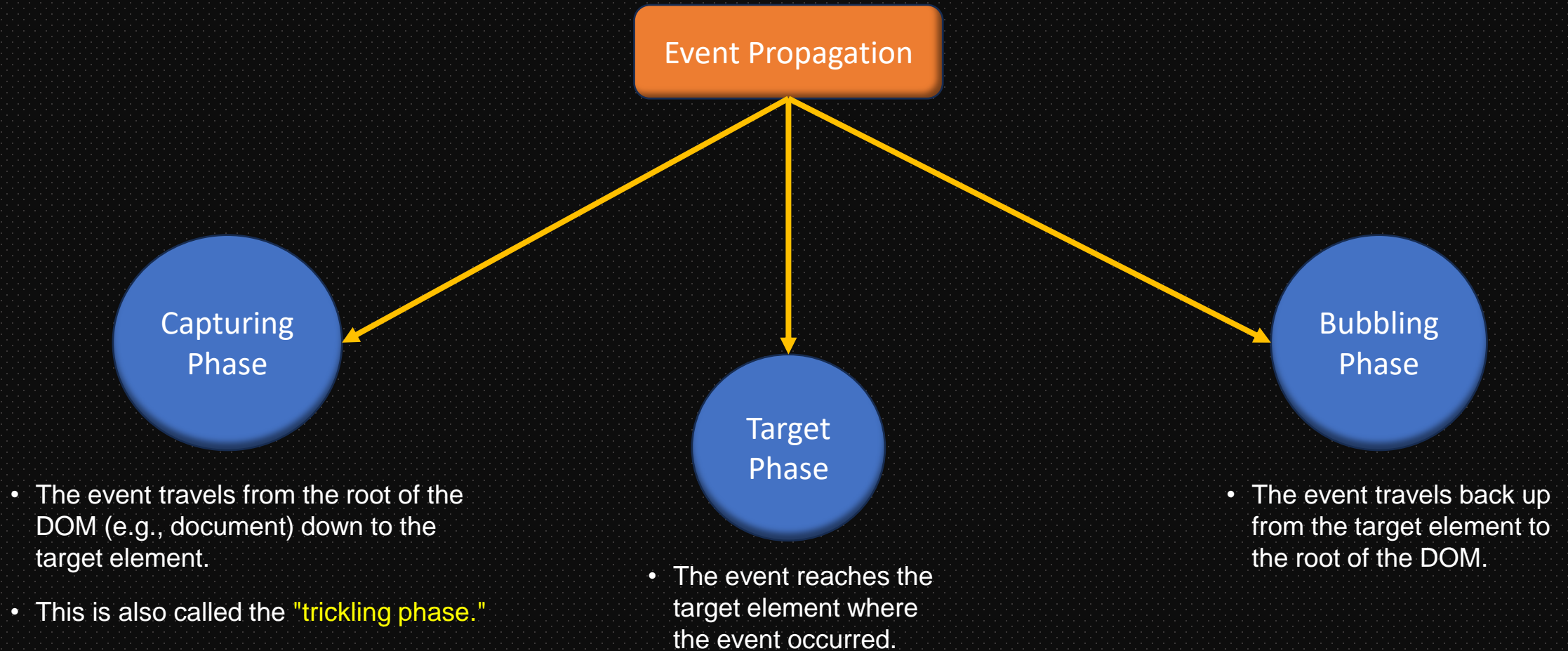
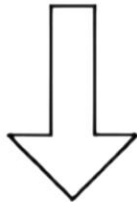


Event propagation describes the way events travel through the DOM after they are triggered. It consists of **three phases**:

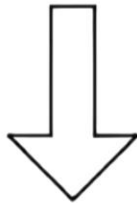


Capturing
Phase

Event 1



Event 2



Event 3

Top-Most
Element

Child Element

Descendant
Element

Event 6



Event 5



Event 4

Bubbling
Phase

event.target v/s event.currentTarget v/s this

"event.target" : Identifying which child element was clicked or interacted with.

"event.currentTarget" : Always points to the element where the listener is attached, regardless of where the event originated or which element triggered it.

"this": Refers to the context in which the function is executed. Inside an event listener, this points to the element to which the event listener was attached (the element where addEventListener was used).

How To Stop Bubbling & Capturing?

1. `stopPropagation()`
2. `stopImmediatePropagation()`

What is Event Delegation?

Event delegation is a technique in JavaScript where you add an event listener to a parent element instead of adding it to multiple child elements. The parent "delegates" the event handling to its child elements using the bubbling phase of event propagation.

This technique is useful when:

1. You have a large number of child elements that need the same event handler.
2. Child elements are dynamically added or removed.