Understanding Inline Event Handling in JavaScript

A **responsive website** reacts to user actions like clicking buttons, moving the mouse, or typing in a field. JavaScript handles these actions using **events**. The code that responds to an event is called an **event handler**.

◆ Inline Event Handling

One of the simplest ways to handle events is by **embedding JavaScript directly into HTML elements**. This is called **inline event handling**.

Example: Displaying an Alert When Clicking a Link

```
<a href="#" onClick="alert('Hi');">Click</a>
```

Breaking it down:

- <a href="#"> → Normally, this would link to another page, but # reloads the current page.
- onClick="alert('Hi');" → Executes the JavaScript alert('Hi') when the user clicks the link.

Problem: Page Scrolls to the Top

Using <a href="#"> can cause the page to **jump to the top** when clicked.

✅ **Solution:** Use JavaScript:void(0), which prevents the page from reloading:

```
<a href="JavaScript:void(0)" onClick="alert('Hi');">Click</a>
```

◆ Multiple JavaScript Statements in Inline Event Handling

You can add more than one statement inside onClick, separating them with semicolons (;).

Example: Declaring a Variable and Showing an Alert

```
<a href="JavaScript:void(0)" onClick="var greet = 'Hi'; alert(greet);">Click</a>
```

- The variable greet stores 'Hi'.
- The alert(greet); statement displays the value of greet in an alert box.

Although this works, writing multiple JavaScript statements inside onClick is **not recommended** because it makes the HTML hard to read and maintain.

◆ Using a Function Instead of Inline JavaScript

A better approach is to use a **JavaScript function** and call it inside onClick.

Example: Calling a Function on Click

```
<script>
 function popup(message) {
   alert(message);
 }
</script>

<a href="JavaScript:void(0)" onClick="popup('Hello!');">Click</a>
```

Why is this better?

✅ The function is reusable.

✅ The HTML remains clean.

✅ Easier to maintain and debug.

📌 Summary

- **Inline event handling** allows JavaScript to be embedded directly into HTML elements.
- **onClick="alert('Hi');"** executes a JavaScript function when the user clicks the element.
- **JavaScript:void(0)** prevents the page from reloading.

- **Using a function instead of inline JavaScript** keeps the code clean and maintainable.