Javascript Basics.

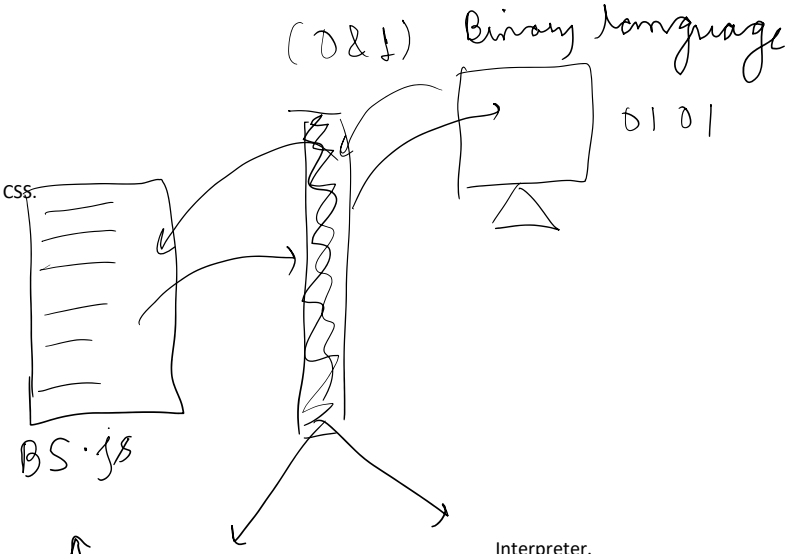
08 August 2022 20:54

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.



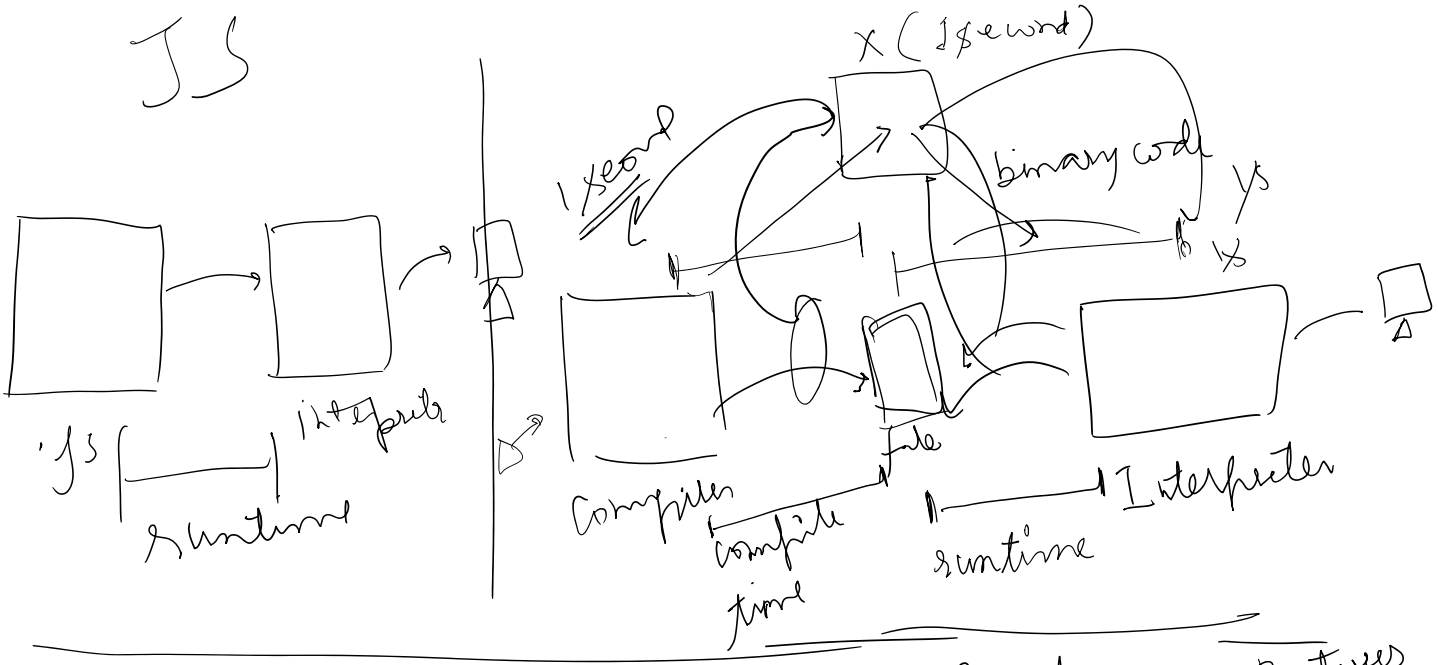
compiler

-> it take whole file and change it into Binary language(machine language)
-> It will return list of errors.

Interpreter.

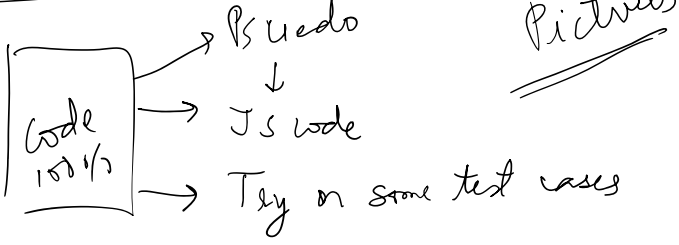
-> it will read file line by line and change it.
-> it will stop at the error line.

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.

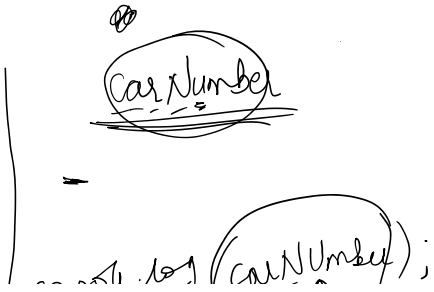
1. 1 month -> pseudo code -> Actual JS code.



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.

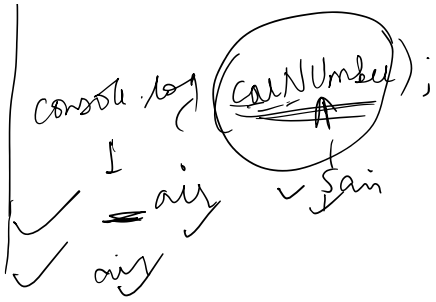
Print.
Console.log()



defining variables.

Print.

Console.log()

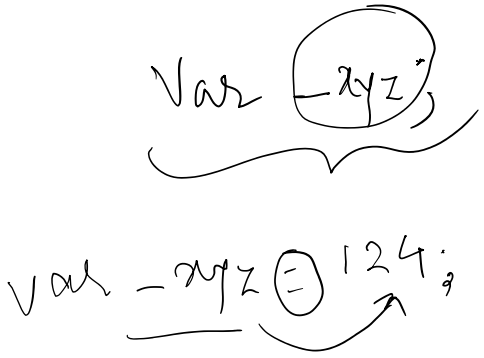


09-08-2022 Basics.

