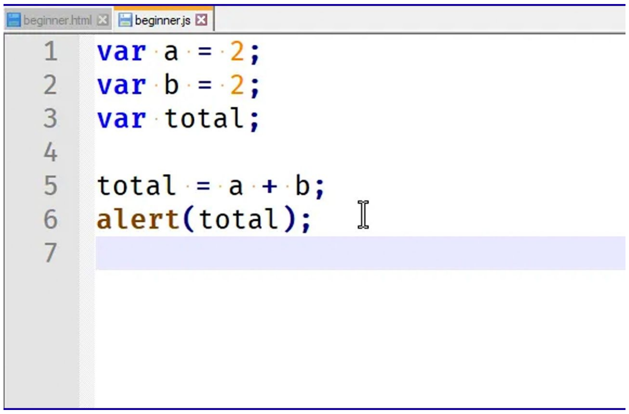
ADV-C78

JAVASCRIPT STATEMENTS



```
var a = 2;
var b = 2;
var total;

total = a + b;
alert(total);
```

What is our GOAL for this MODULE?

Data Structures are one of the most applied concepts of software engineering. We learned and understood the concept of one of the most common data structures - Arrays and applied associated methods.

What did we ACHIEVE in the class TODAY?

- Implemented Arrays to methods

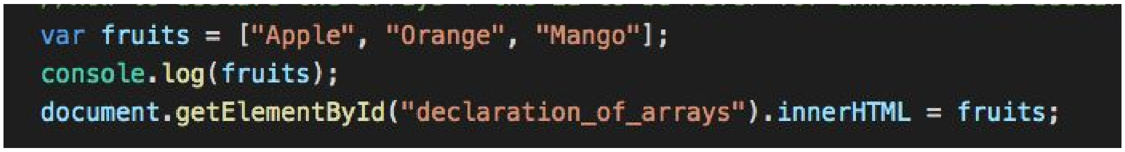
Which CONCEPTS/ CODING BLOCKS did we cover today?

- Arrays and why they are used
- Index in arrays
- Why and how for loops are used to display the elements of arrays
- Various JavaScript functions that can be used on arrays

How did we DO the activities?

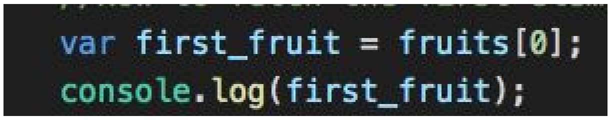
1. Arrays:
    - An array can hold many values under a single name
      For example: var friends = ["Ram", "Rohit", "Rocky"];
    - You can access the values by referring to an index number like:
      Friends[0] // will be Ram
      Friends[1] // will be Rohit
      Friends[2] // will be Rocky

2. Declare the array, console it and display it on HTML elements.



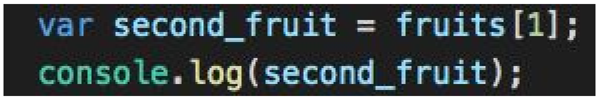
```
var fruits = ["Apple", "Orange", "Mango"];
console.log(fruits);
document.getElementById("declaration_of_arrays").innerHTML = fruits;
```

3. Fetch the first element of the array and console it.



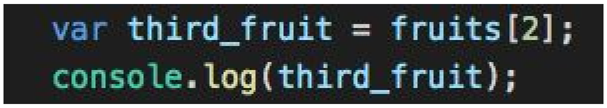
```
var first_fruit = fruits[0];
console.log(first_fruit);
```

4. Fetch the second element of the array and console it.



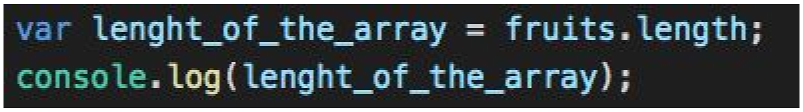
```
var second_fruit = fruits[1];
console.log(second_fruit);
```

