JS INTERVIEW
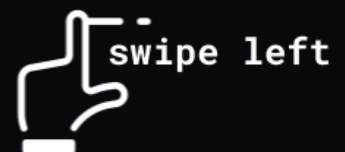# QUESTIONS PART 2

swipe left

Here's the Part 2 of my **JavaScript Interview Questions** series.

If you haven't seen **Part 1** yet, go to my Profile Guides and you will find the series.

If you enjoy this post, please **like** 👍🏾 and **share** 🙏🏾 with others. I will really appreciate it.

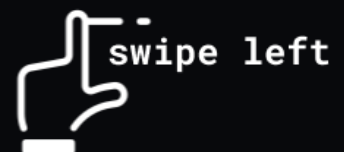Let's look at 5 questions you can be asked in JavaScript interviews with their answers.

swipe left

1. What does the "scope of a variable" mean in JavaScript?

When variables are created, they have an environment. That environment is the only area such variables can be accessed. Other parts of the code, **OUTSIDE THE ENVIRONMENT** cannot access that code.

This environment is called the **VARIABLE'S SCOPE**. The scope of a variable determines its accessibility by different parts of the code.
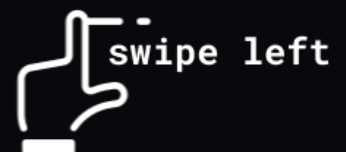
swipe left

2. Explain the different types of variable scope in JavaScript

In JavaScript, you have three scopes:

- **Global scope** is for variables defined outside of functions. This makes such variables accessible **EVERYWHERE**.
- **Local scope** is for variables defined inside functions, which makes the variable **ONLY** accessible **INSIDE** the function
- **Block scope**, introduced in ES6 is for variables assigned with **let** or **const** inside a **BLOCK**. A block is represented by curly braces **{...}** so such variables can be in **if** statements, **loop** statements, e.t.c, and are only accessible in that block.

swipe left

JS

3. Share 4 Rules for Naming Variables in JS

1. A variable name cannot begin with a number. It can only begin with a **letter** or an **underscore** _.

2. You cannot include special characters in variable names. Only the **dollar sign $** and **underscore _** are allowed.

3. Variable names cannot be a keyword. So you cannot name a variable **if**, **const**, **switch**, e.t.c

4. Variable names are **case-sensitive** so **Animal** and **aniMal** are two different variables in JavaScript.

swipe left

4. What is a Callback in JavaScript?

Functions receive different data types as arguments. It could be a **number** argument, a **string**, **array** or even a **function**.
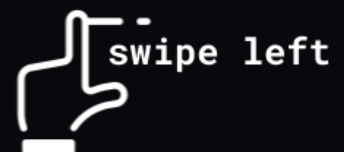
A function (say **FunctionA**) that is passed as an argument to another function (say **FunctionB**) is called a **callback** (or a **callback function**). This makes it possible for **FunctionB** to call **FunctionA** when it likes.

```
function functionA() {
  // some code
}
```

```
function functionB(callback) {
  // some code
  callback() // Execute callback
}
```

```
functionB(functionA)
```

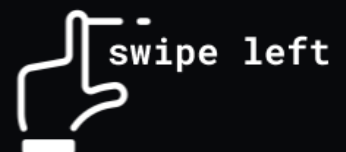functionA, passed as argument to functionB, then functionB can execute functionA()

swipe left

5. What is the difference between Local Storage and Session Storage in JS?

Local Storage and Session Storage are two ways of storing data on the browser's storage for different domains.

They both have the same syntax for adding, updating, retrieving and delete such data but the difference is that **local storage** does not have an **expiry** for the data it saves. Such data only expires when you manually delete it.

But, **session storage** has an expiry for the data it stores. Such data expires when the page session ends. A session ending means the page's tab is closed or the browser is closed.

If you enjoyed this part, please share with others so they can learn from and improve their JavaScript knowledge.

Also, you can find the video version of this series on my YouTube channel - **DEEECODE**

In the video version, I wrote code examples and did more explanations, so do check it out 😇