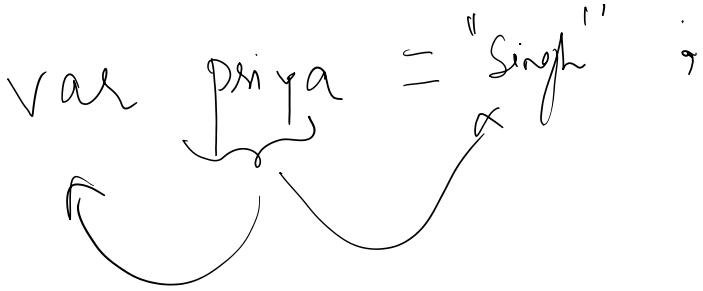
09 August 2022 20:42

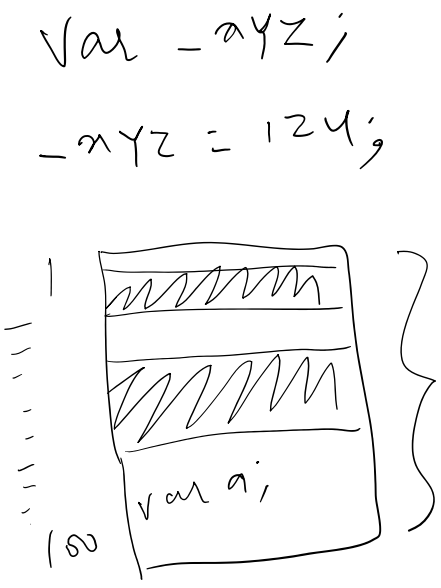
Attendance link:- https://docs.google.com/forms/d/e/1FAIpQLScIeXy95A54S3ggvhdozwkIURXPUIAREfI6_2cgrVZWJZAblw/viewform?usp=sf_link

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



Block:- blocks are in {}.

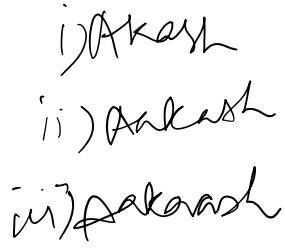




Variables:-

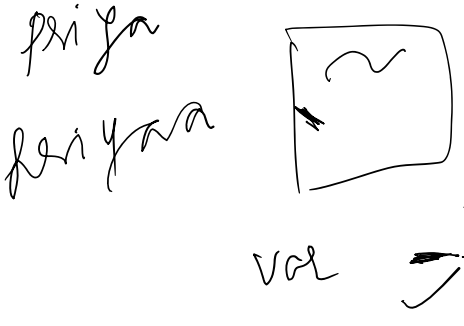
Naming conventions:-

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



Comments:-

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



Data types:-

Primitive:-

Data types:-

Primitive:-

1. Number:- Intergers, 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:-

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

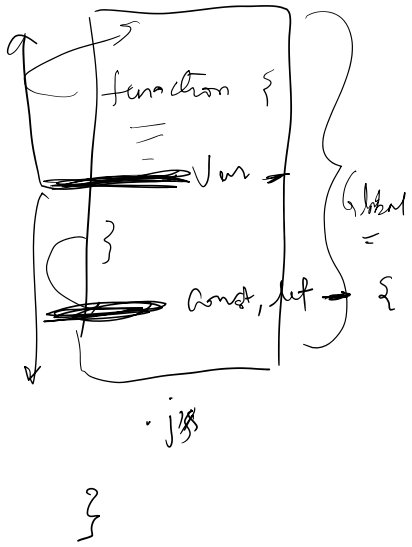
1. Start from undefined, diff b/w undefined and null.
2. Var, let, const

10-08-2022 variables scopes, undefined

10 August 2022 18:49

1. A+E Group - https://docs.google.com/forms/d/e/1FAIpQLSdMhEwu8hioMRlEyX8dP40vXNn-r3Q7gu-1427Uf3bPql5CcA/viewform?usp=sf_link

https://replit.com/teams/join/zdemnyjdgswcjsdjrearuzchzosijobw-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. le
   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.

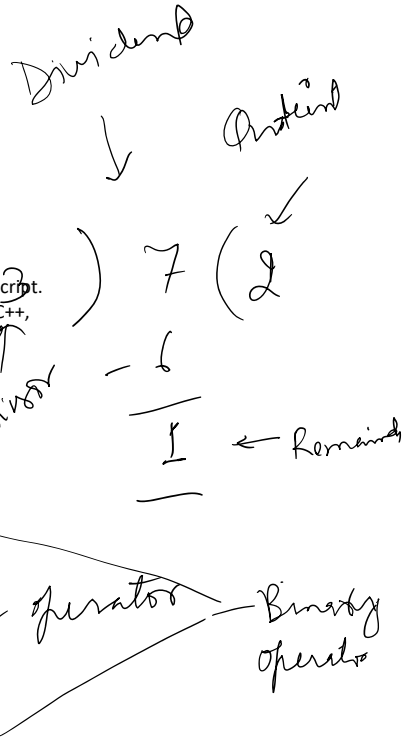
11-08-2022 typeof, strong weak, static dynamic, operators, basic class ques.

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

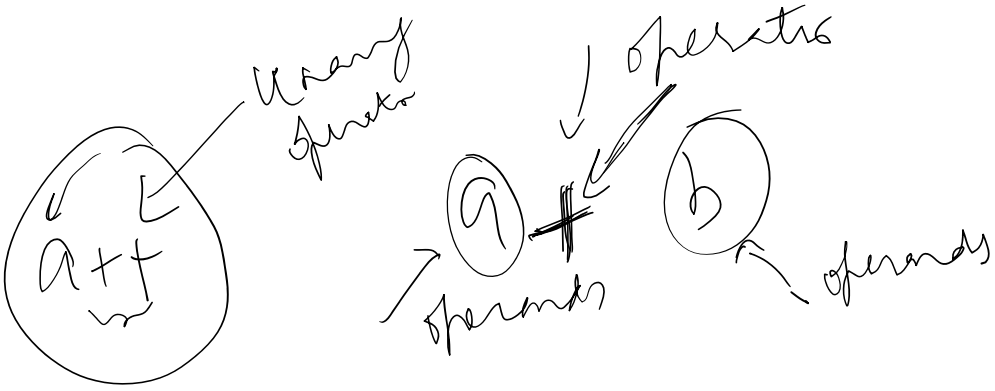
Weak typed languages:- you can able to do operation on different Data types variables (implicit type conversion). Eg: - Javascript.
Strong typed languages:- you can't able to do operation on different Data types variables (explicit type conversion). Eg: - C, C++, python.

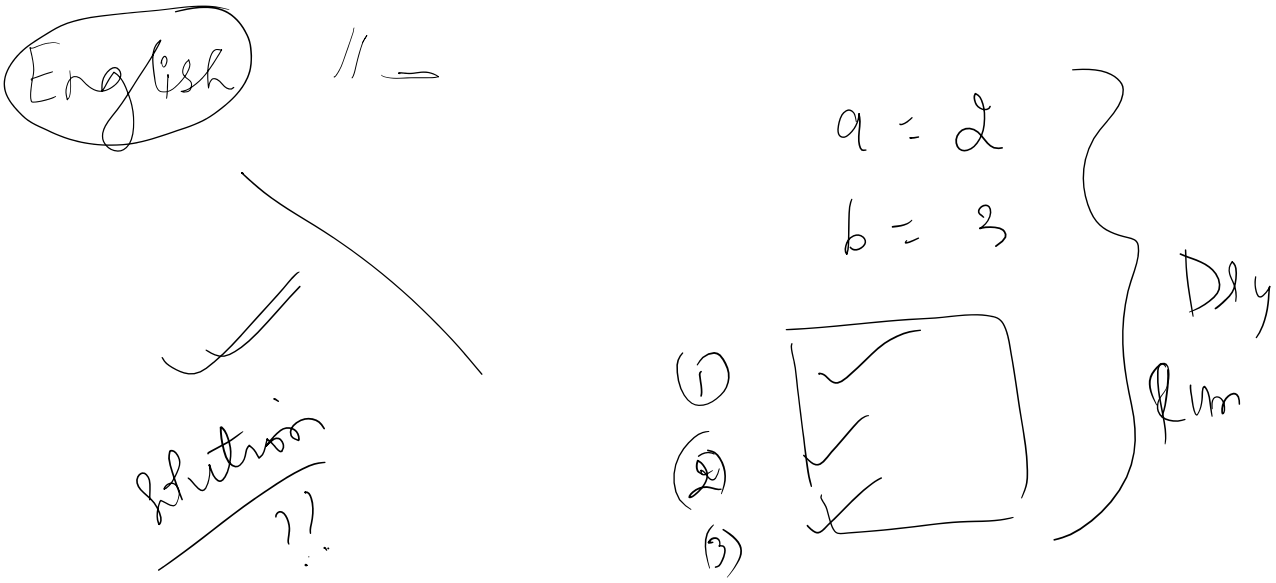


Operator:

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. eg var x = 2; x-- => 1.