5. Fetch the third element of the array and console it.



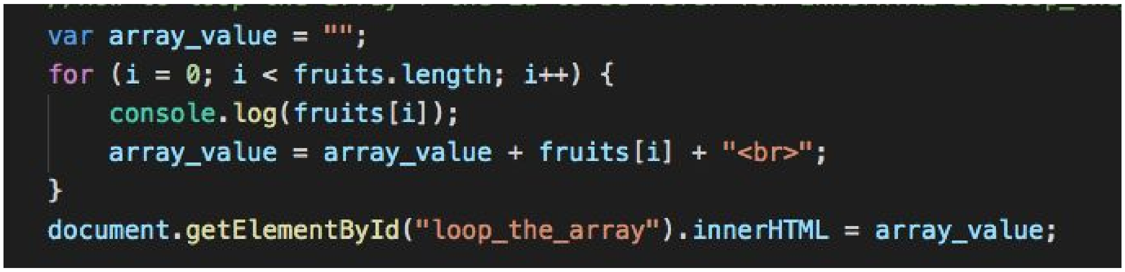
```
var third_fruit = fruits[2];
console.log(third_fruit);
```

6. Find the length of the array. Length of the array means the number of elements inside the array.



```
var lenght_of_the_array = fruits.length;
console.log(lenght_of_the_array);
```

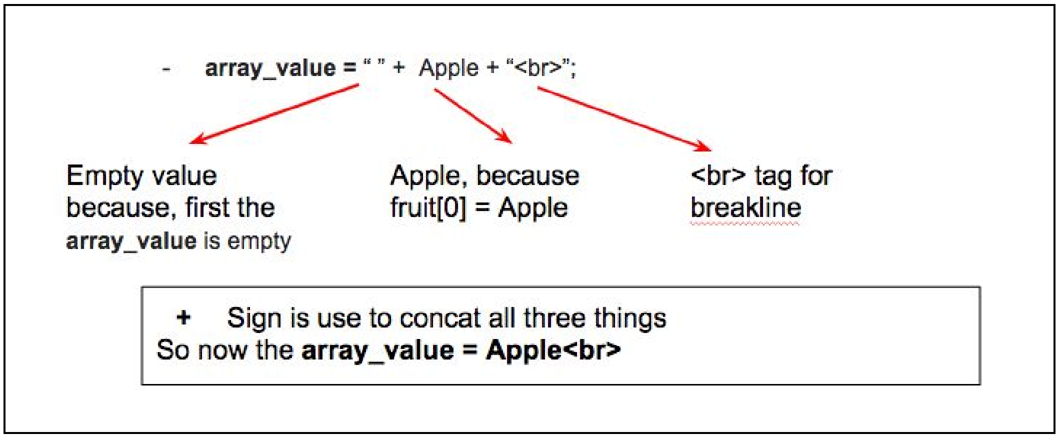
7. Loop the array.
   - We will use for loop:
     - Set i=0; Set it to 0 because the arrays first index is 0.
     - Set the termination condition to fruits.length, because we want the loop to stop when it reaches the last element of the array.
     - Increase the value of variable i by 1, so one by one we get the elements from the array.
     - Console the elements of the array one by one.
     - Update the variable array_value by the element of the array plus a br tag.



```
var array_value = "";
for (i = 0; i < fruits.length; i++) {
    console.log(fruits[i]);
    array_value = array_value + fruits[i] + "<br>";
}
document.getElementById("loop_the_array").innerHTML = array_value;
```

- **First i = 0**
  console.log(fruits[i])
  means console.log(fruits[0]) // because i =0
  fruits[0] , means Apple // As we discussed in point 2 that fruits[0] = Apple

- **array_value** = array_value + fruits[i] + “<br>”;
  means **array_value** = “ ” + Apple + “<br>”;



- **array_value** = “ ” + Apple + “<br>”;

