Javascript DOM Events

To achieve this we have to add an event listener that fires when a user causes any event e.g. clicks a button

Events are fired to notify code of **"interesting changes"** that may affect code execution. These can arise from user interactions such as using a mouse or resizing a window, changes in the state of the underlying environment (e.g., low battery or media events from the operating system), and other causes.

https://codepen.io/FARZANA-TABASSUM-Lecturer-CSE/pen/PwNrmQY

Examples of HTML events:

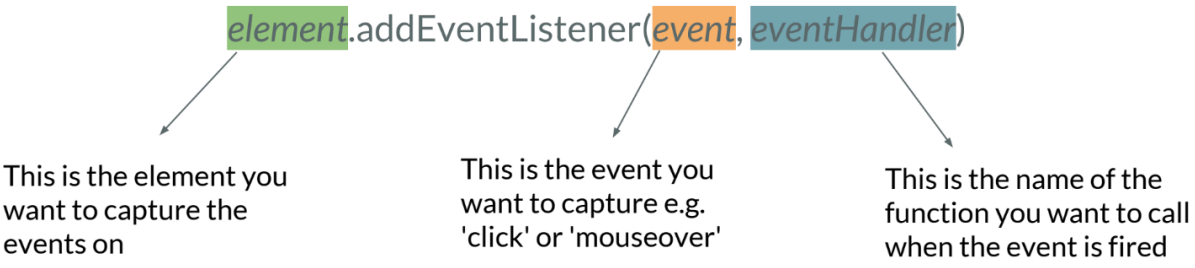
- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

Think of it like this:

> **Event** = Doorbell rings
> **Event listener** = You get up and open the door

To achieve this we have to add an event listener that fires when a user causes any event e.g. clicks a button

Add an Event Handler to an Element



You can add many event handlers to one element.

Passing Parameters

```
element.addEventListener(event, function(){ myFunction(p1, p2); });
```

Event Bubbling or Event Capturing

https://codepen.io/FARZANA-TABASSUM-Lecturer-CSE/pen/dPMBROb

There are two ways of event propagation in the HTML DOM, bubbling and capturing.

Event propagation is a way of defining the element order when an event occurs. If you have a <p> element inside a element, and the user clicks on the <p> element, which element's "click" event should be handled first?

In **bubbling** the inner most element's event is handled first and then the outer: the <p> element's click event is handled first, then the element's click event.

In **capturing** the outer most element's event is handled first and then the inner: the element's click event will be handled first, then the <p> element's click event.

With the addEventListener() method you can specify the propagation type by using the "useCapture" parameter:

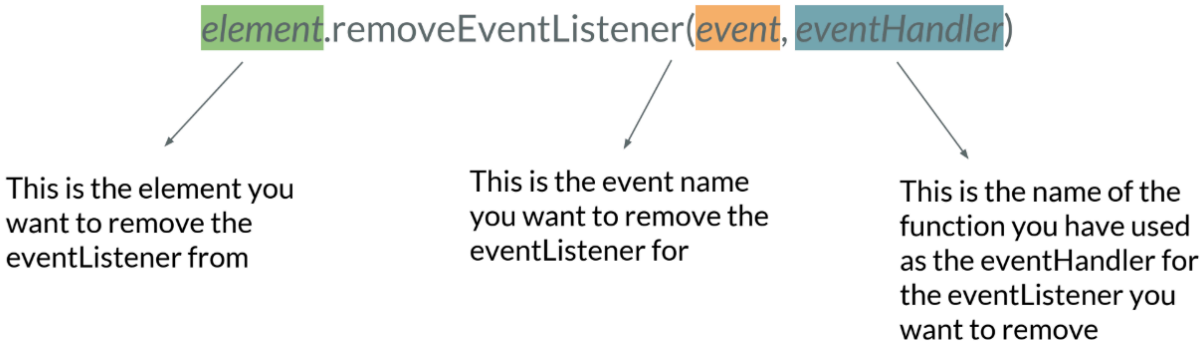
```
addEventListener(event, function, useCapture);
```

The default value is false, which will use the bubbling propagation, when the value is set to true, the event uses the capturing propagation.

Remove an Event Handler to an Element

https://codepen.io/FARZANA-TABASSUM-Lecturer-CSE/pen/OPNegWb

The `removeEventListener()` method removes event handlers that have been attached with the addEventListener() method:



When should YOU use removeEventListener()?

- Preventing duplicate actions (forms, payments)
- Temporary listeners (drag, resize, hover)
- Performance optimization
- State-based UI behavior