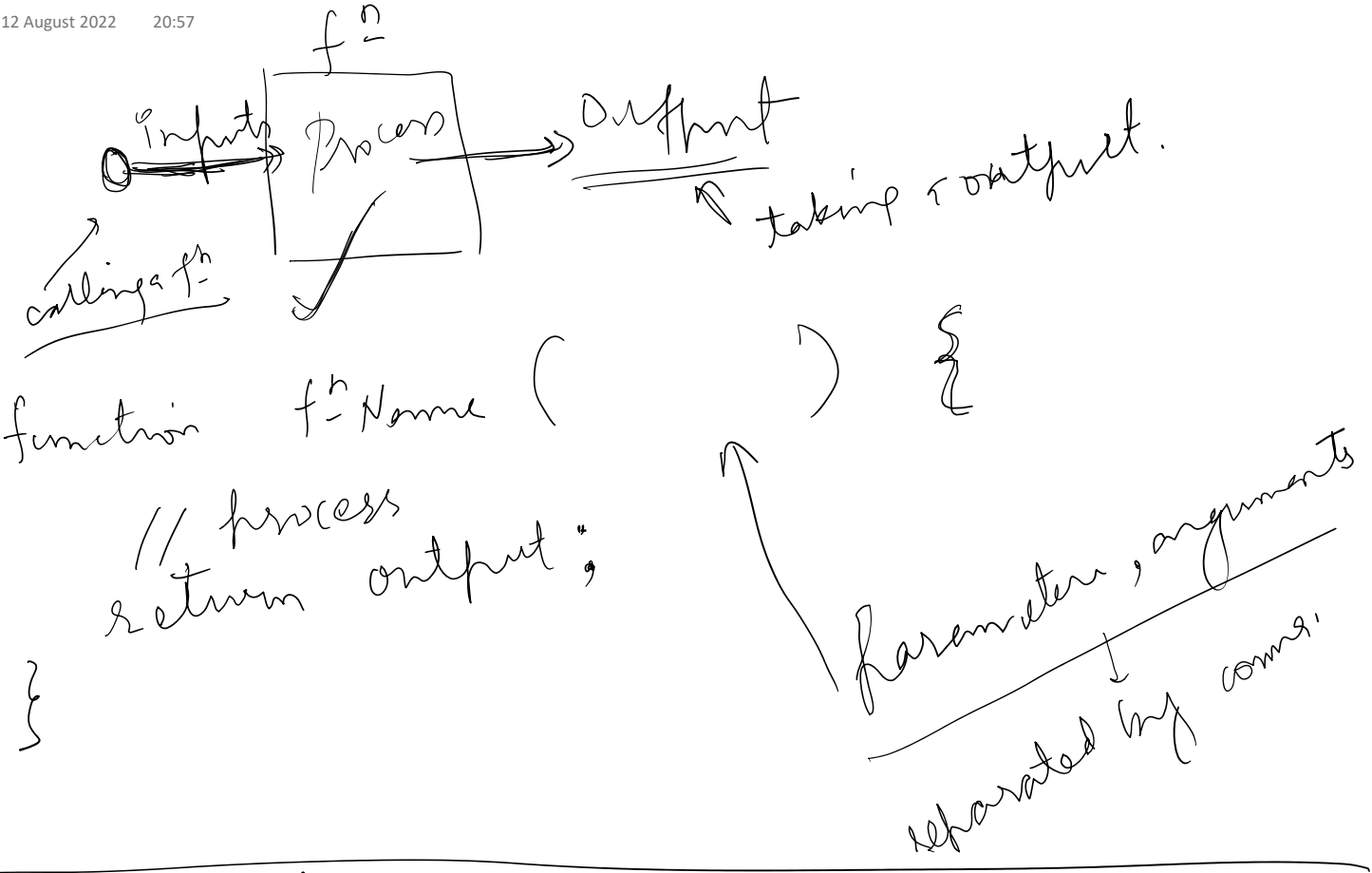
+ looks good, - is not working.





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

12 August 2022 20:57



Square average = 'N' items → prices = Sum of every value / 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.

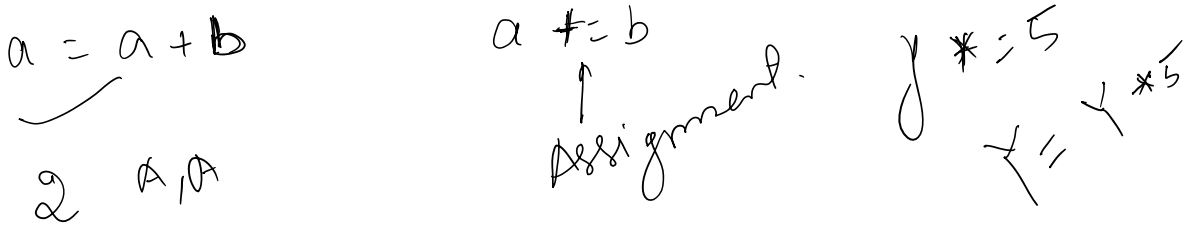
a = b - a
a -= b
a = @ b

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

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

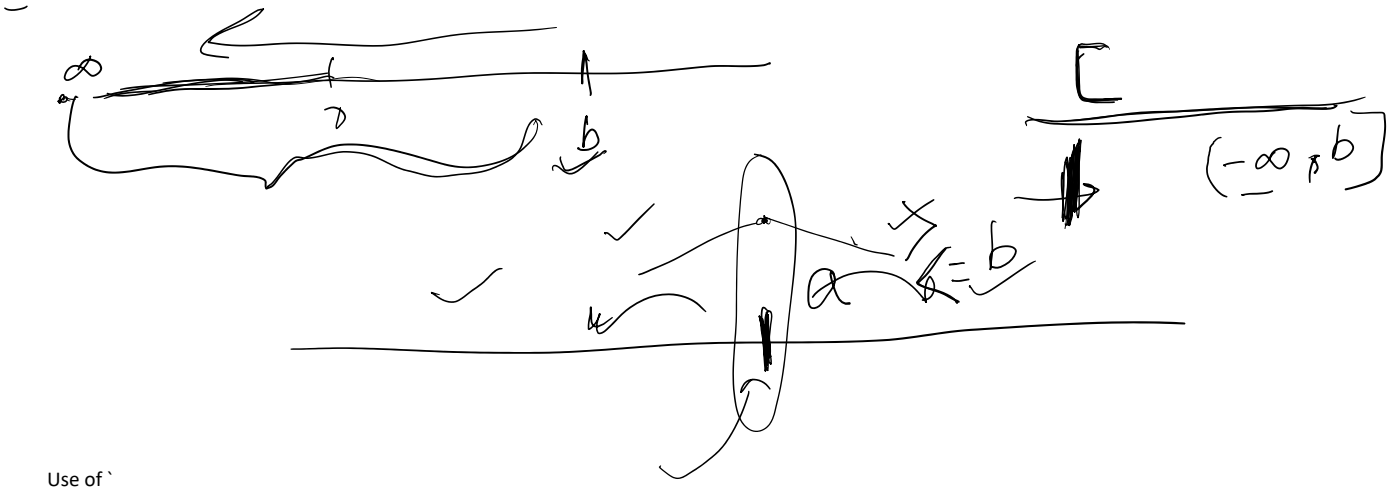
variable. LHS <- RHS

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



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";



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.

