

JavaScript Essentials

Day 2 Assignment

Question 1: Write a JS code which takes input from the user and logs it in the console.

Answer:

```
let name = prompt("What is your name?");  
console.log(name);
```

Question 2: Explain with examples the remaining methods of string and array.

Answer:

Methods of array:

1. forEach()

This method can help you to loop over array's items.

```
const arr = [1, 2, 3, 4, 5, 6];  
arr.forEach(item => {  
    console.log(item); // output: 1 2 3 4 5 6  
});
```

2. includes()

This method check if array includes the item passed in the method.

```
const arr = [1, 2, 3, 4, 5, 6];  
arr.includes(2); // output: true  
arr.includes(7); // output: false
```

3. filter()

This method create new array with only elements passed condition inside the provided function.

```
const arr = [1, 2, 3, 4, 5, 6];  
// item(s) greater than 3  
const filtered = arr.filter(num => num > 3);  
console.log(filtered); // output: [4, 5, 6]  
console.log(arr); // output: [1, 2, 3, 4, 5, 6]
```

4. map()

This method create new array by calling the provided function in every element.

```
const arr = [1, 2, 3, 4, 5, 6];  
// add one to every element  
const oneAdded = arr.map(num => num + 1);  
console.log(oneAdded); // output [2, 3, 4, 5, 6, 7]  
console.log(arr); // output: [1, 2, 3, 4, 5, 6]
```

5. reduce()

The reduce() method applies a function against an accumulator and each element in the array (from left to right) to reduce it to a single value.

```
const arr = [1, 2, 3, 4, 5, 6];  
const sum = arr.reduce((total, value) => total + value, 0);  
console.log(sum); // 21
```

6. some()

This method check if at least one of array's item passed the condition. If passed, it return 'true' otherwise 'false'.

```
const arr = [1, 2, 3, 4, 5, 6];  
// at least one element is greater than 4?  
const largeNum = arr.some(num => num > 4);  
console.log(largeNum); // output: true  
// at least one element is less than or equal to 0?  
const smallNum = arr.some(num => num <= 0);  
console.log(smallNum); // output: false
```

7. every()

This method check if all array's item passed the condition. If passed, it return 'true' otherwise 'false'.

```
const arr = [1, 2, 3, 4, 5, 6];
// all elements are greater than 4
const greaterFour = arr.every(num => num > 4);
console.log(greaterFour); // output: false
// all elements are less than 10
const lessTen = arr.every(num => num < 10);
console.log(lessTen); // output: true
```

8. sort()

This method used to arrange/sort array's item either ascending or descending order.

```
const arr = [1, 2, 3, 4, 5, 6];
const alpha = ['e', 'a', 'c', 'u', 'y'];
// sort in descending order
descOrder = arr.sort((a, b) => a > b ? -1 : 1);
console.log(descOrder); // output: [6, 5, 4, 3, 2, 1]
// sort in ascending order
ascOrder = alpha.sort((a, b) => a > b ? 1 : -1);
console.log(ascOrder); // output: ['a', 'c', 'e', 'u', 'y']
```

9. Array.from()

This change all thing that are array-like or iterable into true array especially when working with DOM, so that you can use other array methods like reduce, map, filter and so on.

```
const name = 'frugence';
const nameArray = Array.from(name);
console.log(name); // output: frugence
console.log(nameArray); // output: ['f', 'r', 'u', 'g', 'e', 'n', 'c', 'e']
```

working with DOM

```
// I assume that you have created unordered list of items in
our html file.
```

```
const lis = document.querySelectorAll('li');  
const lisArray =  
Array.from(document.querySelectorAll('li'));  
// is true array?  
console.log(Array.isArray(lis)); // output: false  
console.log(Array.isArray(lisArray)); // output: true
```

10. Array.of()

This create array from every arguments passed into it.

```
const nums = Array.of(1, 2, 3, 4, 5, 6);  
console.log(nums); // output: [1, 2, 3, 4, 5, 6]
```

Methods of string:

1. concat()

concat(v1, v2,...) Combines one or more strings (arguments v1, v2 etc) into the existing one and returns the combined string. Original string is not modified.

```
var message="Sam"  
var final=message.concat(" is a"," hopeless romantic.")  
//alerts "Sam is a hopeless romantic."  
alert(final)
```

2. indexOf()

indexOf(substr, [start]) Searches and (if found) returns the index number of the searched character or substring within the string. If not found, -1 is returned. "Start" is an optional argument specifying the position within string to begin the search. Default is 0.

```
var sentence="Hi, my name is Sam!"  
if (sentence.indexOf("Sam")!=-1)  
alert("Sam is in there!")
```

3. splice()

slice(start, [end]) Returns a substring of the string based on the “start” and “end” index arguments, NOT including the “end” index itself. “End” is optional, and if none is specified, the slice includes all characters from “start” to end of string.

```
var text="excellent"
text.slice(0,4) //returns "exce"
text.slice(2,4) //returns "ce"
```

4. substr()

substr(start, [length]) Returns the characters in a string beginning at “start” and through the specified number of characters, “length”. “Length” is optional, and if omitted, up to the end of the string is assumed.

```
var text="excellent"
text.substring(0,4) //returns "exce"
text.substring(2,4) //returns "ce"
```

5. substring()

substring(from, [to]) Returns the characters in a string between “from” and “to” indexes, NOT including “to” itself. “To” is optional, and if omitted, up to the end of the string is assumed.

```
var myString = 'javascript rox';
myString = myString.substring(0,10);
console.log(myString)
//output: javascript
```

Question 3: Ask the user if he is 21+ and log the value “can drink”, “cannot drink”.

Answer:

```
let age = prompt("enter your age:");
if(age>21)
console.log("can drink");
else
console.log("cannot drink");
```