

## Question 2

### 1.Unshift(), shift() and split()

#### Unshift():

##### Explanation:

The unshift() method allows us to add elements to the beginning of an array. It can also be used to return the length of the array with the new element/elements added to the beginning.

##### Example:

JS Assignment 1, Q2 · Holly

JS

```
1 const CFG = ["First", "Girls"]; // To create array.
2 const length = CFG.unshift("Code"); // Use unshift() to add element "Code" to the beginning of
  the array.
3 console.log(CFG); //To print unshifted array with new element in beginning.
4 console.log(length); //To print length of unshifted array.
```

Console

```
// [object Array] (3)
["Code","First","Girls"]
```

3

>

Console Assets Comments ⌂ Keys Last saved SEPTEMBER 9, 2023 – 5:40:22 PM ⌂ Delete Add to Collection ⌂ Fork Embed Export Share

#### Shift():

##### Explanation:

The shift() method can be used to remove the first element in an array. It can also be used to return the first element that was removed. When used it also lowers the indexes of all the remaining elements in the array.

##### Example:

A screenshot of the JS Bin code editor. The title bar says "JS Assignment 1, Q2" and "Holly". The tabs at the top are "HTML", "CSS", and "JS", with "JS" selected. The code in the JS tab is:

```
1 const CFG = ["Code", "First", "Girls"]; // To create array.
2 const FG = CFG.shift(); // Use shift() to remove element "Code" from the beginning of the array.
3 console.log(CFG); //To print shifted array with new element removed from beginning.
4 console.log(FG); //To return removed element.
```

The Console output shows:

```
// [object Array] (2)
["First","Girls"]
"Code"
```

At the bottom, there are buttons for "Console", "Assets", "Comments", and "⌘ Keys".

## Split():

### Explanation:

The Split() method allows us to convert a string into an array. Depending upon what we put into the brackets, dictates where the contents of the string will be converted into elements for the array. For example, split (" ") splits on spaces, (",") on commas and ("|") on the line symbol.

### Example:

A screenshot of the JS Bin code editor. The title bar says "JS Assignment 1, Q2" and "Holly". The tabs at the top are "HTML", "CSS", and "JS", with "JS" selected. The code in the JS tab is:

```
1 const CFGString = "Code First Girls"; // Create the String
2
3 console.log(CFGString.split(" ")); // Use Split(" ")to split the string into an array on spaces in
        // the string. Console.log to return it in the console. Alternatively split can be used with (",") or
        // ("|") to split the string into array elements on commas or lines.
4
```

The Console output shows:

```
// [object Array] (3)
["Code", "First", "Girls"]
```

At the bottom, there are buttons for "Console", "Assets", "Comments", and "⌘ Keys".

## 2.Object Methods

### Explanation:

Objects are variables in which we can store multiple values. Each value in an object is assigned its own name / property. An object method is a function that is stored inside an object, as a property.

### Example:

The screenshot shows a code editor interface with the following details:

- Title Bar:** JS Assignment 1, Q2 · Holly
- Toolbar:** Includes icons for Heart, Run, Save, Settings, and others.
- Code Editor:** A dark-themed code editor with syntax highlighting for JS.

```
JS
1 //Create an object:
2 const programminglanguages = {
3   language: "JavaScript",
4   Abbreviation: "JS",
5   Use: "Creating interactive web pages",
6   Created : 1995,
7   //Create an object method by creating a function -here the function combines the abbreviation with the
language name.
8   Name: function () {console.log(this.Abbreviation + " " + "Stands for" + " " + this.language);
9 }
10 //Call the function
11 programminglanguages.Name();
```
- Console:** Displays the output of the console.log statement: "JS Stands for JavaScript"
- Bottom Navigation:** Includes tabs for Console, Assets, Comments, % Keys, and buttons for Last saved, Delete, Add to Collection, Fork, Embed, Export, and Share.