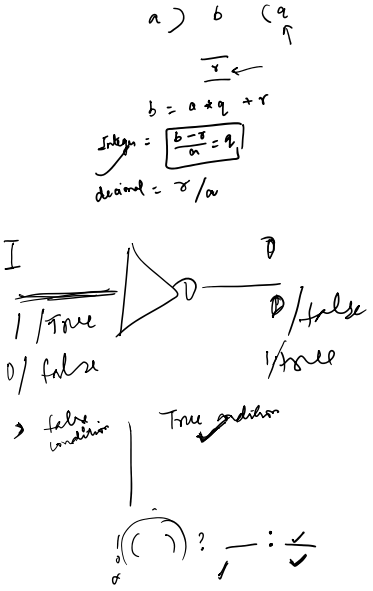
Input	Output
1	0
0	1

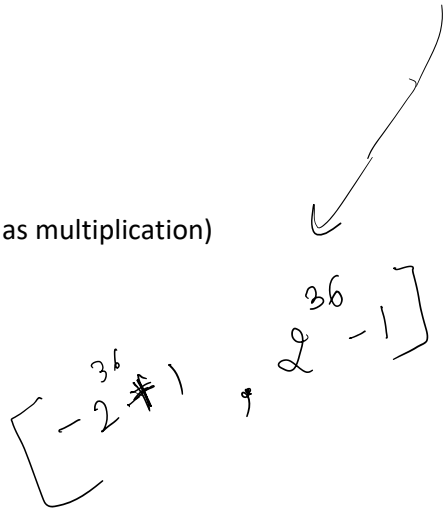
ii) AND(&&) -> (for learning, it works as multiplication)

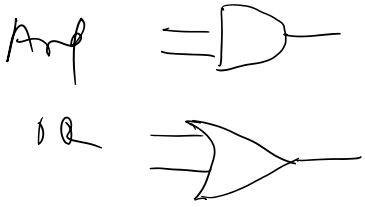
Input1	Input2	Output
0	0	0
0	1	0
1	0	0
1	1	1

iii) OR (||) ->
(for learning, it works as Addition)

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







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

$123 = 1\times10^2 + 2\times10^1 + 3\times10^0$

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 x

$1\times2^2 + 5\times2^1 + 3\times x^0$

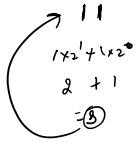
$\frac{1}{3} = 0.\overline{3}$

$\frac{1}{7} = iR$

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

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

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



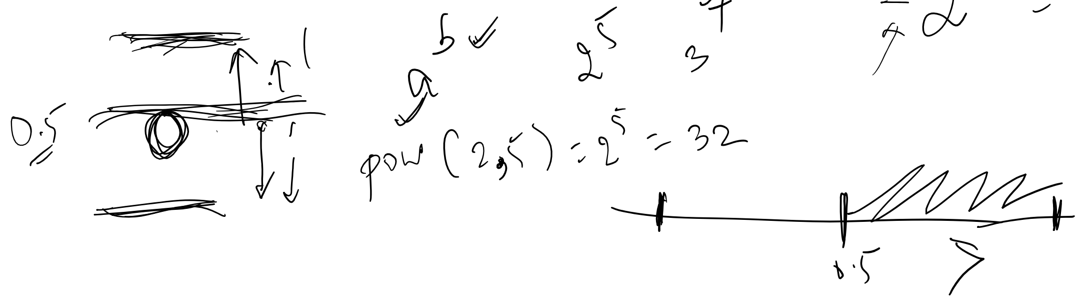
1. toString() => changing the given value into string. e.g var a = true; a = a.toString();
2. toFixed() => return a string, with the number written with a specified number of decimals.
   a. 9.789.toFixed(2) => 9.79
   b. 9.789.toFixed(0) => 10
   c. 9.789.toFixed(4) => 9.7890
3. toPrecision() => it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

Math.PI

1. abs :- it returns absolute value. e.g it return the positive part of the negative values. 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(included) and 1(excluded).
9. sqrt:- square root
10. cbrt: cube root.

magnitude / absolute value

sign, direction



6 - 7

[6,7] = Math.random() + 6

[0,1] + 6 => [6,7]

