*addEventListner:*

*The****addEventListener()****sets up a function that will be called whenever the specified event is delivered to the target.*

*The addEventListener() method is the recommended way to register an event listener. The benefits are as follows:*

* *It allows adding more than one handler for an event. This is particularly useful for libraries, JavaScript modules, or any other kind of code that needs to work well with other libraries or extensions.*
* *In contrast to using an onXYZ property, it gives you finer-grained control of the phase when the listener is activated (capturing vs. bubbling).*
* *It works on any event target, not just HTML or SVG elements.*

*Syntax:*

*element*.addEventListener(*event*, *function*, *useCapture*)

*event :  
 Donnot use on prefix .*

|  |  |
| --- | --- |
| *function* | Required. The function to run when the event occurs. |

stopPropagation VS stopImmediatePropagation

The **stopImmediatePropagation()** method of the [Event](https://developer.mozilla.org/en-US/docs/Web/API/Event) interface prevents other listeners of the same event from being called.

The main difference between event.stopPropagation() and event.stopImmediatePropagation() is how they handle event listeners in the DOM. event.stopPropagation() prevents the event from bubbling up to parent elements but allows other listeners of the same event on the same target to execute. In contrast, event.stopImmediatePropagation() not only stops the event from bubbling but also prevents any other listeners from executing on that specific event, even those attached to the same element.