Selector Methods in JavaScript

document.getElementById (*'insert the ID you want to target'*);

The **getElementById ()** method returns the element that has the **ID** specified inside the parentheses.

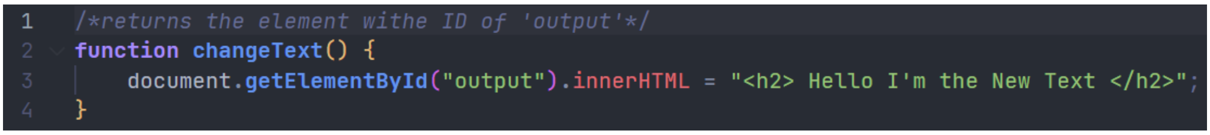
This method is one of the most common methods in the **HTML DOM** and is used frequently when you want to manipulate or get values from an element in your document.

Download the ***selectors.zip*** file from Moodle – open in Visual Code

Open *index.html* and *script.js – create the function below in script.js*

1 Change the output text:

getElementById() retrieves the element with an **ID** of ***"output"*** from *index.html* – then sets its **innerHTML** property to contain the **<h2>** tags with text.

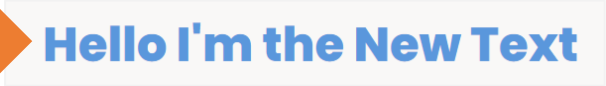


```
/*returns the element withe ID of 'output'*/
function changeText() {
    document.getElementById("output").innerHTML = "<h2> Hello I'm the New Text </h2>";
}
```

NOTE: The <h2> html tags and text are surrounded in double quotes -you can also use single quotes – when setting the **innerHTML** property it completely overwrites everything inside the parent tag(***'output'***)

View in **Live Server** and click the first button

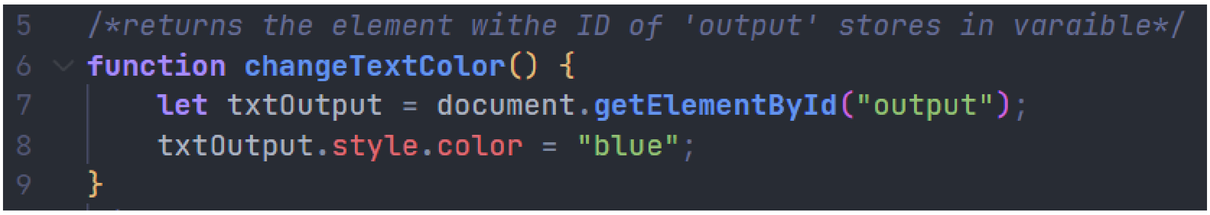
Updated text now displays and is inside the new **<h2>** tag



Hello I'm the New Text

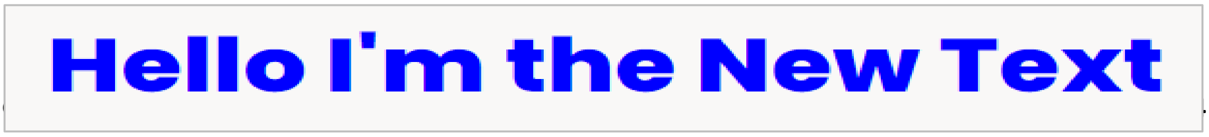
2 Change the text colour:

This time the element returned is stored in a variable named ***txtOutput*** making it easy to access the ***style*** object followed by the ***colour*** property



```
/*returns the element withe ID of 'output' stores in varaible*/
function changeTextColor() {
    let txtOutput = document.getElementById("output");
    txtOutput.style.color = "blue";
}
```

Test in ***Live Server*** - Text now changed to **Royal Blue** – *experiment with HEX values and change to other colours*