0.5, 0.8

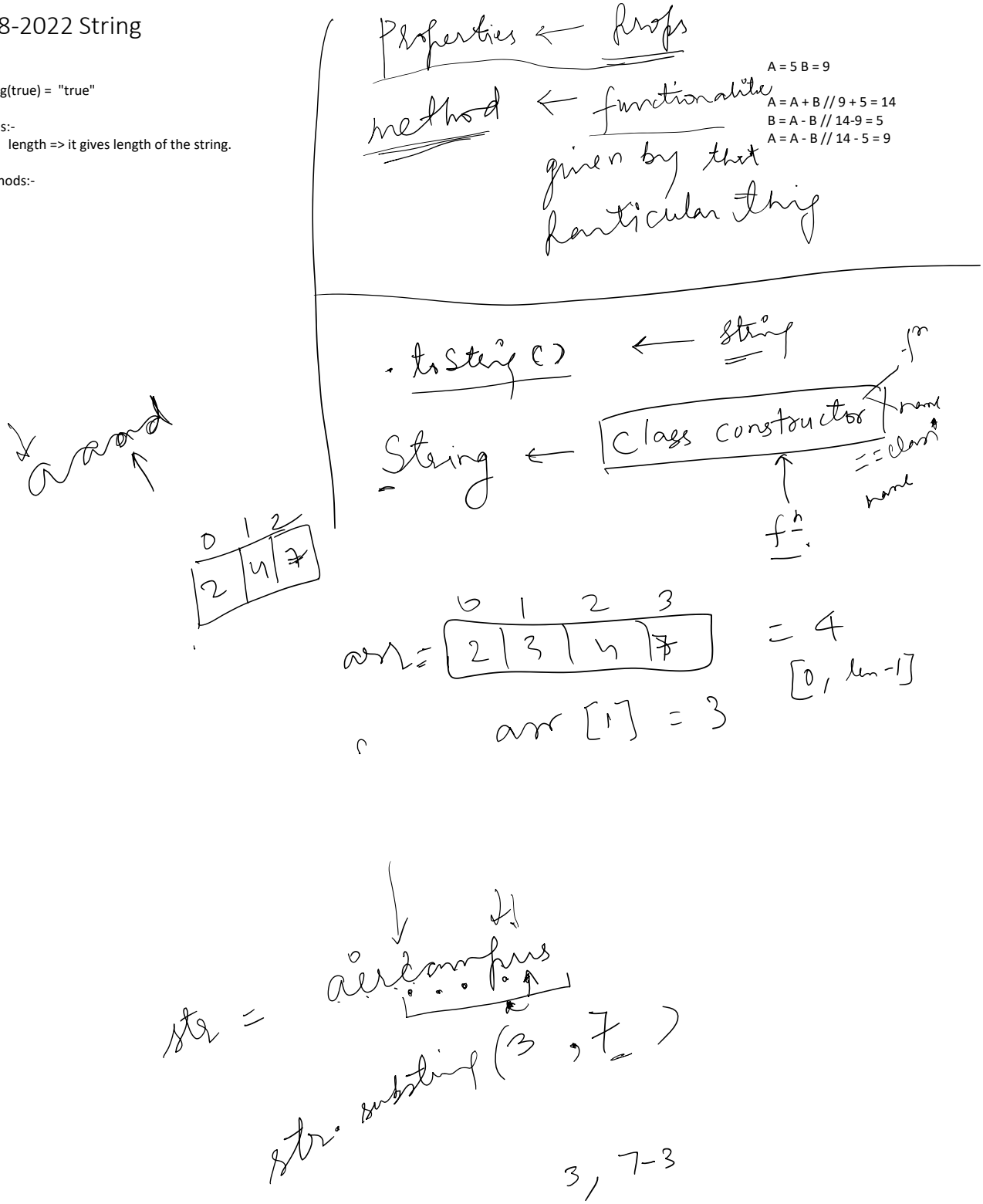
19-08-2022 String

String(true) = "true"

Props:-

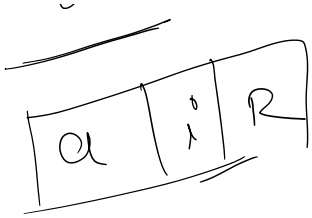
1. length => it gives length of the string.

Methods:-

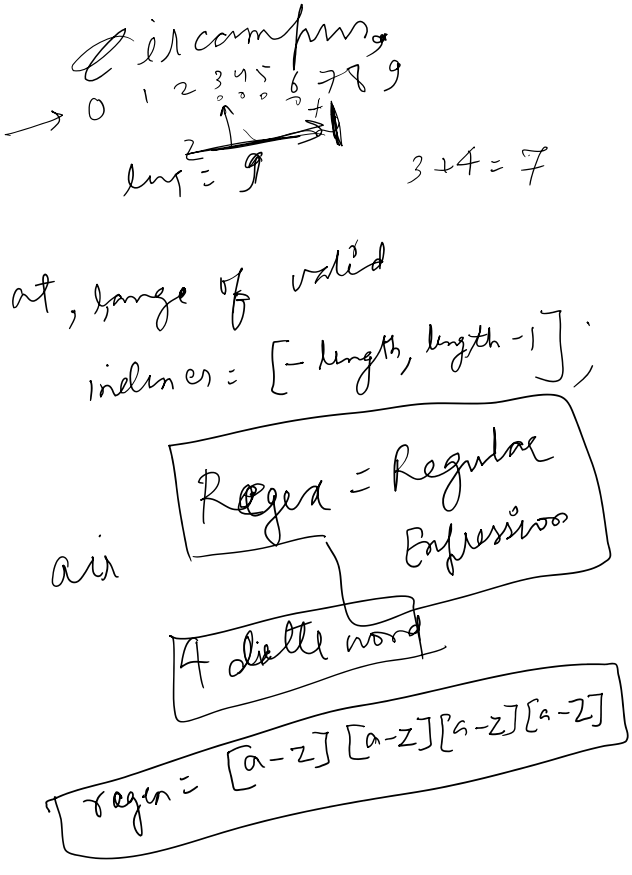


Non-negative numbers => positive number including 0,
positive numbers
Non-positive numbers=> negative number including 0,
Negative numbers.

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



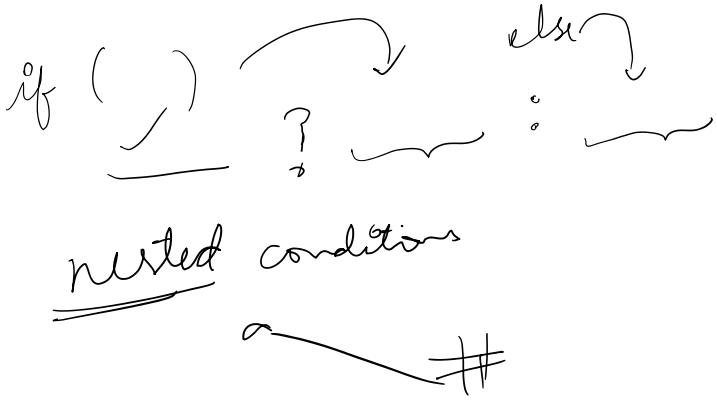
```
var str = String(2477777444);

// 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 occurence 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));
```



```
// 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 value1:
         // code
         break;
      case value:
         // code
         break;
      default:

      Switch case matches by ===.
      If the statement is true, than it consider every further cases are true.
      So break statement is needed.

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

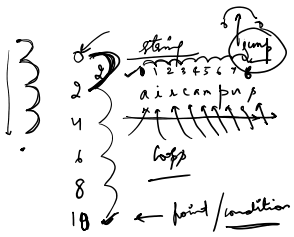
Loops:- For loop

22 August 2022 20:50

for(start point, end condition, jump)

for(starting point, termination condition, increment)

```
for(var i = 0; i <= N; i = i + 1) {

}
```



$i = i + 1$

$i += 1$

$i++$

$$1+2+3+\ldots+n = \frac{(n*(n+1))}{2}$$

$1+2+3+4+5 = \frac{5*6}{2} = \frac{30}{2} = 15$

15

$\rightarrow 2+4+6+8+\ldots+n$ (n/2 terms)

$\rightarrow 2(1+2+3+4+\ldots+\frac{n}{2})$

$\rightarrow 2\left(\frac{50*51}{2}\right)$

$\left(\frac{n}{2}\right)*\left(\frac{n}{2}+1\right)$

$50*51$

Nested loops.

```
For() {
        For() {
                For() {
                }
        }
}
```

4 rows ✓

for (4 rows) {

for (space)


4 rows
n stars
for
for (space)
for (stars)

Loops:- While, Do-While.

23 August 2022 21:28

start point, end condition, jump

WHILE:-

```
// start point;

while( termination condition ) {

      // jump;
}
```

```
For (var I = 0; I < 10 ; i++)

Equivalent while code;
Var I = 0;

While( I < 10 ) {

      i++;
}
```

start point, end/termination condition, jump

```
for( start point; end condition; jump )
```

WHILE:-

```
// start point;

