



Week 1.4

🕒 Created December 27, 2023 9:48 PM

👤 Created by A Aditya Kulkarni

🏷️ Tags Empty

Loops:

- **Dumb way:**
 - A basic approach would be manually adding numbers from 0 to 100.
- **Better way - For loops:**
 - Using a loop to iterate through numbers and calculate the sum.
 - Example:

```
let ans = 0; for (let i = 1; i <= 100; i++) { ans += i; } console.log  
(ans); // Output: 5050
```

- This is more efficient and less error-prone than adding numbers individually.

Functions:

- **What is a function?**
 - A function in JavaScript is a set of statements that performs a task or calculates a value. It takes input, processes it, and returns an output with a clear relationship between the input and output.

- **Syntax:**

- Function declaration:

```
function functionName(arg1, arg2) { // Function body return result; }
```

- Example:

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

- **How to call a function:**

- Example:

```
let result = addNumbers(5, 10); console.log(result); // Output: 15
```

- **Why do we need functions?**
 - **Reusability:** Avoid repeating code.
 - **Modularity:** Divide code into manageable parts.
 - **Readability:** Enhance code readability.

Callback Functions:

- **Can you call one function inside another function?**
 - Yes.
- **Is DRY being violated here?**
 - Yes, if there's repetition in the code.
- **Solution:**
 - Use callback functions or anonymous functions to avoid repetition.

- Anonymous functions:
 - Functions without a name.
 - Example:

```
let add = function (a, b) { return a + b; }; let result = add(3, 7);  
console.log(result); // Output: 10
```

Visualization of Loops:

- Loupe: A great way to visualize the execution of JavaScript code, especially loops. Allows step-by-step visualization of code execution.