

In [7]: `//string => group of character`

```
//1)

var str1 = 'this is a string'
var str2 = "this is another string"

var str3 = `this sign is specially for
interpolation and multi
line`

var name = "suyash"
var greet = `Hello ${name} Good Morning`
console.log(greet)

var strObj1 = new String("suyash")
var strObj2 = new String("suyash")

console.log(strObj1.toString() == strObj2.toString())

var test1 = "suyash"
var test2 = "suyash"

console.log(test1 == test2)
```

Hello suyash Good Morning  
true  
true

In [10]:

```
//prop
var name = 'geeksdoor'

console.log(name.length)

for(var i=0; i<name.length; i++){
  console.log(name[i])
}

//from
//Array.isArray()
var arr = Array.from(name)
console.log(arr)
```

g  
g  
e  
e  
k  
s  
d  
o  
o  
r  
[  
  'g', 'e', 'e',  
  'k', 's', 'd',  
  'o', 'o', 'r'  
]

In [26]:

```
var arr1 = '123'
var arr2 = ['1', '2', '3']

//check equality
// var isSame = true
// arr1.forEach((e, ind)=>{
//   if(e != arr2[ind]){
//     isSame = false
//   }
// })

//type => array
//length equal or not
//every element must be same(with type)

isSame = Array.isArray(arr1) && Array.isArray(arr2) && arr1.length == arr2.length && arr1.every((el, ind)=>el === arr2[ind])

console.log(isSame)

var arr1 = [{name:"suyash", age:27}, {age:25, name:"yash"}]
var arr2 = [{age:27, name:"suyash"}, {name:"yash", age:25}]

//json => javascript object notation (string)
var json1 = JSON.stringify(arr1)
var json2 = JSON.stringify(arr2)
console.log("same ", json1 == json2)
console.log(typeof(arr1))
console.log(typeof(json1))

var arr1 = [1, 2, 3]
var arr2 = [3, 1, 2]
```

false  
same false  
object  
string

In [27]:

```
var str1 = "suyash"
var str2 = "suresh"

console.log(str1 > str2)
```

true

```
In [11]: var str1 = "surash"
var str2 = "suyash"

function strcmp(str1, str2){

    var ind = 0
    do{
        var diff = str1.charCodeAt(ind) - str2.charCodeAt(ind)
        ind++
        if(ind == str1.length-1){
            break
        }

    }while(diff == 0)

    return diff
}

console.log(strcmp(str1, str2))

-7
```

```
In [ ]: // var ind = 0
// var diff = str1.charCodeAt(ind) - str2.charCodeAt(ind)
// ind++

// while(diff == 0){
//     diff = str1.charCodeAt(ind) - str2.charCodeAt(ind)
//     ind++
//     if(ind == str1.length-1){
//         break
//     }
// }
```

```
In [13]: console.log(String.fromCharCode(97))

a
```

```
In [29]: //trim()
//trimStart()
//trimEnd()
var name = " suyash "
var trimmed = name.trim()

console.log(name.length)
console.log(trimmed.length)

7
6
```

```
In [30]: //indexOf //lastIndexOf
var str = "this is mern class"
console.log(str.indexOf("mern"))

8
```

```
In [31]: //split
var str = "this is mern class"
var splitted = str.split(" ")
console.log(splitted)

[ 'this', 'is', 'mern', 'class' ]
```

```
In [33]: //replace

var str = "Hello USER_NAME, hope you are doing well, USER_NAME"
var newStr = str.replace("USER_NAME", "suyash")
console.log(newStr)

Hello suyash, hope you are doing well, USER_NAME
```

```
In [36]: //replaceAll

var str = "Hello USER_NAME, hope you are doing well, USER_NAME"
var newStr = str.replaceAll("USER_NAME", "suyash")
console.log(newStr)

evalmachine.<anonymous>:4
var newStr = str.replaceAll("USER_NAME", "suyash")
                        ^

TypeError: str.replaceAll is not a function
    at evalmachine.<anonymous>:4:18
    at Script.runInThisContext (vm.js:120:18)
    at Object.runInThisContext (vm.js:309:38)
    at run ([eval]:1054:15)
    at onRunRequest ([eval]:888:18)
    at onMessage ([eval]:848:13)
    at process.emit (events.js:314:20)
    at emit (internal/child_process.js:876:12)
    at processTicksAndRejections (internal/process/task_queues.js:85:21)
```

```
In [38]: // join => implode
// split => explode

var queryStr = "?name=suyash&age=27&jd=developer"

//var req = {name:'suyash', age:27, jd:'developer'}

//replace ? with ''
//split with &
var req = queryStr
    .replace("?", "")
    .split("&")
    .reduce((acc, current)=>{
        var tmp = current.split("=")
        acc[tmp[0]] = tmp[1]
        return acc
    }, {})

console.log(req)

{ name: 'suyash', age: '27', jd: 'developer' }
```

```
In [105]: //number string
function genOtpDigit(len){
  var otp = ""
  for(var i=0; i<len; i++){
    otp += Math.round((Math.random()*1874))%10
  }
  return otp
}
console.log(genOtpDigit(6))
```

