JavaScript Arrays .

JavaScript's approach to arrays is a bit different. There isn't a data type called array in JavaScript. Instead,arrays are objects in JavaScript.
A JavaScript Object can hold different data types (string, integer, boolean) at once.
Example:

```
let user = {
        name: "Sumaiya",
        age: 22,
        location: "Lagos"
}
```

Since an array in JS is an object (with a different syntax), they can also contain different data types.
Example:

```
let user = [22, 3.214,  "Lagos", true]
```

"JavaScript arrays are used to store multiple values in a single variable." w3schools.

Creating an array in JavaScript.

There are 2 different ways to create a JavaScript array:

a). Array literals.

b). Array Constructor.

a). Array literals.

This way is used more commonly. We declare an array with empty square brackets and assign it to a variable.

Example:

```
let user = [ ];          //this is how you declare an array using array literals
```

The values of an array are written inside the brackets, separated by commas.

Example:

```
let myCar = [ "Mercedes Benz", 2020, true]
```

b). **Array Constructor**.

There is also a global Array Objects in JavaScript, which has its own built–in properties & methods. We use an instance of the Array Object with the new keyword.

Example 1:

```
let myCar = new array( );
```

Example 2:

```
let myCar = new array( "Mercedes Benz", 2020, true);
```

Personally i would recommend using array literals due to it's simplicity, readability and execution speed.

Accessing the Elements of an Array

- Each element of an array has a location called as index.
- We can access an element of the array by referring its index.
- Array indexes always start with 0, which points to the first element of the array.

Getting the Size of an Array.

JavaScript provides an easy way to get how many elements an array has: the length property.

The length property is one of the built–in properties of the Array Object in JavaScript. We can call it simply with dot (.) notation.

```
let numbersArray = [6, 7, 8, 9, 10];

numbersArray.length;
```

Looping an Array.

Like in other programming languages, we can use the for / while loops in JavaScript to return values from an array at once.

```
let numbersArray = [1, 2, 3, 4, 5];
for(let i=0; i < numbersArray.length; i++) {
    console.log((numbersArray[i]);
}
```

The for loops must have a starting & an ending point. “i” will start from 0 and keeps running until the end of the array. Using the length property is useful here. If the size of the array changes, the length property will get a new size automatically and the loop won’t break.

Then give the same “i” value as an index property to the array, so it gets directly the elements and prints them to console. Remember this is not the only way to loop arrays

JavaScript Basic Array Methods

1). **Array.push()** : Adding Element at the end of an Array. As array in JavaScript are mutable object, we can easily add or remove elements from the Array. And it dynamically changes as we modify the elements from the array.

Example:

```
// Adding elements at the end of an array
// Declaring and initializing arrays
var number_arr = [ 10, 20, 30, 40, 50 ];
var string_arr = [ "piyush", "gourav", "smruti", "ritu" ];
// push()
// number_arr contains [10, 20, 30, 40, 50, 60]
number_arr.push(60);
// We can pass multiple parameters to the push()
// number_arr contains
// [10, 20, 30, 40, 50, 60, 70, 80, 90]
number_arr.push(70, 80, 90);
// string_arr contains
// ["piyush", "gourav", "smruti", "ritu", "sumit", "amit"];
string_arr.push("sumit", "amit");

// Printing both the array after performing push operation
console.log("After push op " + number_arr);
console.log("After push op " + string_arr);
```

2). **Array.unshift()** : Adding elements at the front of an Array.

Example

```
// Adding element at the beginning of an array
// Declaring and initializing arrays
var number_arr = [ 20, 30, 40 ];
var string_arr = [ "amit", "sumit" ];

// unshift()
// number_arr contains
// [10, 20, 20, 30, 40]
number_arr.unshift(10, 20);

// string_arr contains
// ["sunil", "anil", "amit", "sumit"]
string_arr.unshift("sunil", "anil");

// Printing both the array after performing unshift operation
console.log("After unshift op " + number_arr);
console.log("After unshift op " + string_arr);
```

Array.pop() : Removing elements from the end of an array

```
// Removing elements from the end of an array
// Declaring and initializing arrays
var number_arr = [ 20, 30, 40, 50 ];
var string_arr = [ "amit", "sumit", "anil" ];
// pop()
// number_arr contains
// [ 20, 30, 40 ]
number_arr.pop();

// string_arr contains
// ["amit", "sumit"]
string_arr.pop();
// Printing both the array after performing pop operation
console.log("After pop op " + number_arr);
console.log("After popo op " + string_arr);
```

4). Array.shift() : Removing elements at the beginning of an array

```
// Removing element from the beginning of an array
// Declaring and initializing arrays
var number_arr = [ 20, 30, 40, 50, 60 ];
var string_arr = [ "amit", "sumit", "anil", "prateek" ];

// shift()
// number_arr contains
//  [30, 40, 50, 60];
number_arr.shift();

// string_arr contains
// ["sumit", "anil", "prateek"]
string_arr.shift();

// Printing both the array after performing shifts operation
console.log("After shift op " + number_arr);
console.log("After shift op " + string_arr);
```

Best Wishes.

Lux Tech Academy.