Event Bubbling and Event Capturing

Event bubbling and event capturing are two phases of event propagation in the DOM in web development. They define how events flow through the elements in the DOM tree.

Event Bubbling

Event bubbling is the phase where an event starts at the target element and then bubbles up to the root of the document.

Event Capturing

Event capturing (or event trickling) is the opposite of event bubbling. In this phase, the event starts from the root and trickles down to the target element.

Example:

```
<div id="parent">
    <button id="child">Click me</button>
</div>


const parent = document.getElementById('parent');
const child = document.getElementById('child');

parent.addEventListener('click', () => {
    console.log('Parent clicked!');
}, false); // false for bubbling

child.addEventListener('click', () => {
    console.log('Child clicked!');
}, false); // false for bubbling

parent.addEventListener('click', () => {
    console.log('Parent clicked!');
}, true); // true for capturing

child.addEventListener('click', () => {
    console.log('Child clicked!');
}, true); // true for capturing
```

Order of Events

1. **Capturing Phase:** Parent capturing -> Child capturing
2. **Target Phase:** Child target
3. **Bubbling Phase:** Child bubbling -> Parent bubbling

Stopping Event Bubbling

```
document.getElementById("child").addEventListener("click", (event) => {
  event.stopPropagation(); // Stops bubbling
  console.log("Child Clicked");
});
```

Event Delegation

Event delegation is a pattern in JavaScript where you attach a single event listener to a parent element instead of multiple child elements. This is useful when dealing with dynamically created elements and helps to avoid adding an event listener to each element.

Example:

```
<div id="parent">
  <button>Button 1</button>
  <button>Button 2</button>
</div>
```

```
document.getElementById("parent").addEventListener("click", (event) => {
  if (event.target.tagName === "BUTTON") {
    alert(`${event.target.innerText} clicked`);
  }
});
```

When you click on Button 2, the following happens:

1. The event is triggered on Button 2.
2. It bubbles up to the parent `<div>`, where the listener is attached.

3. `event.target` still refers to Button 2, even though the event has bubbled up to `<div>`.
4. The condition `if (event.target.tagName === 'BUTTON')` is true, so it logs "Button 2".