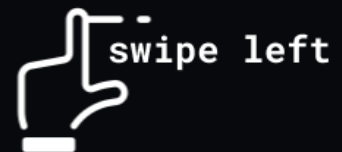


# JS INTERVIEW QUESTIONS **PART 3**



## JS INTERVIEW QUESTIONS - PART 3

Here's the Part 3 of my **JavaScript Interview Questions** series.

If you haven't seen **Part 1** or **2** yet, go to my YouTube channel **DEEEOCODE** and search for the **JavaScript Interview Questions playlist**.

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

Let's look at 5 questions (on **JavaScript events**) that you can be asked in JavaScript interviews with their answers.



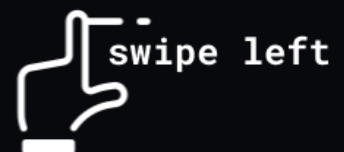
## 1. What is Event Bubbling in JS?

**Event bubbling** is a term in the DOM where the event an element receives is bubbled (transmitted) to its parent and ancestors elements (upward in the DOM tree).

```
<div>
  <span>
    <button>Click Me!</button>
  </span>
</div>
```

Look at this example. The **button** is a child of a **span** which is a child of a **div**.

When you **click** the button, the event is bubbled to the **span** element and the **div** element (the ancestors). So the div element can also handle such event (that originally occurred on the button) due to **Event Bubbling**.



## 2. What is Event Delegation

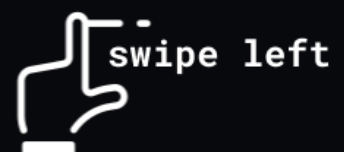
**Event Delegation** is a concept (based on **Event Bubbling**) that allows you to handle an event on a parent element instead of the original element that receives the event. Basically, you **delegate** the handling of that event to the parent element.

```
<div id="div">
  <button>Click Me!</button>
  <button>Also Click Me!</button>
</div>
```

```
const div = document.getElementById("div")

div.addEventListener("click", () => {
  console.log("button clicked")
})
```

In this example, you see we delegate the handling of the **click** event to the **div**. So when the first or second button is clicked, the div handles the event. This could be better than adding the event handler to the two buttons.



### 3. Share 3 ways of handling events in JS?

**1. Inline Event Handlers:** where the handling function is attached to the event attribute on the element

```
<button onclick="handler()"></button>
```

**2. addEventListener:** this method on element objects allows you to add functions for events.

```
const button = document.getElementById("button")

button.addEventListener("click", handler)
```

**3. Event Handler Property:** you can assign function handlers to event properties of the element object.

```
const button = document.getElementById("button")

button.onclick = handler
```



## JS INTERVIEW QUESTIONS - PART 3

### 4. What is the function of `event.preventDefault()`?

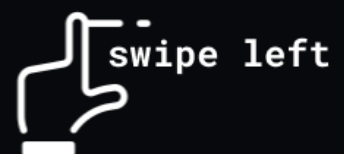
Some elements have default behaviours when an event is triggered on them. For example, when you **submit** a form, it sends the form data to the URL you provide on the **action** attribute of the form. When you **click** a link, the browser navigates to the value of the **href** attribute.

The **`event.preventDefault()`** method stops these default behaviours.

```
const link = document.getElementById("link")

link.addEventListener("click", (event) => {
  event.preventDefault()
})
```

By using the method here, clicking on such link will not do anything (no navigations).



## JS INTERVIEW QUESTIONS - PART 3

## 5. What is the function of `event.stopPropagation()`?

This method stops the propagation of events (which is the case of **event bubbling**).

In some cases, you do not want the parent element to receive an event that the child element receives. By using **stopPropagation** for an event on the child element, that event will not bubble to its parent or ancestors.

```
<div>
  <button id="button"></button>
</div>
```

```
const button = document.getElementById("button")

button.addEventListener("click", (event) => {
  event.stopPropagation()
})
```

Here, the div will not receive the **click** event anymore.



**JS INTERVIEW QUESTIONS - PART 1**

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 - **DEEEECODE - Simplify the Web**

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

