**LOOPS**

**Question 1: Explain the different types of loops in JavaScript (for, while, do-while). Provide a basic example of each.**

**Types of Loops in JavaScript**

JavaScript provides different types of loops to execute a block of code multiple times. The three most common types are:

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

Each loop type has its own use case, and they differ in how the condition is evaluated and when the loop runs.

**1. for Loop**

The for loop is used when you know in advance how many times you want to iterate over a block of code. It consists of three parts:

* **Initialization**: Setting up the loop counter (e.g., let i = 0).
* **Condition**: A boolean expression that determines if the loop should continue (e.g., i < 5).
* **Increment/Decrement**: A statement to update the loop counter after each iteration (e.g., i++).

**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. while Loop**

The while loop executes a block of code as long as a specified condition is true. It checks the condition **before** each iteration.

**Syntax:**

while (condition) {

// Code to be executed

}

**Example:**

let i = 0;

while (i < 5) {

console.log(i);

i++;

}

**Output:**

0

1

2

3

4

**3. do-while Loop**

The do-while loop is similar to the while loop, but it executes the block of code **at least once**, even if the condition is false initially. The condition is checked **after** each iteration.

**Syntax:**

do {

// Code to be executed

} while (condition);

**Example:**

let i = 0;

do {

console.log(i);

i++;

} while (i < 5);

**Output:**

0

1

2

3

4

**Key Differences Between the Loops:**

1. **for loop**: Typically used when the number of iterations is known ahead of time.
2. **while loop**: Ideal when you want to execute a block of code as long as a condition is true, but you may not know how many times it will run.
3. **do-while loop**: Useful when you want to ensure that the loop runs **at least once**, even if the condition is false initially.

**When to Use Which Loop:**

* Use a **for loop** when you know the number of iterations (e.g., iterating through an array with a known length).
* Use a **while loop** when you want to repeat code based on a condition but don’t know how many times it will be executed.
* Use a **do-while loop** when you need to execute a block of code at least once, regardless of the condition.

**Question 2: What is the difference between a while loop and a do-while loop?**

The primary difference between a **while loop** and a **do-while loop** in JavaScript lies in when the condition is evaluated and how many times the loop runs.

**1. while Loop**

* **Condition Check**: The condition is checked **before** the loop starts. If the condition is false from the beginning, the loop **will not execute** even once.
* **Execution**: The code inside the while loop is executed **only if** the condition evaluates to true at the start of each iteration.

**Syntax:**

while (condition) {

// Code to execute as long as the condition is true

}

**Example:**

let i = 5;

while (i < 5) {

console.log(i); // This will not run because the condition is false initially

i++;

}

**Output:** (no output, because the condition is false at the start)

**2. do-while Loop**

* **Condition Check**: The condition is checked **after** the loop executes. This means the code inside the do-while loop **will always run at least once**, even if the condition is false initially.
* **Execution**: The loop guarantees that the block of code inside the loop will be executed **at least once**, and subsequent executions depend on the evaluation of the condition.

**Syntax:**

do {

// Code to execute at least once

} while (condition);

**Example:**

let i = 5;

do {

console.log(i); // This will run once, even though the condition is false

i++;

} while (i < 5);

**Output:**

5 // The code inside the loop runs once even though the condition is false after the first iteration

**Key Differences:**

| **Feature** | **while Loop** | **do-while Loop** |
| --- | --- | --- |
| **Condition Check** | Checked **before** the loop starts | Checked **after** the loop runs |
| **Guaranteed Execution** | Will not execute if the condition is false from the start | Will **always execute at least once** |
| **When to Use** | When you may want to skip the loop entirely if the condition is false at the start | When you need the loop to run at least once, regardless of the condition |

**When to Use Which Loop:**

* **while loop**: Use it when you want to repeat a block of code only if a condition is true at the start.
* **do-while loop**: Use it when you need the code to run at least once before checking the condition.