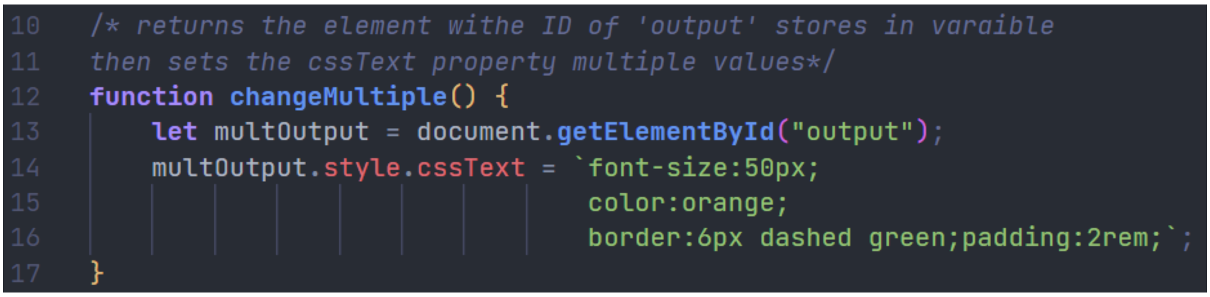


Hello I'm the New Text

3 Setting more than one value with the cssText property

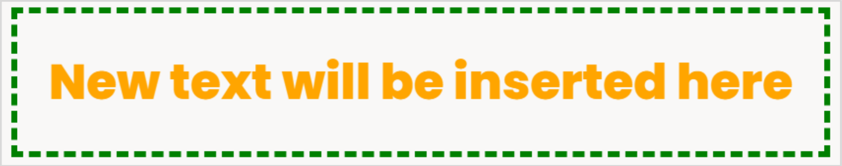
Same process again except this time **cssText** property is being set

IMPORTANT! – Below the css properties/values are surrounded by `back ticks`(*left of the number one key on the keyboard*) not quotation marks – This is a new feature in JavaScript – one of the benefits being you can wrap the text over multiple lines, and it will still work – you cannot do this using quotes



```
/* returns the element withe ID of 'output' stores in varaible
then sets the cssText property multiple values*/
function changeMultiple() {
    let multOutput = document.getElementById("output");
    multOutput.style.cssText = `font-size:50px;
                                color:orange;
                                border:6px dashed green;padding:2rem;`;
}
```

Preview in ***Live Server***, click **Change Multiple Properties** button – to view the orange text with the **6px** dashed border– experiment with other properties

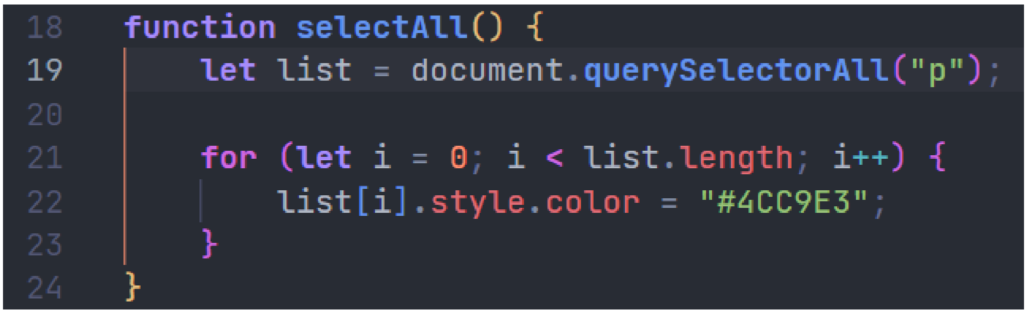


New text will be inserted here

4 Setting multiple elements on the page using querySelectorAll()

querySelectorAll() returns the elements in a node list(similar to an array) matching the supplied css selector

- Below all **<p>** tags are returned in a node list
- A **for loop** is used to loop through the elements in the node list, changing each elements colour to **#4CC9E3**(*light blue*)