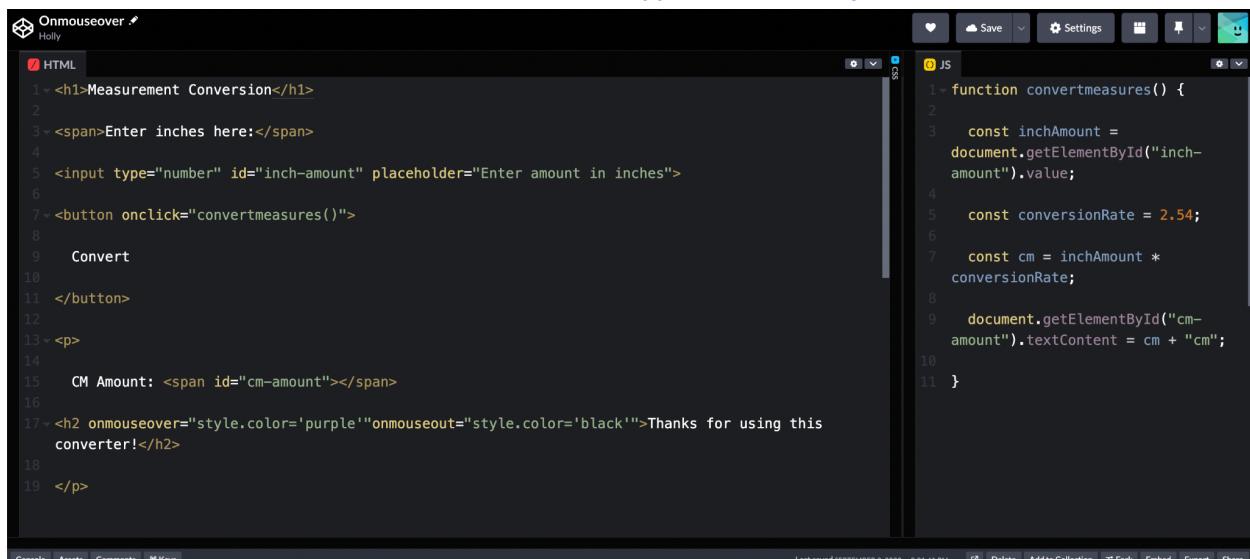
### 3. DOM Events

Document Object Model events are triggers that run code when they are activated by either an element loading, a form of specified user input, a mouse movement or click.

#### 1. Onmouseover

Explanation: When the user passes their mouse over a specific aspect of the web page it will trigger code to run.

Example: Link to codepen: <https://codepen.io/hollyyates/pen/qBLrjvV>



The screenshot shows the CodePen interface with the title "Onmouseover". The left panel displays the HTML code for a measurement conversion converter. The right panel displays the corresponding JavaScript code. The HTML includes an input field for inches, a conversion button, and a span for the resulting CM amount. The JavaScript uses event listeners to change the color of the "Thanks for using this converter!" text to purple when the mouse hovers over it and back to black when it moves away.

```
HTML
1 <h1>Measurement Conversion</h1>
2
3 <span>Enter inches here:</span>
4
5 <input type="number" id="inch-amount" placeholder="Enter amount in inches">
6
7 <button onclick="convertmeasures()">
8
9   Convert
10
11 </button>
12
13 <p>
14
15   CM Amount: <span id="cm-amount"></span>
16
17 <h2 onmouseover="style.color='purple'" onmouseout="style.color='black'">Thanks for using this
18   converter!</h2>
19 </p>
```

```
JS
1 function convertmeasures() {
2
3   const inchAmount =
4     document.getElementById("inch-amount").value;
5
6   const conversionRate = 2.54;
7
8   const cm = inchAmount *
9     conversionRate;
10
11   document.getElementById("cm-amount").textContent = cm + "cm";
```

Before mouse over 'Thanks for using this converter!':



The screenshot shows the measurement conversion converter. The "CM Amount:" text is black. Below it, the "Thanks for using this converter!" text is also black.

A Pen by Holly  
Onmouseover

**Measurement Conversion**

Enter inches here:  Enter amount in inch

CM Amount:

**Thanks for using this converter!**

After mouse over 'Thanks for using this converter!' : Colour changed to purple.



The screenshot shows the measurement conversion converter. The "CM Amount:" text is black. Below it, the "Thanks for using this converter!" text is purple.