Empty value because, first the **array_value** is empty

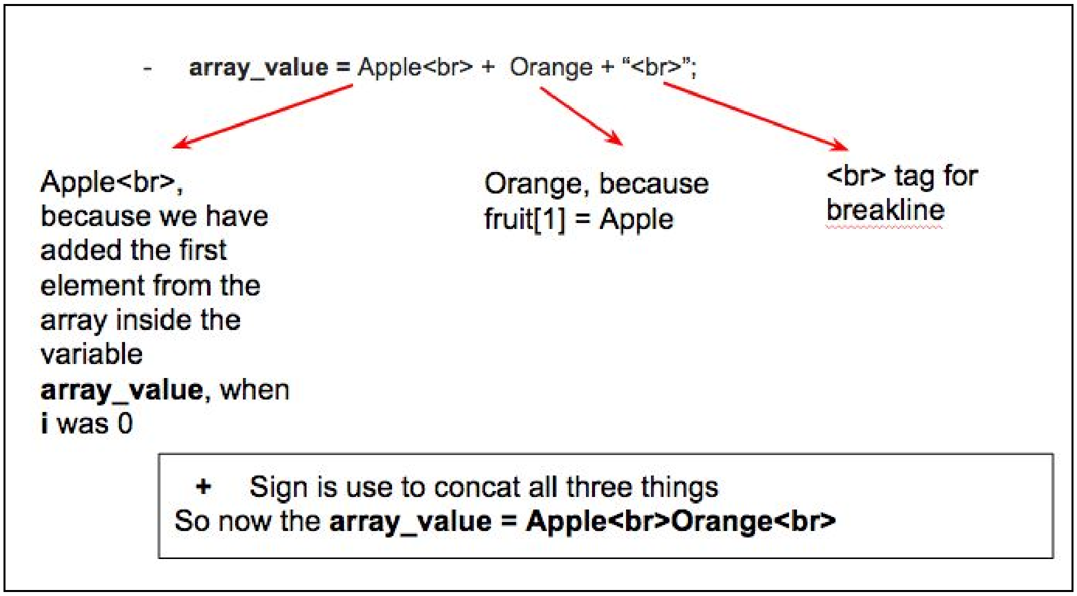
Apple, because fruit[0] = Apple

 tag for breakline

+ Sign is use to concat all three things
So now the **array_value = Apple<br>**

- Now i will be incremented by 1 because we have used **i++** in for loop
  So now i = 1
  console.log(fruits[i])
  means console.log(fruits[1]) // because i =1
  fruits[1] , means Orange // As we discussed in point 3 that fruits[1] = Orange

- **array_value** = array_value + fruits[i] + “<br>”;
  means **array_value** = Apple<br> + Orange + “<br>



- **array_value =** Apple<br> + Orange + "<br>";

Apple
, because we have added the first element from the array inside the variable **array_value**, when **i** was 0