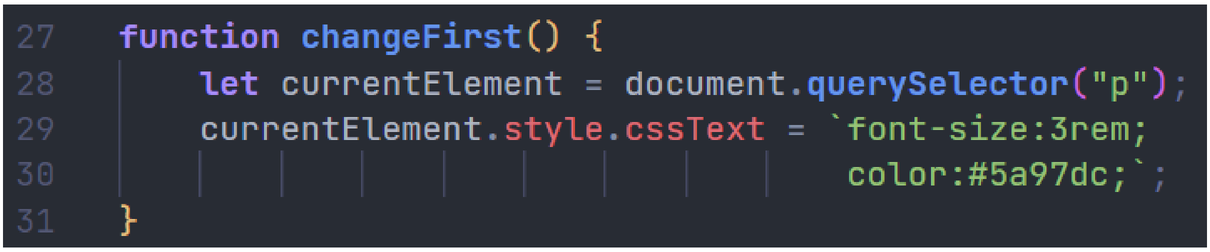
```
function selectAll() {
    let list = document.querySelectorAll("p");

    for (let i = 0; i < list.length; i++) {
        list[i].style.color = "#4CC9E3";
    }
}
```

Preview in **Live Server** – click the **‘Select ALL P Tags’** button

5 Setting an element on the page using querySelector()

querySelector() returns only the **first** element that matches the supplied CSS selector(*will stop looking after it finds the first matched selector*)

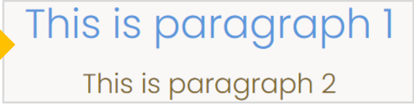
Below **querySelector()** returns the first matched **<p>** tag
cssText property applies the styling - remember those back ticks!



```
function changeFirst() {
    let currentElement = document.querySelector("p");
    currentElement.style.cssText = `font-size:3rem;
                                    color:#5a97dc;`;
}
```

Preview in **Live Server** – click the **'Change First'** button

Paragraph one is now different from the others

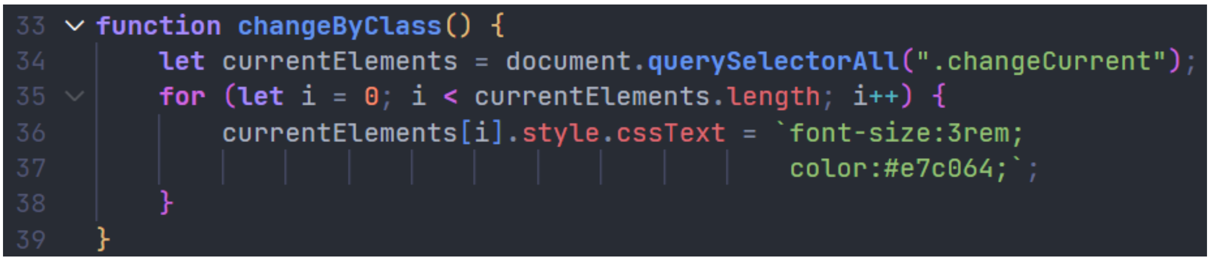


This is paragraph 1

This is paragraph 2

6 Change elements in paragraph list by class name

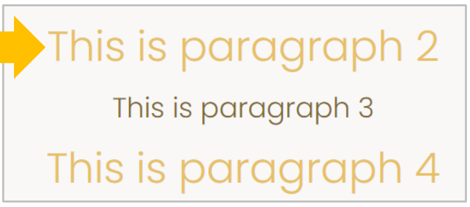
- Below, returns a *node list* containing the elements with the supplied class name(***.changeCurrent*** class is found on **<p>** tags **2** & **4** on *index.html*)
- Loops over the list and updates the **css** values using **cssText** property



```
function changeByClass() {
    let currentElements = document.querySelectorAll(".changeCurrent");
    for (let i = 0; i < currentElements.length; i++) {
        currentElements[i].style.cssText = `font-size:3rem;
                                            color:#e7c064;`;
    }
}
```