A Pen by Holly  
Onmouseover

**Measurement Conversion**

Enter inches here:  Enter amount in inch

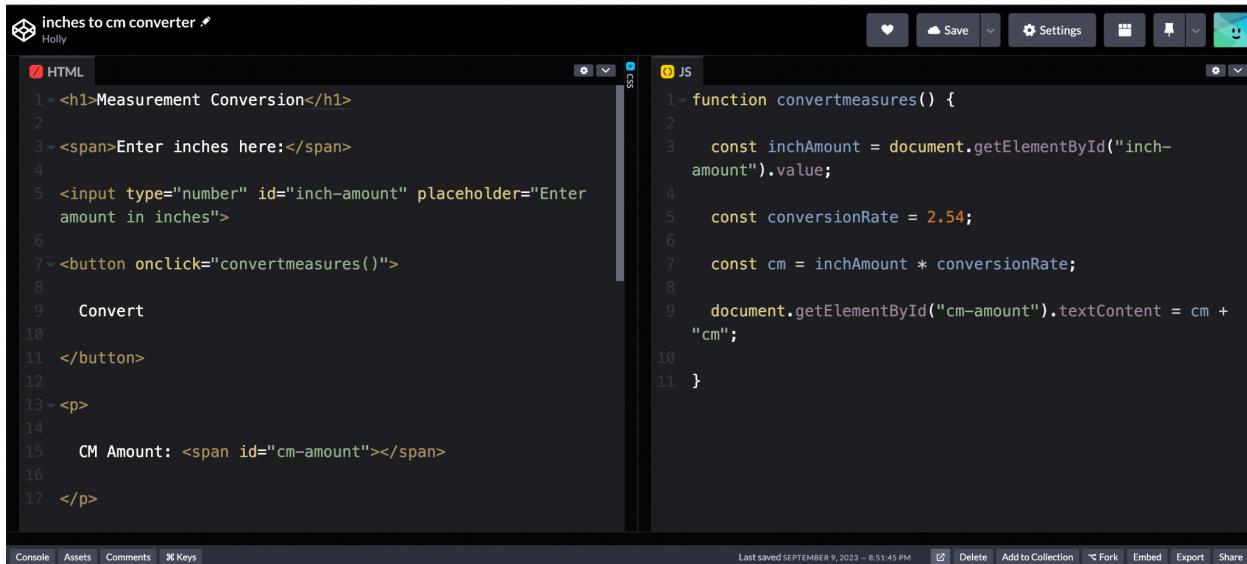
CM Amount:

**Thanks for using this converter!**

## 2.Onclick

Explanation: When the user clicks on a specific aspect of the web page (most commonly a button or image), it will trigger the code to run.

Example: Link to codepen: <https://codepen.io/hollyyates/pen/qBLrjLP?editors=1011>



The screenshot shows a CodePen interface with two main panels. The left panel, titled 'HTML', contains the following code:`1 <h1>Measurement Conversion</h1>
2
3 <span>Enter inches here:</span>
4
5 <input type="number" id="inch-amount" placeholder="Enter amount in inches">
6
7 <button onclick="convertmeasures()">
8
9 Convert
10
11 </button>
12
13 <p>
14
15 CM Amount: <span id="cm-amount"></span>
16
17 </p>`The right panel, titled 'JS', contains the following code:`1 function convertmeasures() {
2
3 const inchAmount = document.getElementById("inch-amount").value;
4
5 const conversionRate = 2.54;
6
7 const cm = inchAmount * conversionRate;
8
9 document.getElementById("cm-amount").textContent = cm +
10 "cm";
11 }`At the bottom of the interface, there are buttons for 'Console', 'Assets', 'Comments', 'Keys', 'Last saved SEPTEMBER 9, 2023 – 8:51:45 PM', and links for 'Delete', 'Add to Collection', 'Fork', 'Embed', 'Export', and 'Share'.

After inputting 5 inches, when the 'Convert' button was pressed, the amount in CM was calculated and then appeared on the screen.



The screenshot shows the final state of the application. The title bar says 'A Pen by Holly inches to cm converter'. The main content area displays the heading 'Measurement Conversion'. Below it, there is an input field with the value '5' and a 'Convert' button. Underneath the button, the text 'CM Amount: 12.7cm' is displayed. At the top right, there are standard CodePen sharing and management buttons.

## 3.Onchange:

Explanation: When the user enters a value into the input box (and therefore, the value changes), it will trigger the code to run.

Example: Link to codepen: <https://codepen.io/hollyyates/pen/pogewOv?editors=1111>

A screenshot of a code editor interface. The top bar includes a logo, the title "Metric Converter", a user name "Holly", and various tool icons. The editor is divided into three main sections: HTML, CSS, and JS.

**HTML:**

```
<h1>Metric Converter</h1>
<span>Kgs Amount:</span>
<input onchange = "metricconversion()" type="number" id="kgs-amount" placeholder="Enter amount in Kgs">
<p>
    Llb Amount: <span id="total"></span>
</p>
```

**JS:**

```
function metricconversion() {
    const kgsAmount =
        document.getElementById("kgs-amount").value;
    const conversion = 2.2;
    const total = kgsAmount *
        conversion;
    document.getElementById("total")
        .textContent = total +
        "llbs";
}
```

At the bottom, there are buttons for Console, Assets, Comments, ⌘ Keys, Last saved (September 9, 2023 - 8:36:02 PM), Delete, Add to Collection, Fork, Embed, Export, and Share.

On entering 9Kgs, and therefore, changing the value, 19.8lbs appears.

