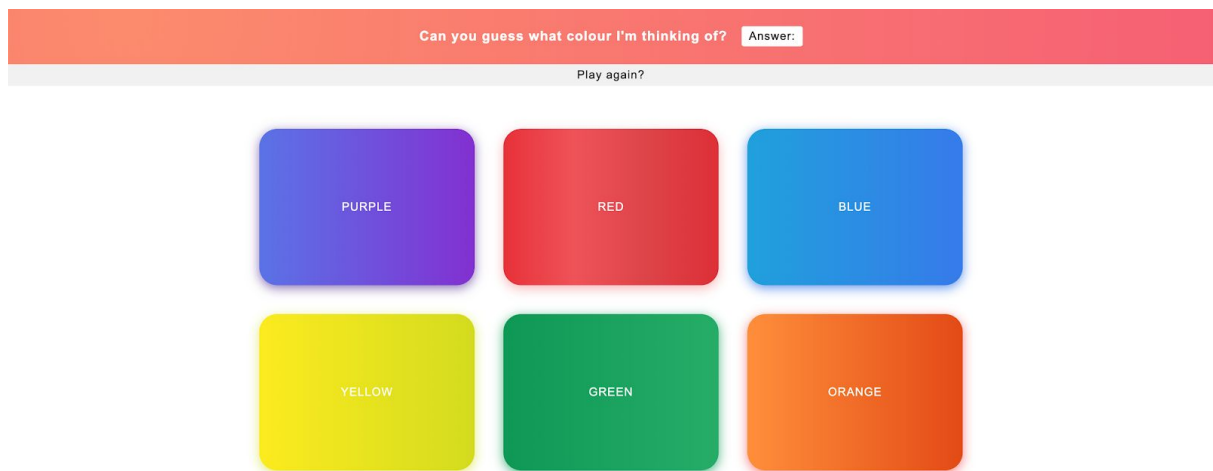


Web Workshop 1 - 13th November 2019

INTRO

In today's workshop, you will work in pairs and research methods to complete several missions 🚀 that will help you build a colour guessing game as shown below.



MISSIONS

1. Download the project folder from [Github](#) and open it in a text editor.
2. Open [index.html](#) ...
 - a. Add an element to show if the guess is correct or incorrect.
`// It should have an id of 'answer'`
 - b. Add 3 more buttons and give each an id of your choice
`// Try and follow the pattern used in the first 3 buttons`
 - c. Add a link to the [index.js](#) file in the project folder
`// This should be at the end of the contents of the body`
 - d. Create a link (a tag) for the 'Play again?' button with an id of 'play-again-btn'
`// It's href value should be '#'`

3. Open `index.css` ...

- a. Change the background colour of the 3 buttons you added earlier.

`// You should use the unique ids you gave each button to assign each button a colour. #id-name will allow you to select that element.`

- b. Change the border radius of the buttons to 25px

`// To select an element in CSS by an id you use #id-name, to select an element in CSS by a class you use .class-name`

- c. Hovering over links of the buttons should make the text colour change to black

`// Look for a CSS selector that is shared amongst all of the button elements. e.g. classes are given to style multiple elements and ids are given to style unique elements.`

- d. Increase the space between the heading and the element with an id of 'answer' that you created earlier

`// Investigate the differences between margin and padding.`

4. Open `index.js` ...

- a. If the game is over, display an alert saying 'The game is over!'

`// You will find an if statement on Line 10, think about the logic of this and where the alert may need to go.`

- b. Add an if statement to check if they have guessed the correct colour if the game is not over, update the text content of the answer container and the game over variable appropriately.

`// Look at the name attribute you gave each button, to reference the selected element inside of the function, you can use the target property of the variable passed in the parameters of the function - e.g. event.target.`

`// Now research how you could access the name attribute of the element object you now have access to.`

- c. For the play again button you should see an event listener and you should see a function that is called when the play again button is clicked. Add some code here to make the game restart.

`// Look for any code that already exists that may do this for you!`