Preview in **Live Server** – click the **'Change by Class** button

Paragraphs 2 & 4 are now changed



This is paragraph 2

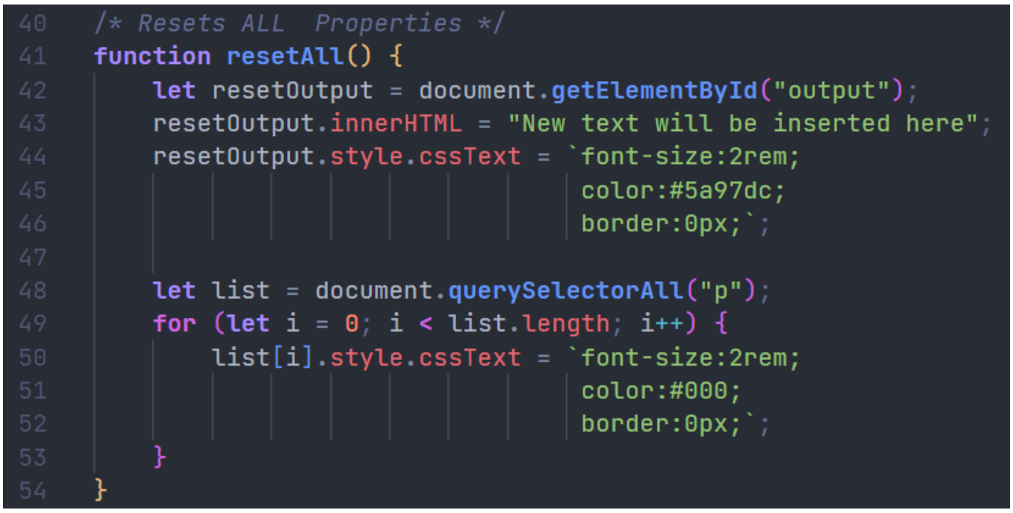
This is paragraph 3

This is paragraph 4

7 Reset properties and values back to initial state

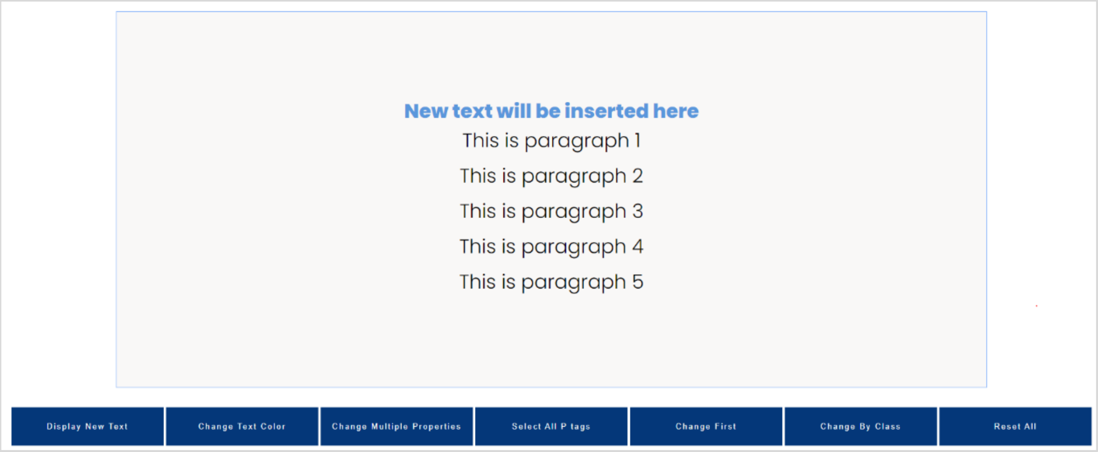
- Below uses a combination of the **2** methods - ***getElementById()*** returns the output panel back to original state

- ***querySelectorAll()*** retrieves the **<p>** tags and returns them back to their original state



```
/* Resets ALL  Properties */
function resetAll() {
    let resetOutput = document.getElementById("output");
    resetOutput.innerHTML = "New text will be inserted here";
    resetOutput.style.cssText = `font-size:2rem;
                                 color:#5a97dc;
                                 border:0px;`;

    let list = document.querySelectorAll("p");
    for (let i = 0; i < list.length; i++) {
        list[i].style.cssText = `font-size:2rem;
                                 color:#000;
                                 border:0px;`;
    }
}
```

Preview in **Live Server** – click the **'Reset All** button



New text will be inserted here

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

This is paragraph 5

TASKS:

1. Add more buttons to the **index.html** page – *copy & paste*

2. One of the buttons should change the *background colour* of the section with the '***dataPanel'*** **ID**

3. Update the select All **<p>** tags function – instead of just the colour update *borders, background colours* and any other properties you can think of