Front End Technologies
Javascript - Day 5

Agenda

- **Arrays**

substring() vs substr()

substr(3)

substr() is used to extract string from another string.
syntax:
substr(index) -> start at the specified index and move till the end of the string and extract the string.
Example:
var str = "Hello World";
console.log(str.substr(3)); //lo World

Syntax2:
str.substr(index, length);

Example2:
console.log(str.substr(3, 4));
The above code will start at the index3 and extract the length number of characters. In our case it will extract 4 characters.

So Output will be lo W

Arrays.
=========

Arrays are a kind of data structure where we can store multiple data. There is no restriction about the data we are storing.

How to declare an array??
Syntax:

var array_name = [];
or
let array_name = [];
or
const array_name = [];

Example

```
var arr = [1,2,3,4,5];
console.log(arr) // [1,2,3,4,5]
```

2nd Approach of declaring arrays

```
var arr2 = new Array(1,2,3,4,5);
console.log(arr2) // [1,2,3,4,5]
```

In JS all the arrays declared are dynamic in nature. Ie as we keep on adding data , the size of the array keeps on growing.

Example:

```
var arr = [1, "Sachin", true, {}, 'c', [10, "Ramu", false],7777,"dakdah",true];
console.log('First Way ',arr);  // [1, 'Sachin', true, {…}, 'c', Array(3), 7777,
'dakdah', true]
```

How can i access the data present in an array?

Arrays can be accessed as shown below.

```
var arr = [1, "Sachin", true, {}, 'c', [10, "Ramu", false],7777,"dakdah",true];
console.log(arr[5]) //[10, "Ramu", false]
```

How to get the size of an array?
We can use the length property on the array.
Example

```
var arr = [1, "Sachin", true, {}, 'c', [10, "Ramu", false],7777,"dakdah",true];
console.log('First Way ',arr.length); //9
```

Array Methods..
==========

1) push()
When we need to push any data to an array we use push method.. push() always appends the data to the end of the array.
Example:

```
var arr = [];
arr.push(10);
arr.push("Sachin");
arr.push(true);
arr.push(null);
console.log(arr); // [10, "Sachin", true, null]
```

2) pop()
Whenever we need to remove an element from the array we use pop(). pop() removes the element which is present at the last index of an array.

Example:

```
var arr = [10,"Sachin",true,null, 15.7655343421];
arr.pop();
console.log(arr); // [10,"Sachin",true,null];
```

3) unshift()
When we need to push the data to the beginning of the array we use the unshift().
Example:

```
var arr = [10,"Sachin",true,null, 15.7655343421];
arr.unshift("Virat");
console.log(arr); // ["Virat",10,"Sachin",true,null, 15.7655343421];
```

4) shift();

To remove an element from the beginning of an array we can use shift().

Example:

```
var arr = ["Virat",10,"Sachin",true,null, 15.7655343421];
arr.shift();
```

```
console.log(arr);
```

5) indexOf()

This method is used to get the index of a specific element present in an array. If the element is present inside the array then its index will be returned. If the element is not available inside the array then the indexOf() will return -1/

Example:

```
var arr = ["Virat",10,"Sachin", true, null, 15.7655343421];
console.log(arr.indexOf("dahdlhakdlhkahldahldhladhla")) //-1
```

6) join()

Whenever we need to join all the elements inside the array join() should be used. join() will join all the elements of an array and return it in the string format.
Example;

```
var arr = ["Virat",10,"Sachin", true, null, 15.7655343421];
console.log(arr.join());
```

join() by default will seperate the strings using comma. But if we want to change it then we need to pass the character to join() as shown below.

```
console.log(arr.join('-')); //Virat-10-Sachin-true--15.7655343421
```

7) includes()
includes() checks whether the data passed to it is present inside the array or not. If its present then it returns true else it returns false.

```
var arr = ["Virat",10,"Sachin", true, null, 15.7655343421];
console.log(arr.includes('Sachin Ramesh... ')); //false
```

8) reverse()
In order to reverse an array we use reverse()
Example:

```
var arr = ["Virat",10,"Sachin", true, null, 15.7655343421];
console.log(arr);
console.log(arr.reverse());
```

9) slice()

var arr = ["Virat",10,"Sachin", true, null, 15.7655343421];

It works exactly the way slice() works in case of string.
Example
var arr = ["Virat",10,"Sachin", true, null, 15.7655343421,98,false];
console.log(arr.slice(2, 6)); // "Sachin", true, null, 15.7655343421,98
Starts at 2nd index and goes till 6th index. But ignores the 6th index data.

If we use the slice() it will not alter the original array. Rather it extracts the data based on the indexes we pass.

slice(start,end);

extraction starts at the start index and ends at the end-1 index;

10) splice()
splice(start,length);
extraction starts at the start index extracts the length number of elements from the array. In case of splice() it also alters the original array.

Example:

```
var arr = [1,2,3,4,5,6,7,8];
console.log("Original Before Splicing ",arr);
var splicedArr = arr.splice(2, 4);
console.log("Spliced Arr",splicedArr);
console.log("Original Array After Splicing ",arr);

//Original Before Splicing  (8) [1, 2, 3, 4, 5, 6, 7, 8]
 Spliced Arr (4) [3, 4, 5, 6]
 Original Array After Splicing  (4) [1, 2, 7, 8]
```

In order to remove an element at a particular index then we must do it this way.

```
var arr = [1,2,3,4,5,6,7, 8];
arr.splice(arr.indexOf(7), 1);
console.log(arr); // [1,2,3,4,5,6,8]
```