JS Events

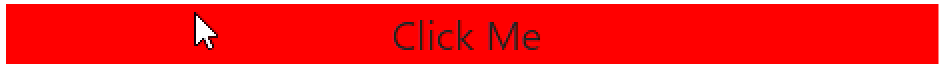
Lesson Time: 60 Minutes

HTML Events are activities that occur in the web browser and on the web page. For example;

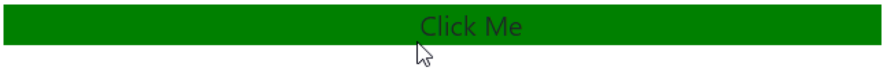
- moving your mouse over an HTML element triggers an event.
- Clicking on a button triggers an event.
- Clicking inside a form field to select it triggers an event.
- The loading of the HTML page itself triggers an event

Many things can occur in the normal course of web browsing that generate HTML events automatically in the background. JS can monitor these events and perform some kind of action when the event occurs.

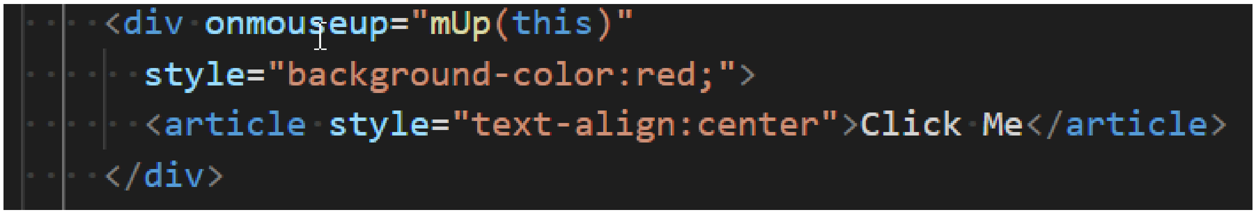
A classic example.
Here we have a red bar with the words click me.



When the HTML element is clicked, the background turns green.



First, the HTML element code. This code use the HTML event **onmouseup**

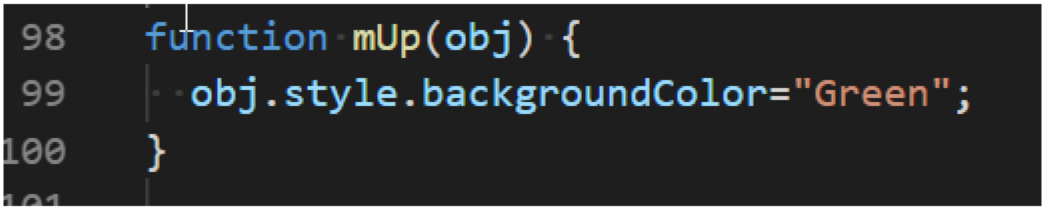


```
<div onmouseup="mUp(this)"
  style="background-color:red;">
  <article style="text-align:center">Click Me</article>
</div>
```

1. The HTML element, in this case a <div> has been given the **onmouseup** event. This event monitors when the user clicks the element is clicked.

2. When the onmouseup event is triggered, it runs the function mUp and passes a reference to itself.

The Javascript code:



```
function mUp(obj) {
  obj.style.backgroundColor="Green";
}
```

3. The javascript function is passed a reference to the HTML element, and executes the style change.

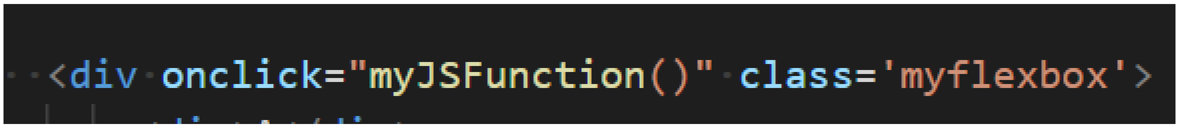
A complete list of events can be found in the MDN reference here: https://developer.mozilla.org/en-US/docs/Web/Events

Important HTML events to know include

HTML Event	Description
onclick	When the mouse is clicked
onmouseover	When the mouse moves over an HTML element
onmousedown	When mouse button is pressed down
onload	When the page loads
onunload	When the page closes
onfocus	When a form input element is selected
onchange	When a form element’s value is changed
onreset	When a form is reset
onsubmit	When a form is submitted

The last thing to cover in this lesson is talk about the JS DOM property **addEventListener**

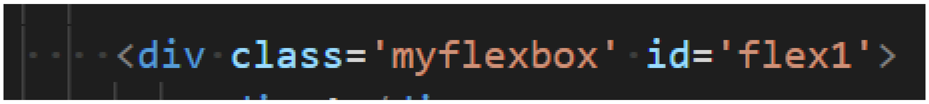
Here is an example HTML with some event added.



```
  <div onclick="myJSFunction()" class='myflexbox'>
```

