

# Lecture 20

11/2/23

[53]

Date: !!  
P. No.

## \* In-built Objects

### (1) Math

→ Math.random()	{	→ generate Random
→ Math.round(1.6)		→ 2
→ Math.min(2, 4)		→ min(2)
→ Math.max(4, 2)		→ max(4)
→ Math.abs(2-6)		→ 4

### (2) String

Create String Object by 'new String()'

Ex:-

Object let firstName = new String('Priyansh')

→ Two types of String in JS

(i) Primitive

let name = 'priyansh'

(ii) Object

let name = new

String('Priyansh')

Convert Primitive to Object String  
by using . dot notation

name. ← object

(i) name.length → 8

(ii) name.includes('pri') // true

(iii) name.startsWith('pri') // true

(iv) name.endsWith('sh') // true

(v) String.trim() // remove white space  
from end & start



replace  
first

replace  
second

Date:  
P. No:

(vi) `string.replace('first', 'second')`

## \* Template Literals

let msg = ``This is a message`` } print as it is.  
backticks

## \* Date Object

let date = new Date();

let date2 = new Date('April 16 2003');

\* get & set in Date

Using set we can set → Date, Hours, milliseconds, Month, Minutes, Time etc

Ex → `date.setFullYear(2003)`

Now Get

`console.log(date.getFullYear());`

## \* Arrays

(i) Collection of Any Item.

(ii) Object of Reference type

Create → let arr = [1, 3, 5, 6, 9];

`console.log(arr[3]) // 6`

## \* Traversal

(i) END → `arr.push(5)`

(ii) Begin → `arr.unshift(1)`

(iii) Middle → `arr.splice(4, 0, 6)`  
index delete elements Idx



## \* Searching

bad  
Index

```
let number = [2, 4, 6, 8, 10]
Console.log(number.indexOf(4)) // 1
           //           //           (12) // -1
```

good  
includes

```
Console.log(number.includes(2)) // true
```

→ indexOf and includes not works with Objects.

```
let courses = [
  { No: 1, Name: 'Priyansh' }
]
```

```
Console.log(courses.includes({ No: 1, Name: 'Priyansh' }))
false [OIP]
```

## \* Callback Function

→ A callback function is a func passed into another function as an argument which is then invoked inside the outer function to complete some kind of action.

→ .find ( ) (method is used to Searching in Objects)

```
let course = courses.find(function(course) {
  // return course.Name = 'Priyansh'
})
```

Course.find (callback func / predicate func)







## \* Sliced

→ let arr = [1, 3, 5, 7, 9]  
 → let newArr = arr.slice() // 1 3 5 7 9  
 newArr = arr.slice(2, 4) // 5 7  
 include exclude

## \* Combine using Spread Operator

let Combined = [...first, ...second]

## \* ForEach Loop

combs

arrayName.forEach (callbackFn)  
 combined.forEach (function(number)  
 { console.log(number) })

## \* Joining Arrays

let arr = [20, 40, 60, 80]  
 let joined = arr.join(", ")  
 Output → 20, 40, 60, 80

## \* Split

let str = "Now This is a msg"  
 let splitted = str.split(" ")  
 Now | This | is | a | msg

## \* Other method

(i) arr.sort() } works only in  
 (ii) arr.reverse() } primitive.  
 etc



filter the elements from an array on a condition

Date: //  
P. No: //  
Pravinsh  
K. Patel

### \* Filtering Arrays

number.filter(callbackfn)

let number = [1, 2, -1, -4, 5]

let fl = number.filter(value => value > 0)

console.log(fl) // 1 2 5

### \* Mapping Arrays

Map each elements of array to something else.

let arr = [2, 4, 5]

let sq = arr.map(val => val \* val)

console.log(sq) // 4, 16, 25

### \* Mapping with Object

let arr = [4, 5, 6, 7, 8, 10]

let mapWithObject = arr.map(value => {  
 // return { Number: value }  
})

### \* Chaining

let chain = arr.filter(val => val > 1)  
 .map(val => val \* val)

### \* Reduce

let arr = [1, 2, 3, 4, 5]

let totalSum = arr.reduce((accumulator, currentValue) => {  
 accumulator + currentValue, 0  
})

console.log(totalSum) // 15