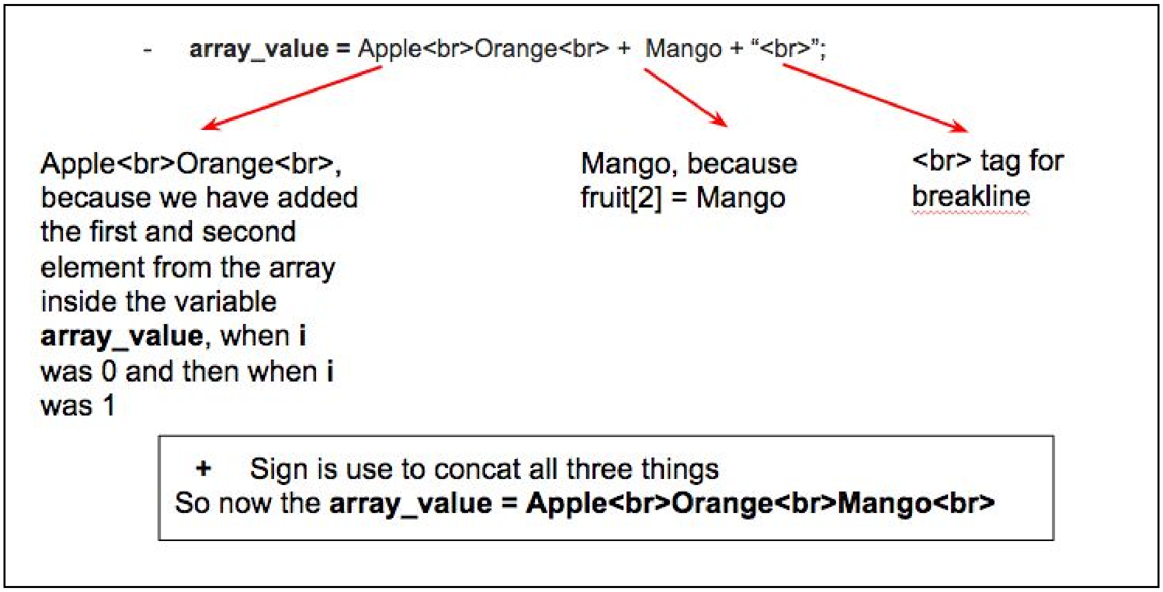
Orange, because fruit[1] = Apple

 tag for breakline

+ Sign is use to concat all three things
So now the **array_value = Apple<br>Orange<br>**

- Again i will be incremented by 1 because we have use **i++** in for loop
  So now i = 2
  console.log(fruits[i])
  means console.log(fruits[2]) // because i =2
  fruits[2] , means Mango // As we discussed in point 4 that fruits[2] = Mango

- **array_value** = array_value + fruits[i] + "<br>";
  means **array_value** = Apple<br>Orange<br> + Mango + "



- **array_value =** Apple<br>Orange<br> + Mango + "
";

Apple<br>Orange
, because we have added the first and second element from the array inside the variable **array_value**, when **i** was 0 and then when **i** was 1

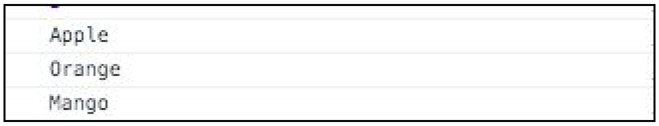
Mango, because fruit[2] = Mango

 tag for breakline

+ Sign is use to concat all three things
So now the **array_value = Apple<br>Orange<br>Mango
**

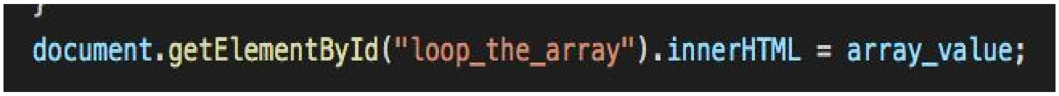
- Again i will be incremented by 1 because we have use i++ in for loop.
- **So now because i = 3:**
    - The loop will stop. This is because the termination condition was when it reaches the last element of the array loop should stop.

- So the output on console will be:



```
Apple
Orange
Mango
```

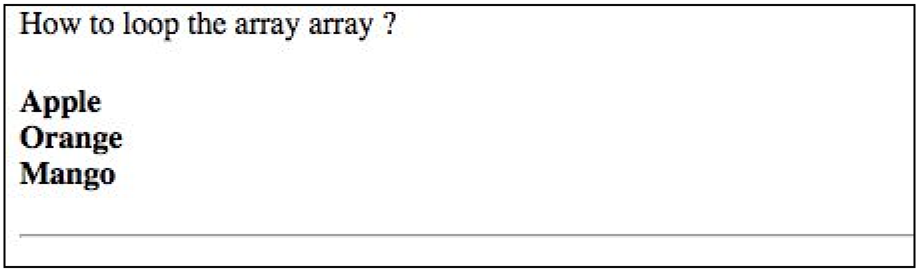
- After the loop we print the output on HTML element.



```
document.getElementById("loop_the_array").innerHTML = array_value;
```

- So the HTML element with id=**loop_the_array** will be updated with the value of the variable **array_value**.

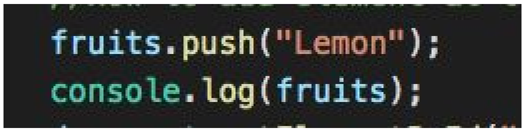
- And the output on the web screen will be:



How to loop the array array ?

