

Javascript Basics.

08 August 2022 20:54

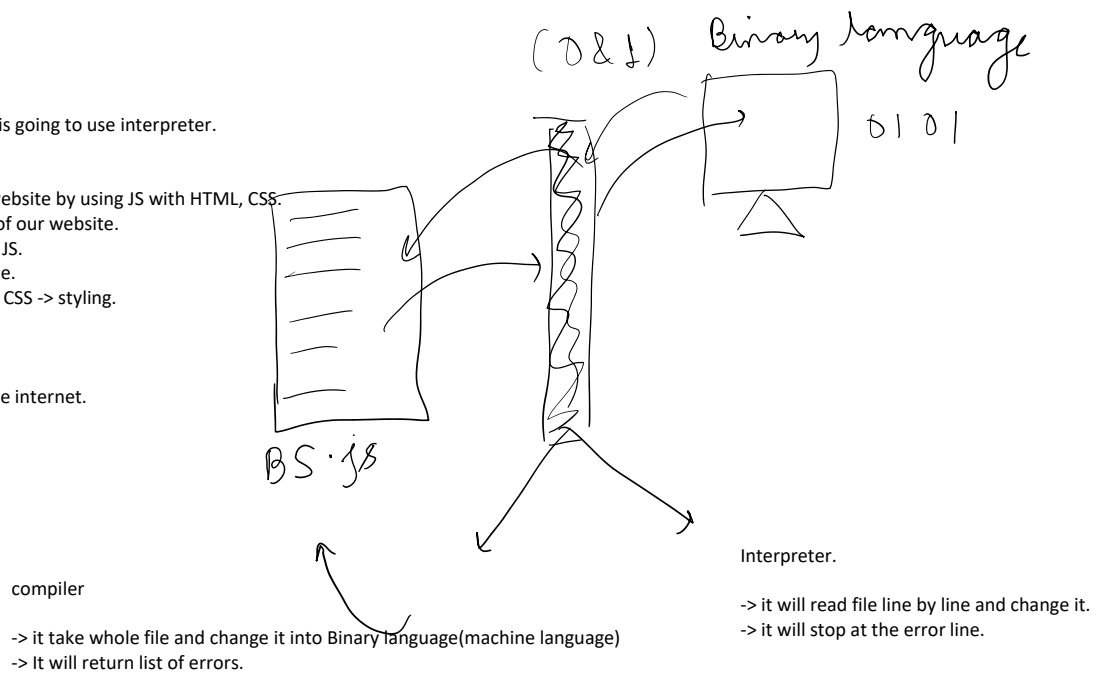
1. Scripting language. => it is going to use interpreter.

Why JS?