while( termination condition ) {

      // jump;
}
```

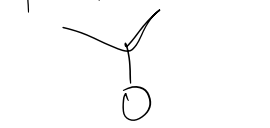
DO_WHILE

```
// start point

Do {
      //code
      //jump
}
While(termination condition);
```

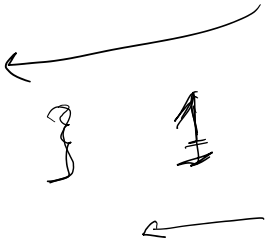
While → Entry Controlled loop

Do-while → Exit controlled

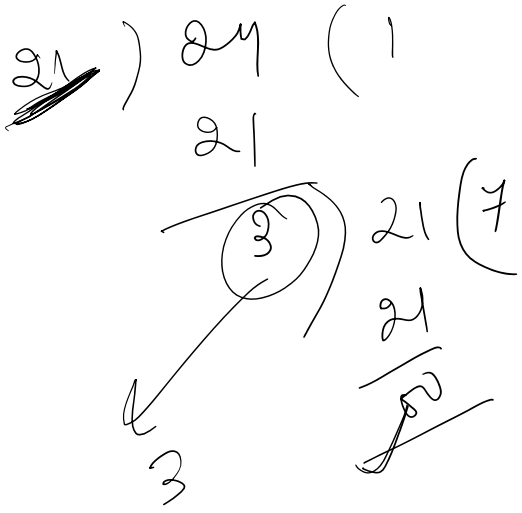
for + while

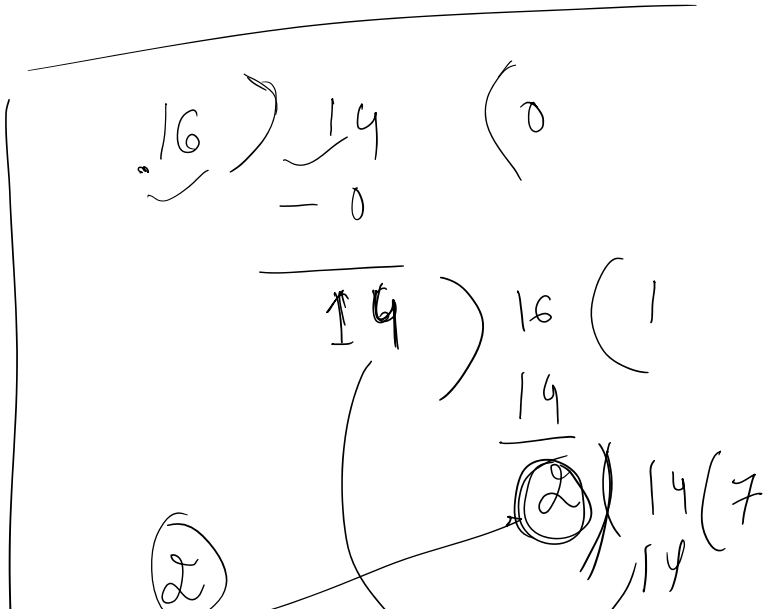


do while

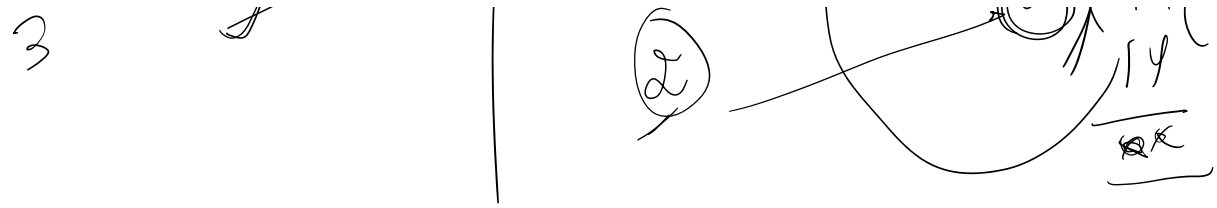


21 24 14, 16





3

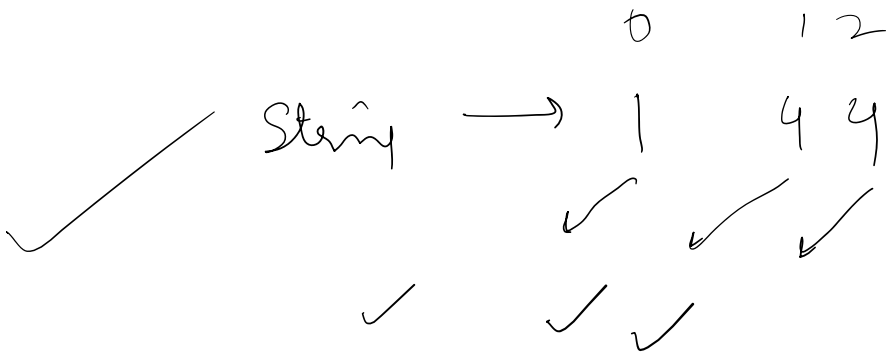


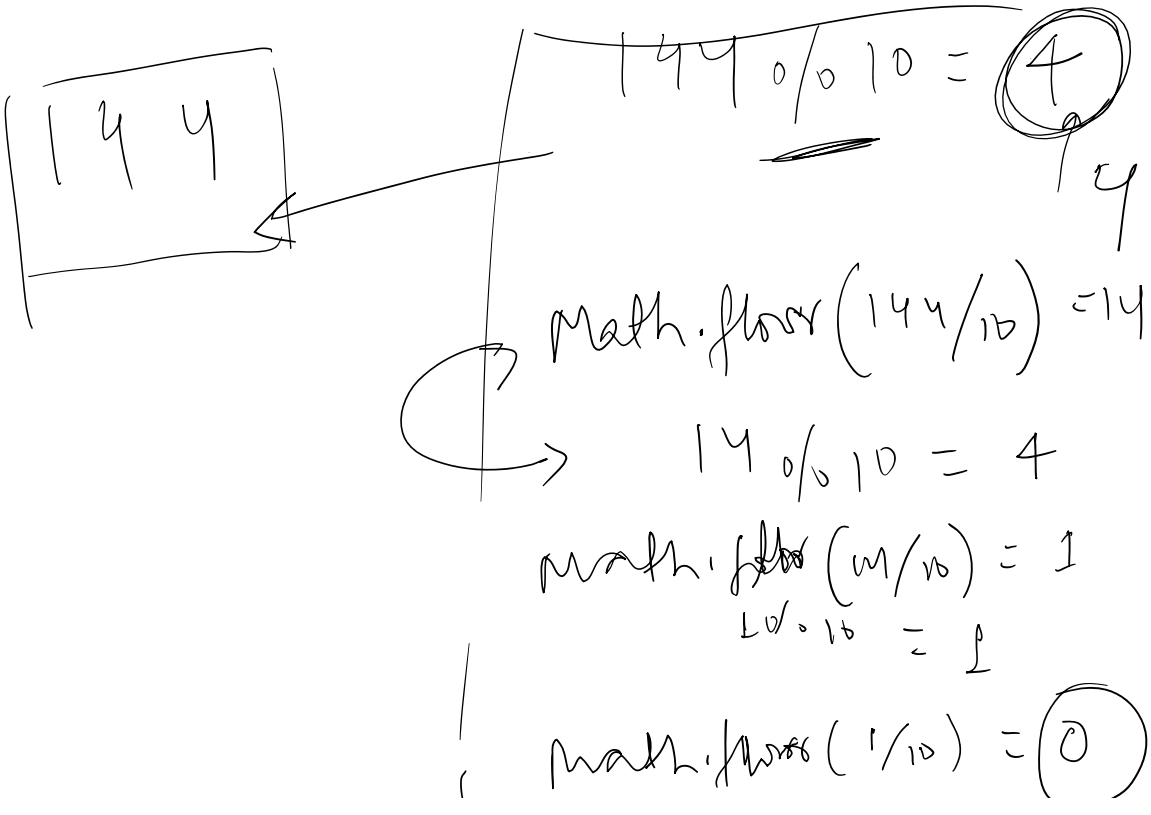
LCM & HCF

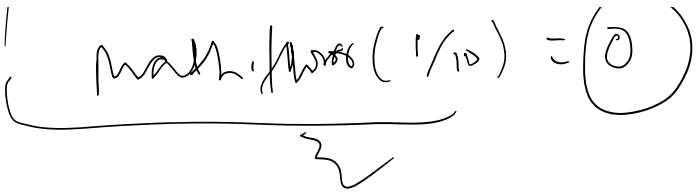
a * b

LCM * HCF = product of nos.

LCM = a*b / HCF



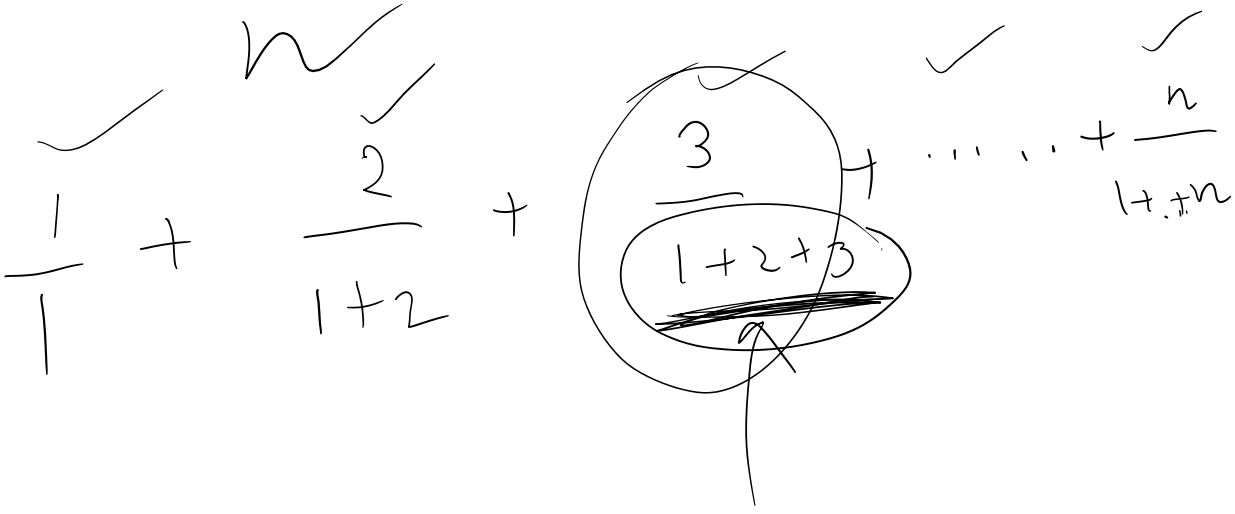




25-08-2022 Practise on loops.

25 August 2022 20:49

MARS NUMBER.
3 and 4



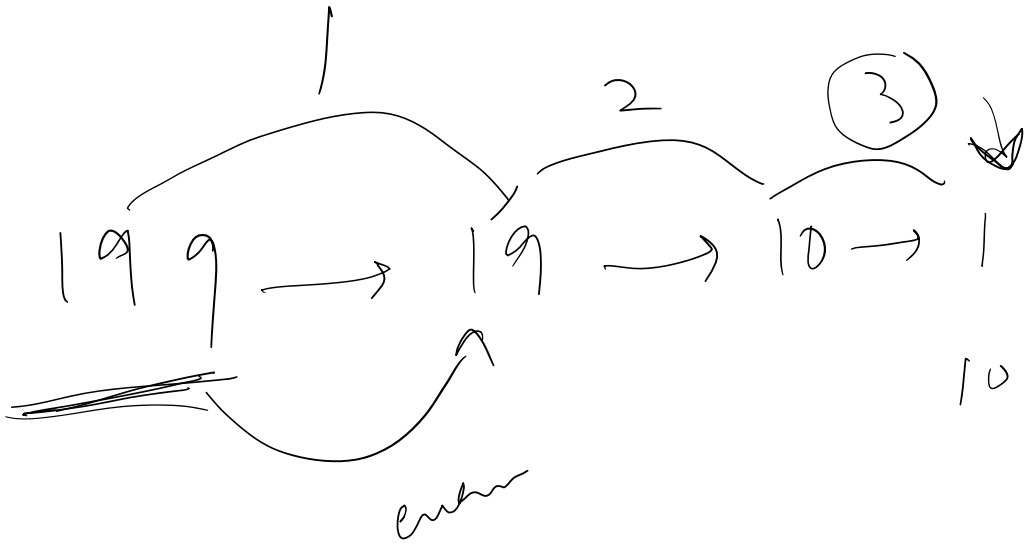
Take an integer input and print 'YES' if the integer is a mars number, else print 'NO'.

Hint:
