**Event Listener**

While we can add the onclick attribute directly on the <button> and add values like “alert(1)” or “<functionName>(1)”, it’s better if we do all these interactions in .js file.

To do this, we use the addEventListener() element method. The method takes in 2 parameters, ‘<event\_type>’ and a callback function with default parameter = e that is passed into that function.

The event type will determine when the function will be executed

Also, e has many different properties that we can use within this callback function.

e.clickX and e.clickY: the position of the mouse as we click the element (position from browser window)

e.offsetX and e.offsetX: position of the mouse as we click the element (relative position in the element)

e.altKey, e.ctrlKey, e.shiftKey: check if element is clicked with different keys held down.

e.type: returns the current event type

1. **Mouse Clicks**

+) <event\_type> = click 🡪 clicking on an element

*element*.addEventListener( ‘click’, function( ){ //*defining actions* } )

+) <event\_type> = dblclick 🡪 double click on an element

+) <event\_type> = mousedown 🡪 as soon as mouse is down, doesn’t need to be released

+) <event\_type> = mouseup 🡪 only after mouse released

+) <event\_type> = mouseenter 🡪 as mouse hovers on the element (exact)

+) <event\_type> = mouseover 🡪 as mouse hovers on the element (self + child elements)

+) <event\_type> = mouseleave 🡪 when mouse is not on the element (exact)

+) <event\_type> = mouseout 🡪 when mouse is not on the element (self + child elements)

+) <event\_type> = mousemove 🡪 whenever mouse is inside element and moves

(*we can play with e.offsetX or e.clickX that shows the tracking of mouse position)*

1. **Keyboard Enters**

*element*.addEventListener( ‘keydown’, function( ){ //*defining actions* } )

+) <event\_type> = keydown 🡪 when a key is pressed or holded

e.target.value: the value typed in (can output it out somewhere else)

+) <event\_type> = keyup 🡪 when a key is released

+) <event\_type> = keypress 🡪 when a key is pressed (disregard holding)

+) <event\_type> = focus 🡪 when element is focused (clicked inside)

+) <event\_type> = blur 🡪 when element is no longer focused (clicked out)

+) <event\_type> = cut 🡪 when we cut the text

+) <event\_type> = paste 🡪 when we paste the text

+) <event\_type> = input 🡪 when we do anything with the input

+) <event\_type> = ‘change’ 🡪 when an element changes

e.target.value: returns the current value

+) <event\_type> = ‘submit’ 🡪 when we submit