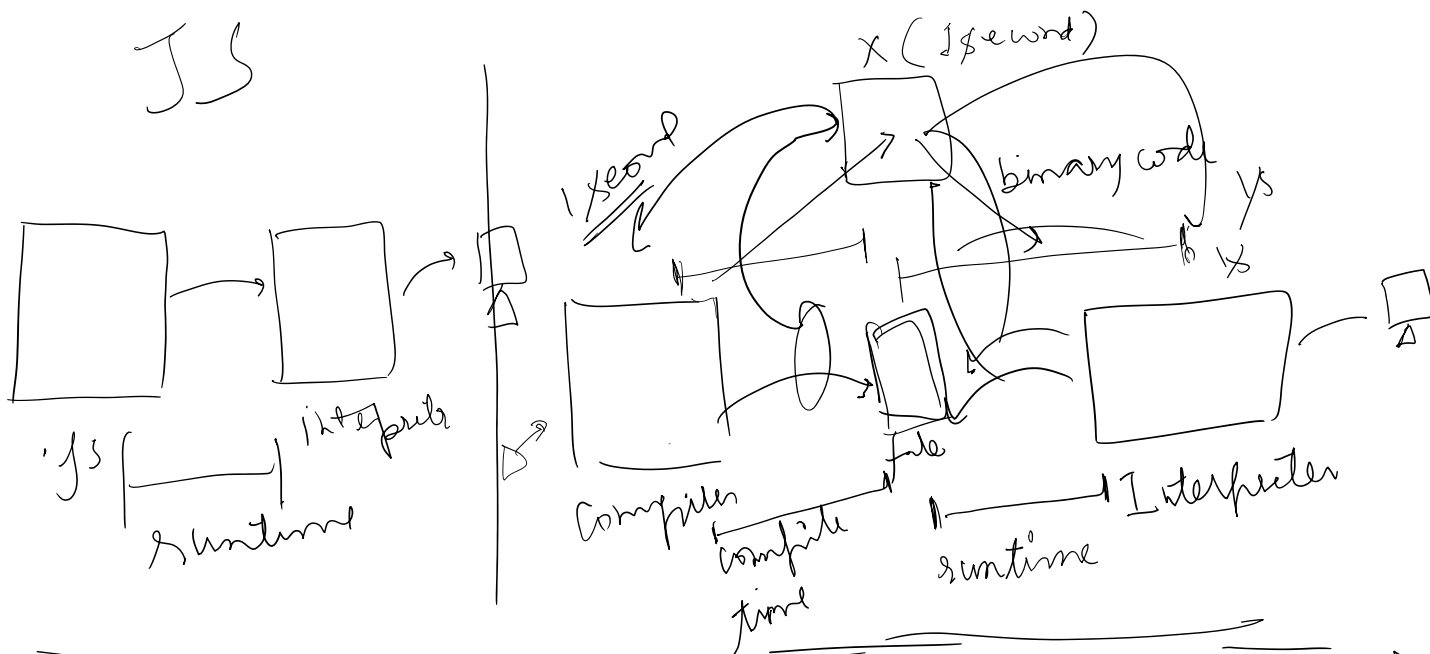
1. We can make dynamic website by using JS with HTML, CSS.
2. It tells us the behaviour of our website.
3. Browser can understand JS.
4. Integrate with HTML code.

HTML -> backbone, JS -> brain, CSS -> styling.

- a. Easy to learn.
- b. Lots of the content on the internet.



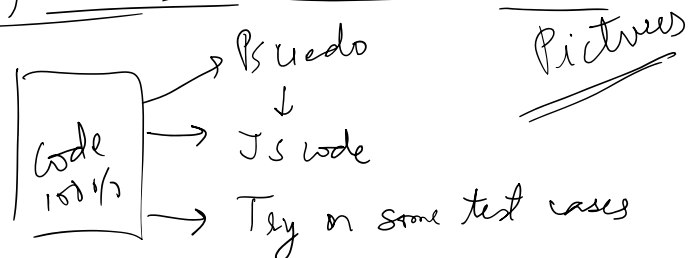
Time complexity for interpreter to interpreting n lines of code is $O(n)$ => linear time.
Time complexity for compiler to compiling n lines of code is $O(1)$ => constant



Why are you here?

1. Learn to code and job.
2. Mentorship with learning,
3. 4-5 months.

1. 1 month -> pseudo code -> Actual JS code.



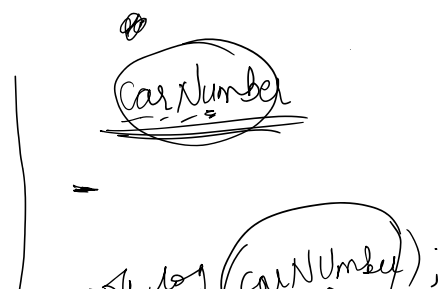
Javascript.

Variables.

1. Every variable name should be start from Alphabets(lower case, upper case), underscore(_), dollar(\$).
2. Javascript is a Case-sensitive language and we have to take care of it while defining variables.

Print.

Console.log()



defining variables.
Print.
Console.log()

console.log(calNumber);
1
✓ = air ✓ \$air
✓ air

09-08-2022 Basics.

