1. Under the comment, add event listeners to the elements, using the functions shown:

|  |  |  |
| --- | --- | --- |
| Element | Event | Handler |
| **button with id of textcontent** | click | blueToRed |
| **Button with id of bgcolour** | click | greenToPink |
| **Button with id of fonts** | click | tnrToArial |

1. Save the file and make sure that the page functions as it did at the end of the last QuickLab.
2. Add an event listener to tnrParagraph that has an event of mouseOver and a handler of mouseOver.
3. Add mouseOver as an arrow function that takes an argument of event:

const mouseOver = event => {

// function body here

}

1. Populate the function body:
   1. Change the background colour of the event target to limegreen;

event.target.backgroundColor = 'limegreen';

* 1. Check to see if the textContent of the event target contains the word background
     1. If it doesn't then add I have had my background colour changed on mouse over to it;
     2. If it does then replace the word out with over in it.

1. Save the page and check to see if the new mouse over functionality works.

You should find that it doesn't. The explanation for this is in JavaScript's hoisting rules. Normal variables and functions are hoisted by JavaScript, meaning that it knows about them and their implementation no matter where they are declared. Arrow functions (and Classes - when using Object Oriented JavaScript) are not. Therefore, the page fails silently.

1. Move the registering of the event listener to beneath the declaration of the arrow function.
2. Save the file and you should see that the event now fires correctly.
3. Add another event handling function to tnrParagraph, using the previous steps as a template, to change the background colour to yellow when the mouse leaves the element and add/change the text accordingly - don’t forget to register the event on tnrParagraph!

The way that these new event handlers have been written means that they are reusable across any elements as they are not tightly bound to the element. When we pass in the event, event.target gives the reference to the element which fired the event.

1. Add an arrow function called elementClick. It should:
   1. Receive event as an argument;
   2. Set the background colour of the element that raised the event to white;
   3. Change the text content of the element that raised the event to I have no event listeners attached to me now;
   4. Remove the click event listener on the button with id of fonts;
   5. Remove the click, mouseover and mouseout event listeners from the element that raised the event;
   6. Checks the id of the element that raised the event and if it was tnrParagraph:
      1. Changes the textContent of blueParagraph to Event listeners enabled;
      2. Adds event listeners to blueParagraph for click, mouseover and mouseout.
   7. If the element that raised the event didn't have an id of tnrParagraph it should:
      1. Change the textContent of tnrParagraph to Event listeners enabled;
      2. Adds event listeners to tnrParagraph for click, mouseover and mouseout.
2. Register the click event and handling function with tnrParagraph.
3. Save the file and check that all of the events are fired at the appropriate time. Use the developer tools and add console.logs if you wish to examine further.