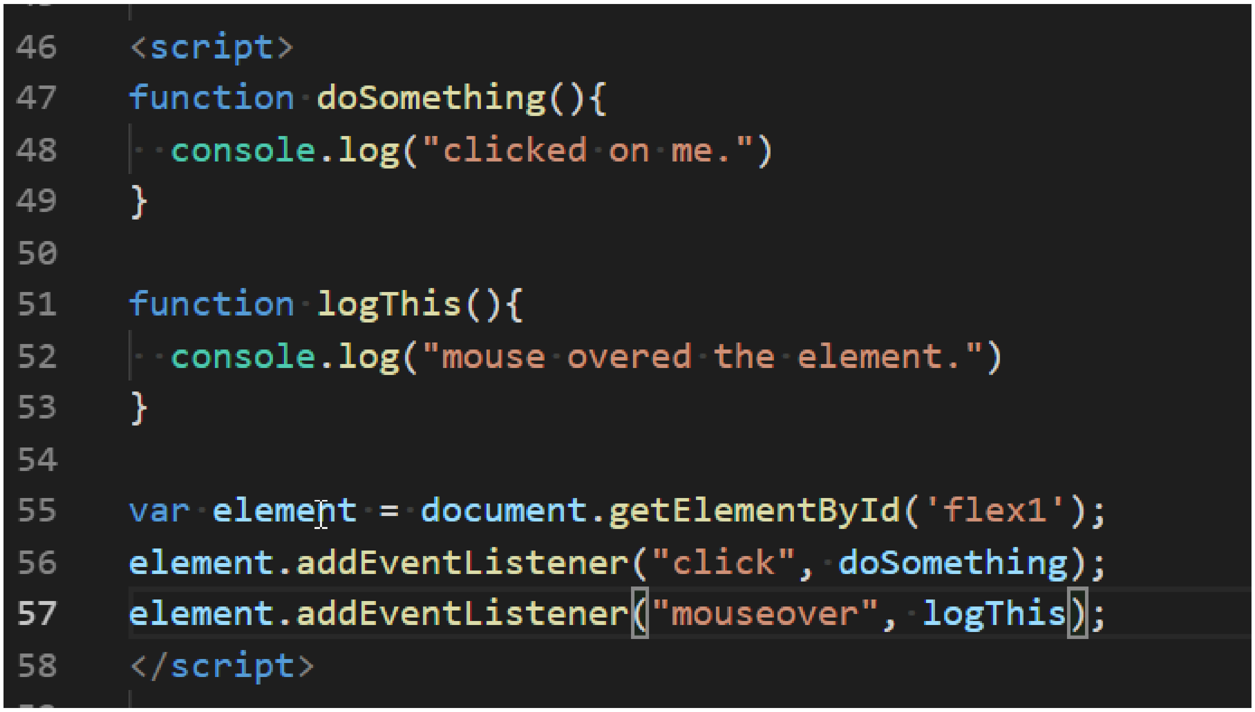
When this <div> element is clicked, a JS function will be executed. However, if we forget to add the onclick event to our HTML, the function would never run.

Here is our HTML with no **onclick** or **onmouseover**.



```
    <div class='myflexbox' id='flex1'>
```

JS will let us add our HTML event without us needing to add it to HTML first. So even if our <div> in this example doesn't include onclick, we can add it and get the same outcome using only JS. *In JS, we drop the "on" and just write "click" or "mouseover".*

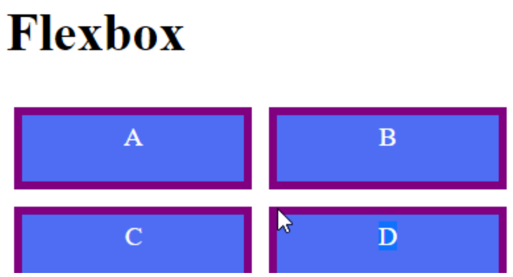


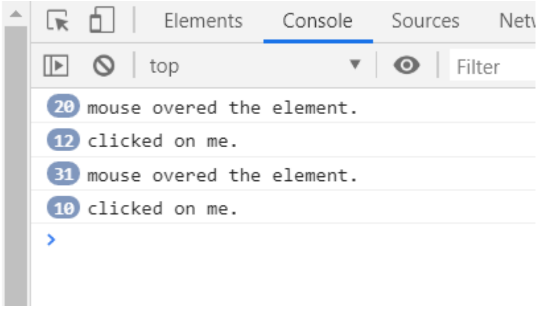
```
<script>
function doSomething(){
  console.log("clicked on me.")
}

function logThis(){
  console.log("mouse overed the element.")
}

var element = document.getElementById('flex1');
element.addEventListener("click", doSomething);
element.addEventListener("mouseover", logThis);
</script>
```

Results





```
20 mouse overed the element.
12 clicked on me.
31 mouse overed the element.
10 clicked on me.
>
```

It becomes your choice when coding if you want to write the events into the HTML document or add them with addEventListener without adding them first to HTML.

Key Terms	Events, addEventListener
Lesson Files	
Additional Resources	https://www.w3schools.com/js/js_events.asp https://developer.mozilla.org/en-US/docs/Web/Events https://www.w3schools.com/jsref/met_document_addeventlistener.asp
Further Learning	