A mars number is a number if the sum of its digits can be reduced to single digit in even number of steps.
Example:
199 => 19 => 10 => 1
Number of steps = 3
Hence the number is not a mars number.

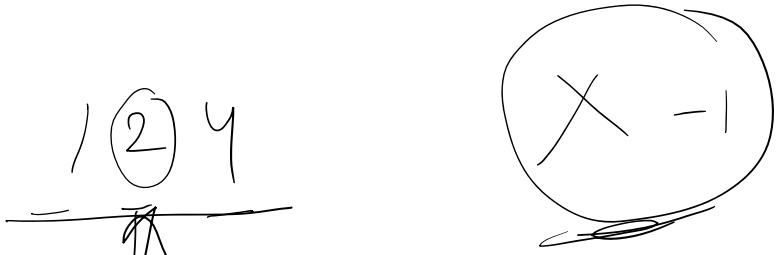
Input:
A single integer input
Output:
Print 'YES' if the number is a mars number, else print 'NO'

Example:
Input:

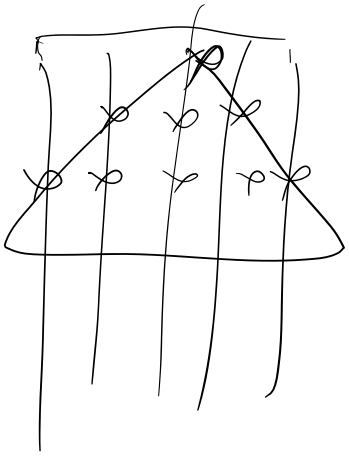
199
Output:
NO

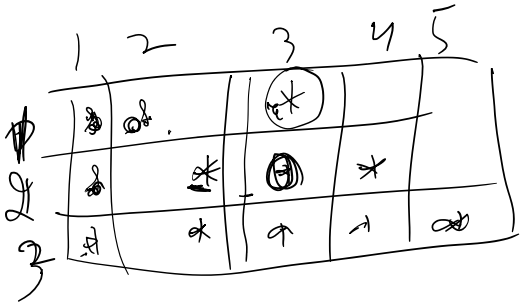


while (num < 10) {
num = sum of digit (num);
count ++;
}



Random() + x = [x,x+1)





Random() = [0, 1)

Searching

14 September 2022 20:49

Question1:- Use two pointer approach. Given a binary array(array contains only O and 1) we have to sort the array OR we have to separate 0 and 1 like 0 should be in left of the array and 1 should in right of the array.

example:- Input = [1,1,0,1,0];
output = [0,0,1,1,1]

1. Approaches.
 separate 0 and 1.

Approach 1

