>	for (initialization; condition; update) {}	while (condition) {}	do {} while (condition);
<b>Example Use Case</b>	Looping through an array with a known size.	Looping while a dynamic condition is met.	Asking for user input at least once before checking if the condition is met.