09 August 2022 20:42

Attendance link:- https://docs.google.com/forms/d/e/1FAIpQLScleXy95A54S3ggvhdozwwkIURXPUIAREf162cgrVZWJZAb1w/viewform?usp=sf_link

1. Keyword:- Some reserved words. e.g var, let, const, function, true, false etc.
2. Declaration/Declare:- To define.
3. Initialisation/Initialise :- assigning the value.
4. Scope:- Area that is covered.
 - a. Global scope
 - b. Function scope
 - c. Block Scope:-

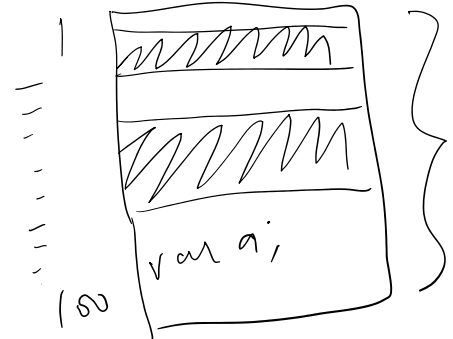
Var xyz;

var xyz = 124;

Block:- blocks are in {}.

var priya = "Singh";

var xyz;
xyz = 124;



Variables:-

Naming conventions:-

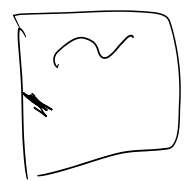
1. We can't use keywords.
2. Start from alphabets(lower and upper case), _ \$.
3. Case sensitive. Xyz and xyz and xyZ are different.
4. e.gs :- aman, Aman, _aman, _123, \$aman

i) Akash
ii) Ankaash
iii) Akaash

Comments:-

1. // to comment a line.
2. /* */ to comment a multiple lines.

priya
priyaa



var 1234;

Data types:-

Primitive:-

Data types:-

Primitive:-

1. Number:- Integers, Decimals
2. String:- Words, Sentences.
 - i. You must have to use either "" or ' ' to define a string.
 - ii. When you have to define a paragraph (multiple lines), we must have to use `` in that case.
3. Boolean:- true/false.
4. Null:- intentional empty value.
5. Object:- {},
 - i. It must be wrap in {}
 - ii. The data looks like Key:Value pair separated by commas.
 - iii. Keys are already in string form, its predefined.
 - iv. Example:-

```
var first_last_name_object = {  
  "aman": "Dokania",  
  "mohit" : "Chopra",  
  "xyz": "uip"  
}
```
 - v. For getting every keys:- `Object.keys(object variable name);`
 - vi. For getting every values:- `Object.values(object variable name);`
6. Undefined:- un-intentional empty value.

Non-primitive:-

1. Start from undefined, diff b/w undefined and null.
2. Var, let, const

val 

2 balls -> BR, RB
0 balls -> 1

10-08-2022 variables scopes, undefined

10 August 2022 18:49

1. A+E Group -

https://docs.google.com/forms/d/e/1FAIpQLSdMhEwu8hioMRIEyX8dP40vXNn-r3Q7gu-1427Uf3bPqI5CcA/viewform?usp=sf_link

<https://replit.com/teams/join/zdemnyjdgswcjsdjearuzchzosijobw-ac102-batch>

Keyword can be used for JS rules.

JS Rule 1:- How to declare a variable?

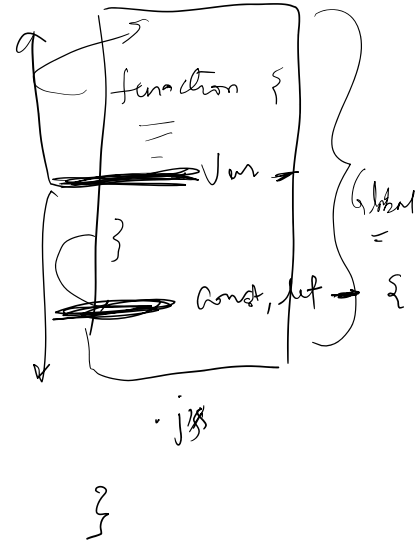
Ans:- var/let/const <variable_name>;

Why Function?

Why Variable?

Keyword use to declare Variables:-

- i. var:
 - a. It has Global or functional scope.
 - b. We can re-declare the variables.
 - c. We can re-assign/update/re-initialise the variable value.
 - d. If we use var variable before declaring, it gives us the value "undefined".
- ii. let:
 - a. It has a block scope.
 - b. We cannot re-declare it.
 - c. We can re-assign/update/re-initialise the variable value.
- iii. const:
 - a. It has a block scope.
 - b. We cannot re-declare it.
 - c. We cannot change the variable value.



1. Object.

11-08-2022 typeof, strong weak, static dynamic, operators, basic class ques.

11 August 2022 21:01

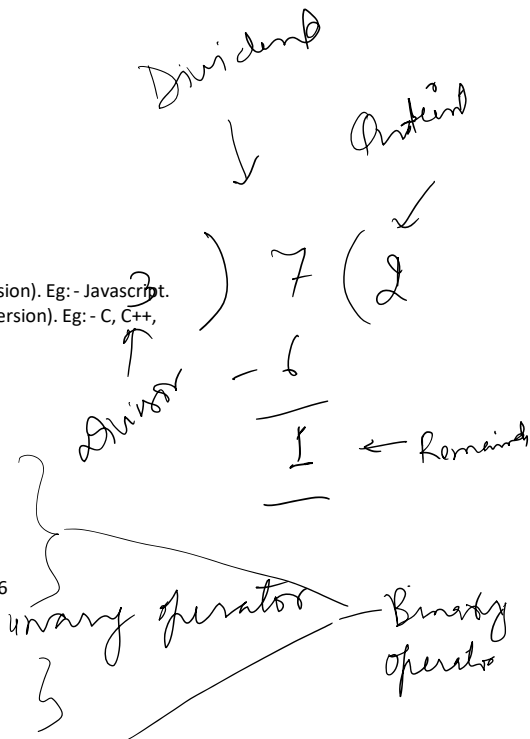
1. typeof: function that gives us Type of variable/value.
 - a. typeof(23) = number
 - b. typeof("900") = string

Static typed language: languages which finds their data type at the compile time. e.g -> C, C++, Java
 Dynamic typed languages: languages which finds their data type at the runtime. e.g -> JavaScript

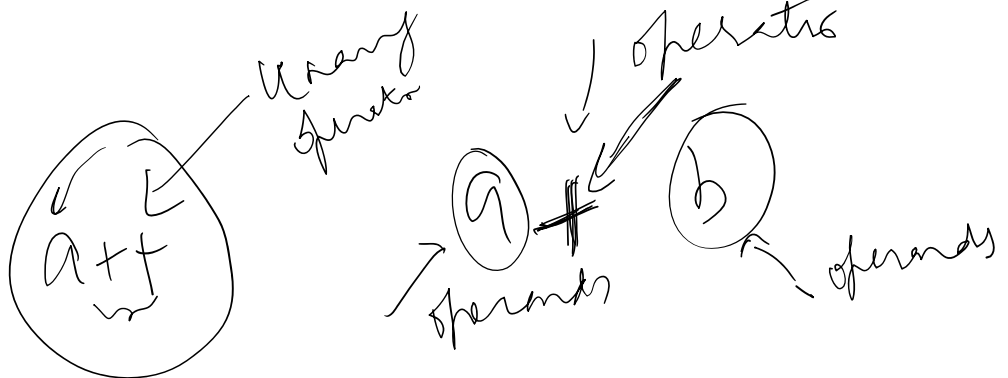
Weak typed languages:- you can able to do operation on different Data types variables (implicit type conversion). Eg:- JavaScript.
 Strong typed languages:- you can't able to do operation on different Data types variables (explicit type conversion). Eg:- C, C++, python.

Operator:

1. Arithmetic operators:-
 - a. Addition (+)
 - b. Subtraction (-)
 - c. Multiplication (*)
 - d. Division (/)
 - e. Modulus(%) :- a%b => remainder when a is divided by b. e.g 7%3 = 1;
 - f. Exponent(**) :- power. a**b => a to the power b. e.g 2**5 = 32; 2**10 = 1024; 2**20 = 1048576
 - g. Increment (++) :- a++ => increase a by 1. e.g var x = 2; x++ => 3.
 - h. Decrement (--) :- a-- => decrease a by 1. e.g var x = 2; x-- => 1.

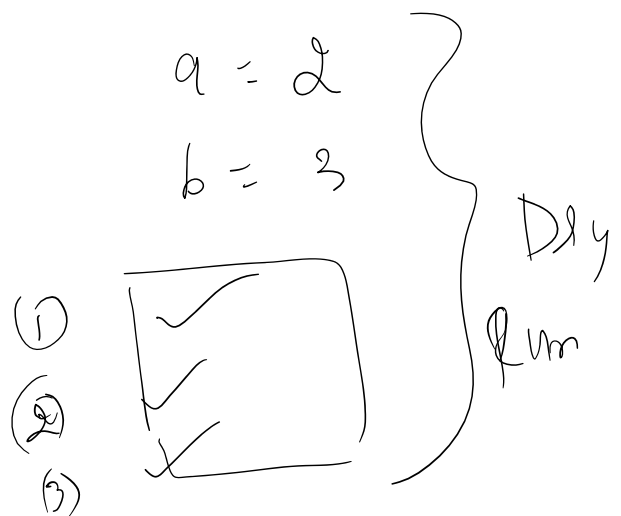


+ looks good, - is not working.



English // =

Solution ??

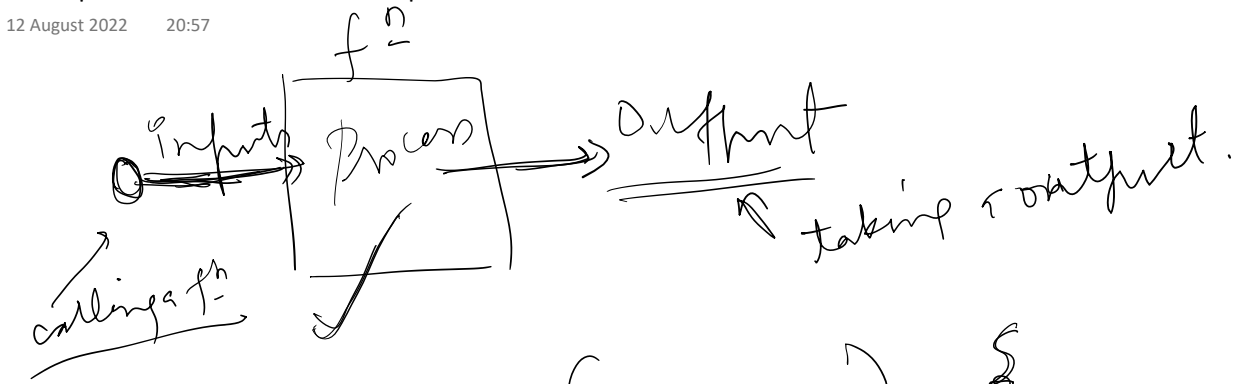


ftw
/ ?!

(2) 'X' /
(3)

12-08-2022 Replit, function, use of ` , Assignment operator, Comparison / Relational operator

12 August 2022 20:57



function f^n Name () {
 // process
 return output;
 }
 parameter, arguments
 separated by comma

Square 2
 Average = $\frac{\text{'N' items} \rightarrow \text{prices}}{N} = \frac{\text{Sum of every value}}{\text{Total no. of items}}$
 $\rightarrow \frac{P_1 + P_2 + P_3 + \dots + P_N}{N}$

- Step 1: calculate the sum of all
 2: divide it by total count.
 3: return output.

$$a = b - a$$

$$a = b$$

$$a = @b$$

Assignment operators:- operators those are used to assign a value to a variable. LHS <- RHS

i. = : e.g $a = 2 + 5;$

variable. LHS <- RHS

- i. = : e.g $a = 2 + 5$;
- ii. += : $a += b \Rightarrow a = a + b$;
- iii. -= : $a -= b \Rightarrow a = a - b$;
- iv. *= : $a *= b \Rightarrow a = a * b$;
- v. /= : $a /= b \Rightarrow a = a / b$;
- vi. %= : $a \% b \Rightarrow a = a \% b$;
- vii. **= : $a ** b \Rightarrow a = a ** b$;

$$a = \frac{a}{b}$$

$$\begin{aligned} 4 - 2 &= 2 \\ 2 - 4 &= -2 \end{aligned}$$

$$a = a + b$$

2 1, 1

$$a \neq b$$

↑
Assignment.

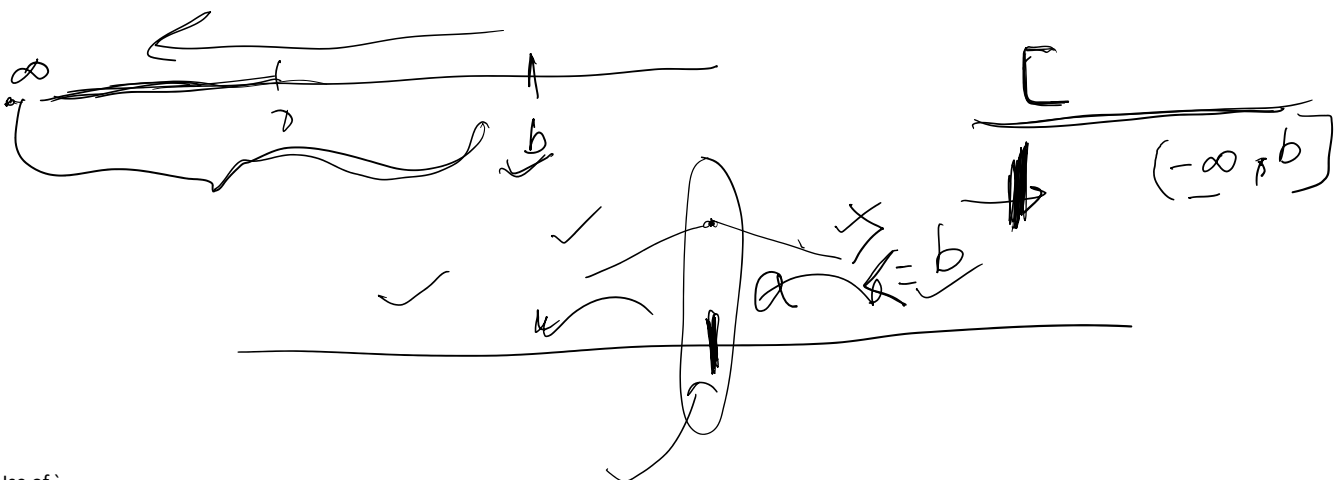
$$y * 5$$
$$x = y * 5$$

Comparison / Relational operator:- these operators are going to check or compare the two things. Return Boolean (True/False)(1/0)

- i. == : this will check, two things are equal **by value**.
- ii. === : this will check, two things are equal **by value and also in data type**.
- iii. ' > ' : greater than
- iv. < : less than
- v. >= : greater than or equal to
- vi. <= : less than or equal to.
- vii. != : not equal to **by value**.
- viii. !== : not equal to **by value and data type**.
- ix. ? : Ternary operator,
a. <comparator/ logical statement> ? " return if true " : " return if false";

(1/0)

}



Use of `

1. For writing the multiple lines.
2. For using variables in the string.
 - a. `\${name} is my name.`

15-08-2022 ?, logical operators, Numbers (floating point, 0.1 + 0.2).
 toString, toFixed, toPrecision. Number(), parseInt, parseFloat, abs,
ceil, floor, round, max, min, pow, random, sqrt, cbrt

15 August 2022 18:59

Logical Operators: these operators always work on binary (0/1)
 (true/false)

i) NOT (!) -> inverts the input.

Input	Output
1	0
0	1

ii) AND(&) -> (for learning, it works as multiplication)

Input1	Input2	Output
0	0	0
0	1	0
1	0	0
1	1	1

iii) OR (||) ->

(for learning, it works as Addition)

Input1	Input2	Output
0	0	0
0	1	1
1	0	1
1	1	1

$[-2^{36}, 2^{36}-1]$

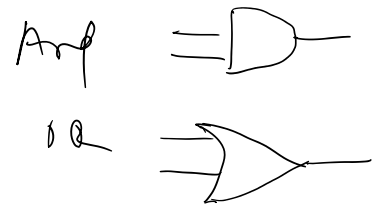
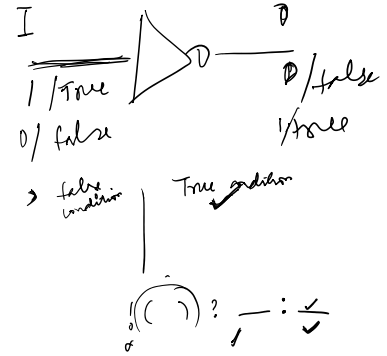
$$a) \quad b \quad (a) \quad \uparrow$$

$$\overline{a} \leftarrow$$

$$b = a * q + r$$

$$\text{Integer} = \left\lfloor \frac{b-r}{a} \right\rfloor = q$$

$$\text{decimal} = r/a$$



Numbers
 why floating
 point error
 comes in JS.

Binary system
 Base 2

1 2

$\frac{1}{1} = 1.0$

$\frac{1}{2} = 0.5$

Decimal system
 Base 10

$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

10

1 2 5 10

$\frac{1}{1} = 1$

$\frac{1}{2} = 0.5$

$\frac{1}{5} = 0.2$

$\frac{1}{10} = 0.1$

Base 2

153

$1 \times 2^7 + 5 \times 2^6 + 3 \times 2^5$

$\frac{1}{3} = 0.\overline{3}$

$\frac{1}{7} = 0.\overline{142857}$

$\frac{1}{9} = 0.\overline{111111}$

$\frac{1}{4} = 0.25$

$\frac{1}{8} = 0.125$

11

$1 \times 2^1 + 1 \times 2^0$

$2 + 1$

$= 3$

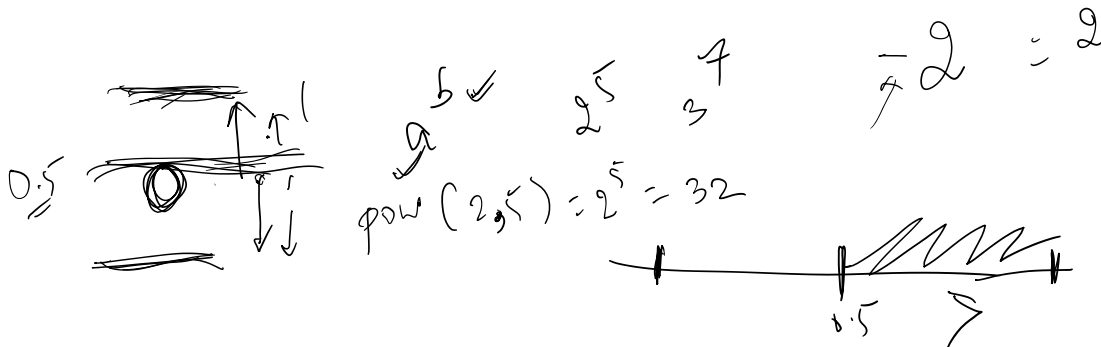
1. toString() => changing the given value into string. e.g var a = true; a = a.toString();
2. toFixed() => return a string, with the number written with a specified number of decimals.
 - a. 9.789.toFixed(2) => 9.79
 - b. 9.789.toFixed(0) => 10
 - c. 9.789.toFixed(4) => 9.7890
3. toPrecision() => it basically returns the exact number of digits that you want.
 - a. 9.7898.toPrecision(2) => 9.8

• — — — — —
 ↑
 round off

Math.PI

1. abs :- it returns absolute value. e.g it return the positive part of the negative values. $\text{abs}(-2) = 2$.
2. ceil :- 9.4 -> 10
3. floor :- 9.8 -> 9
4. round :- 9.8 -> 10, 9.4 -> 9
5. max:- max(a1,a2,a3,a4)
6. min:- min(a1,a2,a3,a4)
7. pow:- pow(a, b)
8. random:- it is returning any random value between 0 and 1.
9. sqrt:- square root
10. cbrt: cube root.

magnitude of
absolute value
↓
sign, direction
2



$$6 - 7$$

$$[6, 7] = \frac{\text{Math.random()}}{[0, 1] + 6} + 6 \Rightarrow [6, 7]$$

0.5, 0.8

19-08-2022 String

String(true) = "true"

Props:-

1. length => it gives length of the string.

Methods:-

Properties ← props

method ← functionality
given by that
particular thing

A = 5 B = 9

A = A + B // 9 + 5 = 14

B = A - B // 14 - 9 = 5

A = A - B // 14 - 5 = 9

↓ word ↑

0	1	2
2	4	7

toString() ← string

String ← Class constructor from
f.h. == class name

0	1	2	3
2	3	4	7

= 4
[0, len-1]

arr[1] = 3

str = air.com pus
str.substring(3, 7)

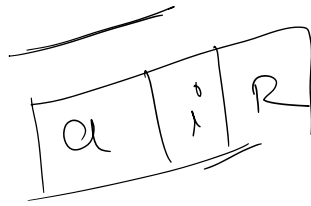
3, 7-3

Non-negative numbers => positive number including 0,
positive numbers

Non-positive numbers => negative number including 0,
Negative numbers.

