{ JavaScript }

Understanding Event Handling in JavaScript

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

{ JavaScript }

What is Event Handling ?

Event handling in JavaScript is the process of capturing and responding to user actions or events in a web application, like clicking a button, pressing a key, or submitting a form. It enables developers to make websites interactive and responsive by executing specific code when these events occur.

Key Concepts:

- Events
- Event Listeners
- Event Handlers
- Event Propagation
- Event Deligation

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

1. Events

Events are actions or occurrences that happen in the web browser. Examples include:

- **click:** When a user clicks on an element.
- **mouseover:** When a user hovers over an element.
- **keydown:** When a user presses a key on the keyboard.
- **load:** When a webpage or resource has finished loading.
- **submit:** When a user submits a form.

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

2. Event Listeners:

Event listeners are functions that wait for a specified event to occur on a particular element. Once the event occurs, the event listener triggers the function to execute.

```
document.getElementById('myButton').addEventListener('click', function() {
    alert('Button was clicked!');
});
```

3. Event Handlers:

vent handlers are the functions or code that execute in response to an event. They define what happens when the event occurs.

```
function handleClick() {
    alert('Button was clicked!');
}
document.getElementById('myButton').addEventListener('click', handleClick);
```

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

4. Event Propagation:

Event propagation describes how events flow through the DOM (Document Object Model). There are two main phases:

- **Bubbling:**

The event starts from the target element and bubbles up to the parent elements.

- **Capturing:**

The event starts from the root element and goes down to the target element.

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

{ JavaScript }

Example for Event Propagation

```
document.getElementById('parent').addEventListener('click', function() {
    alert('Parent element clicked!');
});


document.getElementById('child').addEventListener('click', function(event) {
    event.stopPropagation(); // Stops the event from bubbling up
    alert('Child element clicked!');
});
```

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

5. Event Delegation:

Event delegation is a technique where a single event listener is attached to a parent element to handle events from multiple child elements. This is efficient and useful when dealing with dynamically added elements.

Example for Event Deligation

```
document.getElementById('list').addEventListener('click', function(event) {
    if (event.target && event.target.nodeName === 'LI') {
        alert('List item clicked: ' + event.target.textContent);
    }
});
```

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

{ JavaScript }

Follow for more such content on JavaScript

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala