JavaScript Events - Lab 17

IMPORTANT! Save all your work to a safe location such as OneDrive.
Create a folder for GUI & Web Development into which you will save all your work for this module, arranged how you wish. Ideally you should create a folder each week for your lab exercises. Note that you should create **a separate file** for each exercise.

Exercise 1: HTML Event Handlers

In the last few weeks, we have used an “onclick” event listener to invoke (or run) a function when a button is clicked. An example of this is as shown:

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5 Events</title>
</head>
<body>

    <button onclick="myFunction()">Click me!</button>

</body>
</html>

<script>

    function myFunction(){
        alert("Function has run!");
    }

</script>
```

1: Recreate the page shown above and test. Note that your button works when you single-click it.

2: Amend your page so that the button is no longer listening for an “onlclick” event, but instead for a double-click event, or “on**dbl**click”. This can be achieved by amending the following line:

```
8       <button ondblclick="myFunction()">Click me!</button>
```

Test your page again with a single-click. It should not work. A double-click should trigger the function.

Change your HTML event to try out the following events for the button:

```
8       <button onmouseover="myFunction()">Click me!</button>
```

When you bring the mouse over the button, the function is triggered.

Then:

```
8       <button onmouseout="myFunction()">Click me!</button>
```

When you bring the mouse over the button and then move the mouse away (out), the function is triggered.

Exercise 2: More Event Handlers

1: Recreate the page shown below:

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>HTML5 Events</title>
5   </head>
6   <body>
7
8       <h1 id="myHeading"></h1>
9
10      <input id="inBox">
11
12  </body>
13  </html>
14
15  <script>
16
17
18  </script>
```

In this exercise you will create an HTML event handler on the input box in line 10 above. The event handler will “listen” for input (or typing) inside the box from line 10. Each time the user types a character inside the input box, the event *fires*. This will use the “oninput” event.

```
<input id="inBox" oninput="myFunction()">
```

Add the event listener to the input box.

Create the function specified below:

```
function myFunction(){
    alert("Event has fired!!");
}
```

Test your page. Each time you type into the input box (or if you delete a character inside the box), an alert should appear.

Amend the function so that the H1 “myHeading” is updated (remove the alert) after each input to contain whatever value is in the input box:

```
function myFunction(){
    let textInBox;
    textInBox = document.getElementById("inBox").value;
    document.getElementById("myHeading").innerHTML = textInBox;
}
```

Test your page. The H1 heading should update automatically as you type into the input box.

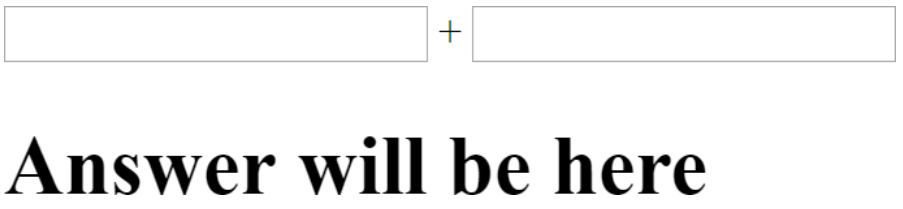
What this code is doing

You do not need to type the following comments in, the screenshot below is merely to explain what the code is doing:

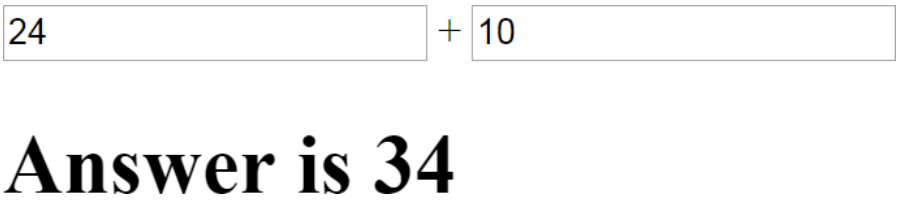
```
function myFunction(){
    //Creates a variable called textInBox
    let textInBox;

    //Statement to make the varaible "textInBox" equals to whatever
    has been typed into the input box, which has an id of "inBox"
    textInBox = document.getElementById("inBox").value;

    //Make the H1 heading equals to the information stored in the
    "textInBox" variable
    document.getElementById("myHeading").innerHTML = textInBox;
}
```

Exercise 3: "Live" Calculator

1. Recreate the page shown below:



This page should allow the user to type numbers into each box, and the addition of the 2 numbers will appear in the H1 underneath (without clicking buttons), as shown:



Exercise 4: DOM level 2 event listeners

Ensure you use DOM level 2 event handlers for all of the following exercises

1: Create a HTML web page with a HTML button. Create an event using a Dom level 2 event handler that will pop up an alert, when the button is clicked.

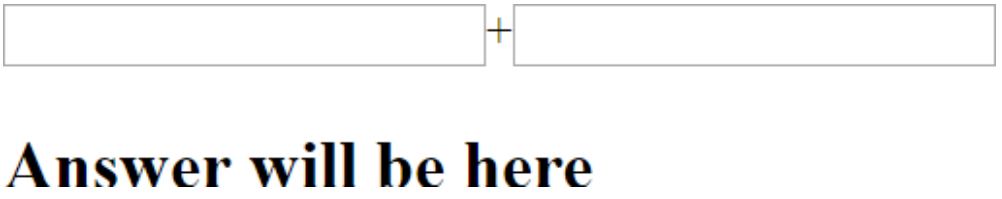
```
var btn = document.getElementById("myBtn");
btn.addEventListener("click", myFunction);
```

2: Create a HTML web page with a H2 heading. Create an event using a Dom level 2 event handler that will pop up an alert when the user clicks on the H2 heading, as shown below.

3: Add a h3 heading to your page that will cause a pop up when the user places the mouse over the h3 heading (Use the “mouseover” event).

4: Add a h4 heading to your page that will cause a pop up when the user places the mouse over the h3 heading and then moves the mouse away (Use the “mouseout” event).

5: Create a HTML web page that automatically adds 2 numbers input into 2 separate input boxes and displays the answer underneath. This is the same result as the exercise 3, but using a DOM level 2 event handler.



Exercise 5

1: Create a HTML web page that has an input box that allows a user to input a username. The username must be 6 characters or more. If it is, an appropriate message is displayed, as shown below. Otherwise, a message is displayed warning the user that more characters are required. Try this with both HTML event handlers and DOM level 2 event handlers.

Exercise 6

1: Create a webpage with one input element. Add an event listener to this element for a blur event, so that when the input element loses focus, a function will update the data typed into this field by replacing any occurrence of the ampersand symbol (&) with the word “and”.

Hint: Use the String method replaceAll – see String.prototype.replaceAll() - JavaScript | MDN (mozilla.org) for further info.

Exercise 7

1: Create a HTML web page that will change the colour of the background to reg, green or blue – if the R, G or B keys are pressed on the keyboard. If any other key is pressed, the background should turn back to white.

This will use the “keydown” event

See here for more info:
https://www.w3schools.com/jsref/event_onkeydown.asp

Exercise 8

In this exercise, you will create a live clock on a webpage. Create a new

webpage with a tag with id=“clock".

Write a JavaScript function which will find out the current time (format HH:MM:SS) and update the content of the tag every 1 second

Hint: use window.setInterval() to perform periodic calls to the function.

Hint: the following pages at w3schools will be useful

https://www.w3schools.com/js/js_date_methods.asp

https://www.w3schools.com/js/js_timing.asp