Increasing, non-increasing, decreasing, non-decreasing.

air
1.1.0



```

var str = String(2477777444);

// console.log(str, typeof(str));

// console.log(str.length);

// console.log(str[0], str[1], str[2], str[3]);

// var firstDigit = Number(str[0]) // str.at(0);
// var secondDigit = Number(str[1]) // str.at(0);

// console.log(firstDigit + secondDigit);

// at method. -> we can also put negative indexes.
// params:- index
// console.log(str.at(3));

// charAt method -> positive indexes.
// params:- index
// console.log(str.charAt(-1));

// concat means merge forwardly. 2 + 4 = 24.
// params:- strings separated by ,
// console.log(str.concat("89", "98"))

// indexOf method => the index of the first occurrence of char. otherwise -1.
// params:- param1 => value that you are looking for.
// param2 = Index from which searching is going to be start.
// return first index (number) or -1.
// default value of this param2 is 0.

str = "abbabb";
// console.log(str.indexOf('a', 4));

// lastIndexOf
// params:- param1 => value that you are looking for.
// param2 = Index from which searching is going to be start.
// default value of this param2 is 0.
// return last index (number) or -1.
// console.log(str.lastIndexOf('a'));

str = "air campus is here";
// includes method
// params:- param1 => value that you are looking for.
// param2 = Index from which searching is going to be start.
// default value of this param2 is 0.
// return boolean
// console.log(str.includes("abbb", 4));

// toLowerCase -> change string into lower case
// toUpperCase -> change string into upper case

// substring -> return the sub string
// param 1 => starting index of substring
// param 2 => ending index of substr, default value is length of string.
// console.log(str.substring(2,4))

// slice -> extracts a section of a string and returns a new string.
// console.log(str.slice(1, 3));
// console.log(str);

// split -> method takes a separator and returns an array of the substrings
// separated by that given separator.
// param1 => Separator.
// param2 => limit of the substring that you want to be returned
// it should be a non-negative number.
// str = "air campus is here";
// console.log(str.split(", 90));

```

air campus
 0 1 2 3 4 5 6 7 8 9
 ↑ ↓ ↑
 2 3 4
 len = 9 3 + 4 = 7

at, range of valid
 indices = $[-\text{length}, \text{length} - 1]$;

air
 A little word
 Regera = Regular
 Expressions

regen = $[a-z][a-z][a-z][a-z]$

```
// startsWith => Is String starts from the passed substring or not.  
// returns boolean.  
// str = "air campus is here"  
// console.log(str.startsWith("cam"))  
  
// endsWith => Is String ends with the passed substring or not.  
// returns boolean.  
// str = "air campus is here"  
// console.log(str.endsWith("e"))  
  
// trim  
// trimStart  
// trimEnd
```

20-08-2022 Conditional statements.

20 August 2022 21:18

1. If- else if - else:-

```
If ( condition is correct ) {
```

```
    } else {
```

```
    }
```

2. Switch-case

a. switch (expression):

```
    case value:
```

```
        // code
```

```
        break;
```

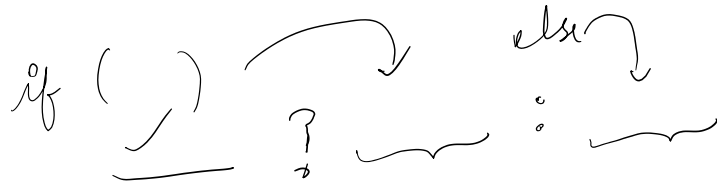
```
    case value:
```

```
        // code
```

```
        break;
```

```
    default:
```

3. default is not necessary



1. Break;

2. default

Pallindrome -> something that is equal when you read it from any side

121

121

amma

amma

20 August 2022 21:37

HOW TO TAKE INPUT FROM USER. loop