

16 Jan 25

Roj Ka Kaom Roj

Date:
Page:

Basic JS API's

- # to get length of string
- (`str.length`)

```
const str = "Aayam Sharma";
console.log(str.length)
```

Result

12 C i g. S p a c e . a d d e d)

11 " not

- # (`findIndex()`)

It tells where the particular thing is present (target).

```
str.indexOf(target);
```

- * If the value is present it will give value if not.
- * The it will return -1

For last

~~String.prototype.lastIndexOf(target);~~

Function findIndexof(sto, target) {

```
    console.log("Original string : " sto);
    " " ("Index", sto.lastIndexOf(target));
```

~~Actual~~ ↴

FindIndexof("Hello world", "world");

Get Slice

To find things in between
like give the ~~Index of~~ ^{slice} 1 and Index

eg

function getSlice(sto, start, end) {

Console.log("Original String:", sto);

```
" " ("AfterSlice:", sto.slice(start, end));
```

↳

~~Actual~~

Short

let ans = "Aryan Sharma".slice(0, 5)

console.log(ans)

(Ans: "Aryan")

Replace

const str = "Hello World";

const str = "Hello world";

console.log(str.replace("world", "broccoli"))
("Hello JavaScript")

3

Result, Ans: Hello JavaScript

Hello JavaScript

Split

const value = "Hi my name is Aryan";

const words = value.split(" ");

const words = value.split(" ");

console.log(words);

Result

```
[ 'hi', 'my', 'name', 'is', 'Aryan' ]
```

or

```
const value = "hi, my, name, is, Aryan"
```

```
const words = value.split(", ");
```

```
console.log(words);
```

Result

```
' hi my name is Aryan'
```

it will split words based on,

Trim

```
const value = "      Aryan      sharma      "
```

```
console.log(value.trim())
```

it will trim space

Result

```
"Aryan-Sharma"
```

#

upper case and lower case

const value = "Aryan Sharma";

console.log(value.toUpperCase());

lower case

const value = "Aryan Sharma";

console.log(value.toLowerCase());

uppercase Result

ARYAN SHARMA

lower case Result

Aryan Sharma

#

Parse Int

(Passing a number into Integer)

console.log(parseInt("42"));

console.log(parseInt("42px"));

console.log(parseInt("3.14"));

Results

42

42

3

Pause Float

→ Can have a decimal number
→ ~~fractional~~ ~~decimal~~ ~~fraction~~

```
console.log(ParseFloat("42"));
```

```
"").ParseFloat("42paaaax"));
```

```
console.log (ParseFloat ("3.14"));
```

42

42

3·14

Away

```
const initialArray = [1, 2, 3];
```

initialA array.push(2);

`console.log (initial Array)`

Rexel

push new number in array

Replace push with pop it will

pop number

Replace

Pop with

.shift();

it will shift array

To put value in front

value you want.

.unshift(0); to add in

front

Result

[0, 1, 2, 3]

Merge Two Array

const first =

const initialArray = [1, 2, 3]

" Second " = [4, 5, 6, 7, 8]

console.log(initialArray + concat(second))

Output: [1, 2, 3, 4, 5, 6, 7, 8]

and now we have it

Result



[1, 2, 3, 4, 5, 6]

or

Additional after 1 and 2nd array

const finalArray = initialArray.concat
([SecondArray]);

console.log(finalArray);

For each

Call each no's of Array 1 by 1

like 1st 2nd 3rd etc.

const initialArray = [1, 2, 3];

function logThing(str);

console.log(str);

const initialArray = numbers;

initialArray.forEach(logthing);

Array()); result: 1, 2, 3

1

2

3

1, 2, 3

Map, Filter, Find and Sort

Task 1: Class

Classes

Classes Animal {

 constructor(name) {

 this.name = name;

}

 speak() {

 console.log(` \${this.name}`);

 makeNoise();

}

}

2

Class Animal {

Constructor (name, legCount, speak) {

 this.name = name;

 this.legCount = legCount;

 this.speak = speak;

0 3
3

x Let dog = {

Let dog = new Animal("dog", 4, "bhow bhow")

#

Date

const currentDate = new Date();

console.log(currentDate.getMonth());

Return

→ will return current month

can add

* currentDate.getDate();

* currentDate.getFullYear();

* currentDate.getMonth();

for more codes check the code on netion

JSON

JavaScript Object Notation

```
const users = { "name": "Aryan", "ge":  
 21, "gender": male };
```

Parse Method

```
const user = JSON.parse(users)  
console.log(user["gender"]);
```

Result

Male

Result

Male

gender

3

user

Con

Co

#

No

Stringify Method

```
const user = {
```

name: "harkirat",

gender: "male"

Stringify
Method

}

#

F

```
const finalString = JSON.stringify(user)
```

```
console.log(finalString)
```

J.SON

JavaScript Object Notation

```
const users = { "name": "Aryan", "age":  
    21, "gender": male }
```

Parse Method

```
const user = JSON.parse(users)  
console.log(user["gender"]);
```

Result

Male

Stringify Method

```
const user = {  
    name: "Harshit",  
    gender: "male"}  
Stringify  
method
```

```
const finalString = JSON.stringify(user)  
console.log(finalString)
```

Result ~~introduction to JS, Encapsulation, objects~~

Male

Gender: "male"

3

user["name"]

const finalString = JSON.stringify(user)

console.log(finalString["name"])

Mathematical functions:

console.log(Math.random())

• 0.0008(8)

• ceil

for more functions check the list
present in Notion notes

Objects

Loops, functions and callbacks in JS

Loops

Dumb way

```
let ans = 1 + 2 + 3 - - - -
```

```
- - 50;
```

```
console.log(ans);
```

Best way

```
let ans = 0
```

```
for (let i=1; i<50; i++) {
```

```
ans = ans + i;
```

3

step 10

```
console.log(ans);
```

In Best way

Ans = 0
then

let ans = 0; will run

then

let i = 1; will run

then

ans = ans + i; will run

Loop will run till $i \leq 50$; will

then

when loop will cross 50, & loop will be break and you will get output.

Functions

Set of statements that perform

a task or calculate a value

it should take some input and return an output where there is some obvious relationship b/w the input and the output.

Date: _____
Page: _____

function findSum(n) {

Name Argument

```
let ans = 0;  
for (let i=1; i<n; i++) {  
    ans = ans + i;  
}  
return ans;
```

function
body:
— w function
return

Another eg

```
function sum(a,b) {  
    return a+b;
```

How to call a function

```
function findSum(n) {
```

```
let ans = 0;  
for (let i=1; i<n; i++) {
```

ans = ans + i

3

function ans;

3.

let ans = fundsum(100)
console.log(ans);

Calling

function

Callback Functions