Notes

Introduction to events in JavaScript

Events are actions that happen on a web page when users perform actions. Events also occur when browsers also perform actions.

Example [1]:

- The user clicking the mouse over a certain element or hovering the cursor over a certain element.
- The user pressing a key on the keyboard.
- The user resizing or closing the browser window.
- A web page finishing loading.
- A form being submitted.
- A video being played, or paused, or finishing play.
- An error occurring.

When these actions are performed by users events are fired in the browser window. Events are also usually attached to a specific element or multiple elements within a web page.

There are 1000's of events in Javascript so we will only go over a few important ones.

If you want to take a look all the events available in JavaScript visit this link https://developer.mozilla.org/en-US/docs/Web/Events

Simple events

Button Clicks

You have already attached functions to button clicks within HTML tags but now we can use **Event Handlers** to do the same in a much more useful way.

Example: HTML CODE SNIPPET:

```
<button id = "changeColor">Click here to change my color</button>
```

JavaScript Code Snippet:

```
var changeButton = document.getElementById('changeColor');

function random(number){
   return Math.floor(Math.random()*number);
}

function switchColor(){
      var randomColor = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) +
')';
}
changeButton.addEventListener('click', switchColor);
```

In the above code we have attached a the click listener to the button element with the id changeColor using the addEventListener() function.

When the button is clicked the listener function switchColor is called.

The listener function then calls the random number to get a random color and sets the style.backgroundColor property of the button to the random color.

Button Double Click

As I mentioned before there are thousands of listeners in JavaScript. Another common listener for buttons is dblclick.

This listener listens for a user double clicking the button.

Example:

HTML CODE SNIPPET:

```
<button id = "changeColorDouble">Click here to change my color</button>
```

JavaScript Code Snippet:

```
ar doubleButton = document.getElementById('changeColorDouble');
function switchColorDouble(){
   var randomColor = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
// random is a function that I have defined above

   doubleButton.style.backgroundColor = randomColor
}
doubleButton.addEventListener('dblclick', switchColorDouble)
```

Removing Listeners

The best part of using the addEventListener function is that you can also remove listeners using the removeEventListener function.

Consider the below example where where we want the button to change colors only 1 time.

Example: HTML CODE SNIPPET:

```
<button id = "changeColorOnce">Double Click here to change my color</button>
```

JavaScript Code Snippet:

```
var buttonOnce = document.getElementById('changeColorOnce');

function switchColorOnce(){
   var randomColor = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
//
   buttonOnce.style.backgroundColor = randomColor
   buttonOnce.removeEventListener('click', switchColorOnce)
```

```
}

buttonOnce.addEventListener('click', switchColorOnce)
```

This is the same code as the first example. However, once the function switchColorOnce() is called we also call the buttonOnce.removeEventListener('click', switchColorOnce) function which removes the event listener from the button.

This means that the button only changes color once when it is clicked. After it is clicked once, if you try and click it again the button does not change color anymore.

Validating Forms and preventDefault:

You might have validated forms before but now you can do it easier with listeners.

Look at the code below:

Example: HTML CODE SNIPPET:

```
<form id  = "nameForm">
  <div>
    <label for="fname">First name: </label>
    <input id="fname" type="text">
  </div>
  <div>
    <label for="lname">Last name: </label>
    <input id="lname" type="text">
  </div>
  <div>
     <input id="submit" type="submit">
  </div>
</form>
<p id = "output"></p>
```

JavaScript:

```
var form = document.getElementById('nameForm');
var fname = document.getElementById('fname');
var lname = document.getElementById('lname');
var submit = document.getElementById('submit');
var para = document.getElementById('output');

function validate(e){
  if(fname.value == '' || lname.value == ''){
    e.preventDefault()
    para.textContent = "First name or last name is empty!"
```

```
  }
}

form.addEventListener('submit', validate);
```

Notice how we attach the listener to the form element and not the submit button!

This is how you should be doing it normally!

When forms are submitted in HTML the **default** or normal action is that the form gets submitted to the back-end.

We use the e.preventDefault() function to override the **default** functionality.

Instead we print some the error message "First name or last name is empty!" to a paragraph tag.

The e argument is called the **Event Object** we will learn more about this later when we discuss **Objects**.

For now just use it to call the e.preventDefault() function.

Example 1:

https://codepen.io/albseb511/pen/LYYXEvV

Example 2:

https://codepen.io/albseb511/pen/oNNQgOj

Mozilla Link

If you want more information on events and listeners visit this link

- https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events

Source Cited:

[1] Introduction to events,
Mozilla, https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events

Below are some of the many available events available

Mouse:

```
onclick, oncontextmenu, ondblclick, onmousedown, onmouseenter, onmouseleave, onmousemove, onmouseover, onmouseout, onmouseup
```

Keyboard:

```
onkeydown, onkeypress, onkeyup
```

Frame:

```
onabort, onbeforeunload, onerror, onhashchange, onload, onpageshow, onpagehide, onresize, onscroll, onunload
```

Form:

onblur, onchange, onfocus, onfocusin, onfocusout, oninput, oninvalid, onreset, onsearch, onselect, onsubmit

Drag:

ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop

Clipboard:

oncopy, oncut, onpaste

Media:

onabort, oncanplay, oncanplaythrough, ondurationchange, onended, onerror, onloadeddata, onloadedmetadata, onloadstart, onpause, onplay, onplaying, onprogress, onratechange, onseeked, onseeking, onstalled, onsuspend, ontimeupdate, onvolumechange, onwaiting

Animation:

animationend, animationiteration, animationstart

Miscellaneous:

transitionend, onmessage, onmousewheel, ononline, onoffline, onpopstate, onshow, onstorage, ontoggle, onwheel, ontouchcancel, ontouchend, ontouchmove, ontouchstart