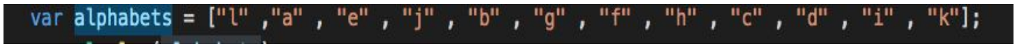
Apple
Orange
Mango

8. Add element to the array using **push** method.
    - **push()** function will add an element at the end of the array.
        - Before using push function the array - Apple, Orange, Mango
        - After using push function the array - Apple, Orange, Mango, Lemon



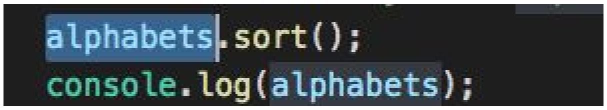
```
fruits.push("Lemon");
console.log(fruits);
```

9. Declaring an array alphabets



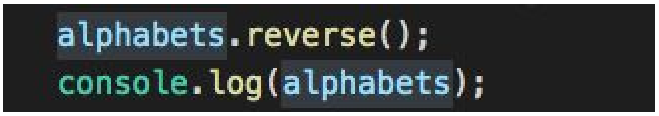
```
var alphabets = ["l" ,"a" , "e" , "j" , "b" , "g" , "f" , "h" , "c" , "d" , "i" , "k"];
```

10. Sort the array.



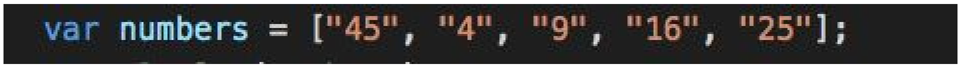
```
alphabets.sort();
console.log(alphabets);
```

11. Reverse the array.



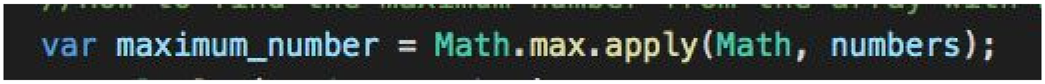
```
alphabets.reverse();
console.log(alphabets);
```

12. Declaring an array numbers



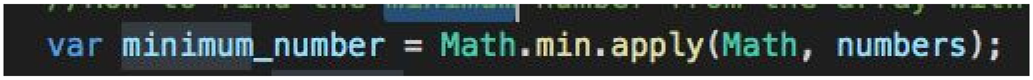
```
var numbers = ["45", "4", "9", "16", "25"];
```

13. Finding the maximum number from the array
    - **Math.max.apply(Math, array_name)** function is used to find the maximum number from the array.



```
var maximum_number = Math.max.apply(Math, numbers);
```

14. Finding the minimum number from the array.
    - **Math.min.apply(Math, array_name)** function is used to find the minimum number from the array.



```
var minimum_number = Math.min.apply(Math, numbers);
```

What's NEXT?

We will continue our coding journey tackling very advanced coding concepts like data structures, implementing Arrays while applying Array methods coupling with Front End leading to an Array-based Web App.

EXTEND YOUR KNOWLEDGE

Here are some Best References we've compiled together to enhance your knowledge and understanding of the concepts we learnt today in the class. This will help you become pro at coding and creating industry-grade tech products!

Short Videos: Watch these Short Videos to understand the application of the concepts learned in class in real-world applications.

1. JavaScript Arrays: https://www.youtube.com/watch?v=oigfaZ5ApsM
2. JavaScript DOM Tutorial: https://www.youtube.com/watch?v=t90K6HExEJo
3. How to sort Arrays: https://www.youtube.com/watch?v=BbuLjEqFlw0

Coding Playground: Try out these code examples to get more practice in making Websites and Playstore ready apps.

1.
JavaScript Arrays
w3schools.com
: https://www.w3schools.com/js/js_arrays.asp

2.
HTML DOM getElementBy...
w3schools.com
: https://www.w3schools.com/jsref/met_document_getelementbyid.asp

3.
JavaScript Array push() M...
w3schools.com
: https://www.w3schools.com/jsref/jsref_push.asp