



THE UNIVERSITY
of ADELAIDE

Web and Database Computing •

adelaide.edu.au



Introduction to JavaScript: Client-Side Events

Javascript Browser Events

What are events?

Web programming is commonly **Event Driven**

Rather than all your code running at once, when the page loads, some parts of your code can be triggered by external inputs These could be:

- General user input, e.g. mouse clicks, keyboard presses.
- A user interacting with specific elements on the page e.g. form input.
- Page elements loading or failing to load
- Network requests finishing
- Arbitrary events that you define!

This means we'll often need to pre-define code that will run at a later time...

Basic events

In the browser there are many types of events:

- **load** (onload attribute) For **body**, **img** and other select tags, fires when that element and all children finish loading.
- **change** (onchange) Fires when the value of an input is changed.
- **click** (onclick attribute) Fires when an element is clicked.
- **mousedown** and **mouseup** Fire when the mouse button is pressed down and then when released respectively.
- **mouseover/mouseenter** and **mouseout/mouseleave** Fire when the mouse cursor enters or leaves an element respectively.
- **keydown** and **keyup** Fire when the a key is pressed/released.
- And many more!
 - https://www.w3schools.com/jsref/dom_obj_event.asp

Using events

Events can be used to trigger our code by attaching our code to an event attribute:

- In the HTML

```
<button id="aButton" onmouseout="myfunction()">Triggers when the mouse leaves</button>
```

- Or in Javascript

```
document.getElementById("aButton").onmouseout = myfunction;
```

Multiple event handlers using listeners

The previous examples only allowed 1 function to run when an event occurs, but what if we want to dynamically add or remove several?

Use an EventListener:

```
var myButton = document.getElementById("aButton");  
myButton.addEventListener('mouseout', myFunction);
```

Where **myFunction** is the function we want triggered

We can add multiple listeners to the same element and event, and the same handler function to multiple listeners

```
var myButton = document.getElementById("aButton");  
myButton.addEventListener('mouseenter', myFunction1);  
myButton.addEventListener('mouseenter', myFunction2);  
myButton.addEventListener('mouseout', myFunction1);  
myButton.addEventListener('mouseout', myFunction2);
```

Multiple event handlers using listeners

We can dynamically add and remove event listeners:

```
var myButton = document.getElementById("aButton");  
myButton.addEventListener('mouseenter', myFunction1);  
myButton.removeEventListener('mouseenter', myFunction1);  
myButton.addEventListener('mouseenter', myFunction1);  
myButton.removeEventListener('mouseenter', myFunction1)
```

See https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp



THE UNIVERSITY
of ADELAIDE

CRICOS PROVIDER NUMBER 00123M