362672

```
In [132]: //cptcha
function genCapta(len){
  var str = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
  var otp = ""
  for(var i=0; i<len; i++){
    otp += str[Math.round((Math.random()*1874))%str.length]
  }
  return otp
}

console.log(genCapta(5))
```

6bjDL

```
In [56]: //Math.random()  => 0-1

// % => 0-9 %10

//0-5    % 6

Math.round((Math.random()*1874))%10
```

Out[56]: 2

```
In [4]: var a = 10
var b = 20

//30

var sum = a - (-b)
console.log(sum)
```

30

```
In [19]: //startsWith()
//endsWith()

var str = "kumarsuyash94@gmail.com"

console.log(str.startsWith("kumar"))

var GMAIL = "@gmail.com"
var OUTLOOK = "@outlook.com"
var YAHOO_IN = "yahoo.com"

var isGmail = str.endsWith(GMAIL)
console.log(isGmail)
```

true
true

```
In [21]: //includes
str.includes('@')

//toUpperCase() toLowerCase()
var str = "kumarsuyash94@gmail.com"
str.toUpperCase()
str.toLowerCase()

//pwd => enc
//lower

//78% => mobile
//82% => android

//SQL Match

// "Kumarsuyash9@gmail.com"

// "kumarsuyash9@gmail.com"

// trim()
// toLowerCase() => store
```

Out[21]: 'KUMARSUYASH94@GMAIL.COM'

```
In [27]: //regex(pattern) => compiler design
var email = "abc"
const re = /^[^<>()[\]\.\,\;\s@"]+(\.[^<>()[\]\.\,\;\s@"]+)*|("[.+"")@(\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\)|((\[[a-zA-Z\0-9]+\.\.)+[a-zA-Z]{2,}))$)/;
console.log(re.test(String(email).toLowerCase()))

evalmachine.<anonymous>:1
//regex(pattern) => compiler design
^

SyntaxError: Identifier 're' has already been declared
    at evalmachine.<anonymous>:1:1
    at Script.runInThisContext (vm.js:120:18)
    at Object.runInThisContext (vm.js:309:38)
    at run ([eval]:1054:15)
    at onRunRequest ([eval]:888:18)
    at onMessage ([eval]:848:13)
    at process.emit (events.js:314:20)
    at emit (internal/child_process.js:876:12)
    at processTicksAndRejections (internal/process/task_queues.js:85:21)
```

```
In [28]: var patternStr = `
    if(){

    }else{

    }
`

obj = {}

var match = ['(', ')', '{', '}']

for(var i = 0; i<patternStr.length; i++){
    var chr = patternStr[i]
    if(match.includes(chr)){
        if(chr in obj){
            obj[chr]++
        }else{
            obj[chr] = 1
        }
    }
}

console.log(obj)

//stack

{ '(': 1, ')': 1, '{': 2, '}': 2 }
```

```
In [37]: //padStart
//padEnd

//INV00001
//INV00010
//INV00100

var str = "1000000"
console.log("INV"+str.padStart(5, "0"))

var userStroy = "HI here"

console.log(userStroy.padEnd(100, "."))

INV10000
HI here.....
```

Out[37]: '\*\*\*\*\*'

```
In [ ]: //repeat()
"★".repeat(10)
```

```
In [39]: function padStart(str, maxLen, padChar){
    var diff = maxLen-str.length
    if(diff > 0){
        return padChar.repeat(diff)+str
    }else{
        return str
    }
}

console.log(padStart("10", 5, "0"))

00010
```

```
In [41]: // sub()
// substr()
var name = "suyahs"
//substring(start)
//substring(start, upto)
console.log(name.substring(1, 3))

uy
```

```
In [ ]: //math
//date => moment.js
//oops + ES6
//dsa

//window object
//document
//jquery

//weired part
//react

//event loop
```