



@CODE.CLASH

# JavaScript Concepts *you* Should Know



NEXT →

1

@CODE.CLASH

**Hey Everyone** 🙌

**Javascript is everywhere.** Millions of webpages are built on JS.

Let's discuss some of the **basic concept of javascript** which are **important to learn** for any Javascript developer.

Do **Like, save and Share** This Post If You Found This Helpful.

**NEXT** ➡



# Scope

Scope determines the **accessibility** of variables, objects and functions.

- In Javascript, a variable has **3 types of scope** :
  - a. Block Scope
  - b. Function Scope
  - c. Global Scope

# Hoisting

Hoisting in Javascript is a **behavior** in which a **function or variable** can be used **before declaration**.

- In terms of variable and constant keyword **var** is **hoisted**, and **let** and **const** does **not** allow hoisting.

```
test = 6;  
// using test before declaring  
console.log(test);  
var test;
```



# Closures

Closure means that an **inner function** always has **access** to the **variable of its outer function**, even after the outer function **has returned**.

```
const first = () => {  
  let greet = 'Hello';  
  const second = () => console.log(greet);  
  return second;  
}  
  
const newFun = first();  
// Hello
```

# Callbacks

A **callback function** can be defined as a function **passed into another function** as a parameter.

```
// function
function greet(name, callback) {
  console.log('Hi' + ' ' + name);
  callback();
}

// callback function
function callMe() {
  console.log('I am callback function');
}

// passing function as an argument
greet('Peter', callMe);
```



# Promises

Promises is a **good way** to **handle asynchronous** operations.

- It's used to **find out** if the asynchronous operation is successfully **completed or not**.
- A promise may have **one of three states**
  - Pending
  - Fulfilled
  - Rejected



# Async & Await

Stop and wait until something is **resolved**.

- We use the **async** keyword with a function to **represent** that the function is an **asynchronous** function.
- The async function **returns a promise**.

```
const showPost = async() => {  
  const res = await fetch('https:xyz.com/post')  
}  
  
showPost();
```



- I have **already made posts on all this concept** in detail.
- If you have any **queries** then let me know in the **comment box**.