```
function separate0And1(arr) {
 var countOf0 = 0; // O(1)
 for( var index = 0 ; index < arr.length; index++) { // O(arr.length)
   if(arr[index] == 0) {
     countOf0++;
   }
 }

 for(var index = 0; index < countOf0; index++) { // O(countOf0)
   arr[index] = 0;
 }

 for(var index = countOf0; index < arr.length; index++) { // O(countOf1)
   arr[index] = 1;
 }

 return arr; // O(1)
}
```

TC
O(1) + O(arr.length) + O(countOf0) + O(countOf1) + O(1);

O(1 + arr.length + countOf1 + countOf0 + 1) ;
O(2 + arr.length + arr.length);
O(2 + 2* arr.length);
O(2*arr.length) => O(n) => n is arr.length;

Approach 2

```
function sortByTwoPointer(arr) {
 var start = 0 , end = arr.length - 1 ;


 while( start < end ) {

        while( start < end && arr[start] %2 == 0 ) {
        start++ ;
        }

        while( start < end && arr[end] %2 == 1 ) {
        end-- ;
        }

        if( start < end ) {
        swap(arr[start], arr[end]);
        start++ ;
        end-- ;
        }
 }

 return arr;
}
```

Question:- Given a interger array, we have to separate even and odd numbers.

// 1. we don't need to maintain the order of input.
// 2. We have to maintain the order of input.

[23, 45, 6, 8, 19, 12];

[6,8, 12, 23,45,19];

[8, 12, 6, 23, 19, 45]

Approach 3:

```
function sorting(arr)
{
 var i=-1;
 for(var j=0;j<arr.length;j++)
 {
   if(arr[j]==0)
   {
     i++;
     var temp=arr[i];
     arr[i]=arr[j]
     arr[j]=temp;
   }
 }
 return arr;
}

console.log(separate0And1([1,1,0,1,0]));
```

Question 2:- Given two objects, check whether both of them are equal or not.
Equal Object means => Both Object have same keys associated with same values.

example:-
Input 1
Obj1 = { a: 23, b: 34, c: 89 }
Obj2 = { a: 23, b: 34, c: 89 }

Ouput true;

Input 2
Obj1 = { b: 34, a:{v:"90"}, c: 89 }
Obj2 = { a: {v:"90"}, b: 34, c: 89 }

Ouput true;

Input 3

Obj1 = { b: 34, a: 23, c: 890 }
Obj2 = { a: 23, b: 34, c: 89 }

Ouput false;

{ b: 34, a:{v:"90"}, c: 89 }
{ a: {v:"90"}, b: 34, c: 89 }

Object {
 keys are Equal
 associated values are Equal

 value can be Object.
}

Recursion.

```
function checkObjectsAreSame(obj1, obj2) {
 var keysOfObj1 = Object.keys(obj1).sort();
 var keysOfObj2 = Object.keys(obj2).sort();

 if( keysOfObj2.length != keysOfObj1.length ) {
  return false;
 }

 for( var index = 0 ; index < keysOfObj1.length; index++ ) {
  if(keysOfObj1[index] != keysOfObj2[index]) {
   return false;
  }

  var value1 = obj1[keysOfObj1[index]];
  var value2 = obj2[keysOfObj2[index]];

  if( typeof(value1) != typeof(value2)) {
   return false;
  }

  if( typeof(value1) == 'object' && !checkObjectsAreSame(value1, value2)) {
   return false;
  }

  if(value1 != value2) {
   return false;
  }
 }

 return true;
}
```

O(number of key value pairs present in all the given object and their
nested object values);

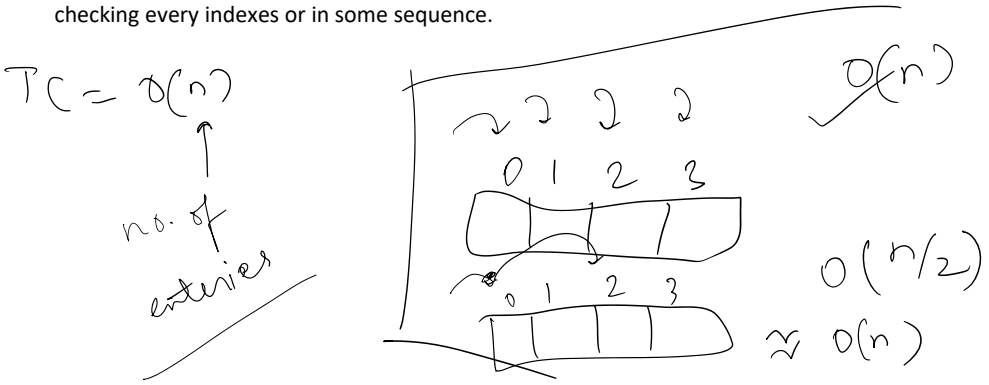
Approach 2:

```
function deepEquality(obj1 , obj2 ) {
 let key1 = Object.keys(obj1) ;
 let key2 = Object.keys(obj2) ;

 if(key1.length !== key2.length ) {
        return false ;
 }

 for( key of key1 ) {
          let val1 = obj1[key] ;
          let val2 = obj2[key] ;

          let areObject = isObject(val1) && isObject(val2) ;

          if( ( areObject && !deepEquality(val1 , val2 ) ) || ( !areObject   && val1 !== val2 ) ) {
          // If the value of the property are objects
          // and not equal
            return false ;
  }
 }
 return true ;
}
```

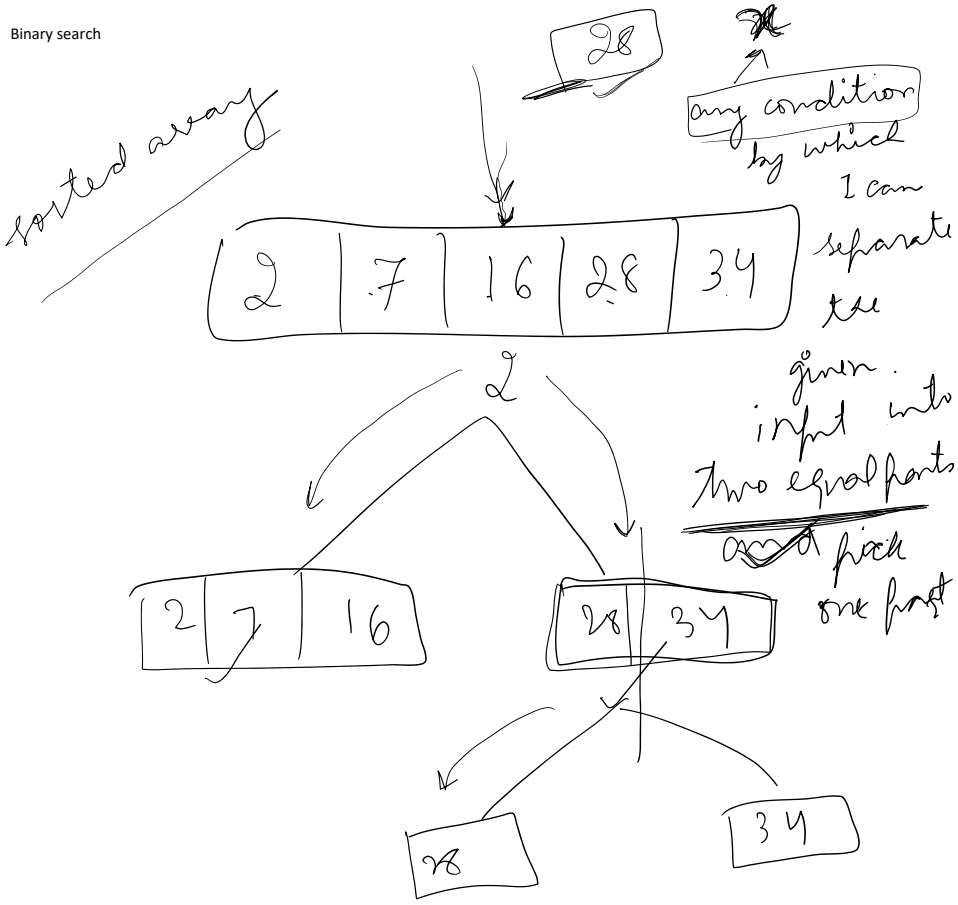
value1 = {v:"90", k: "100"};
value2 = {v:"90", ko: "100"};

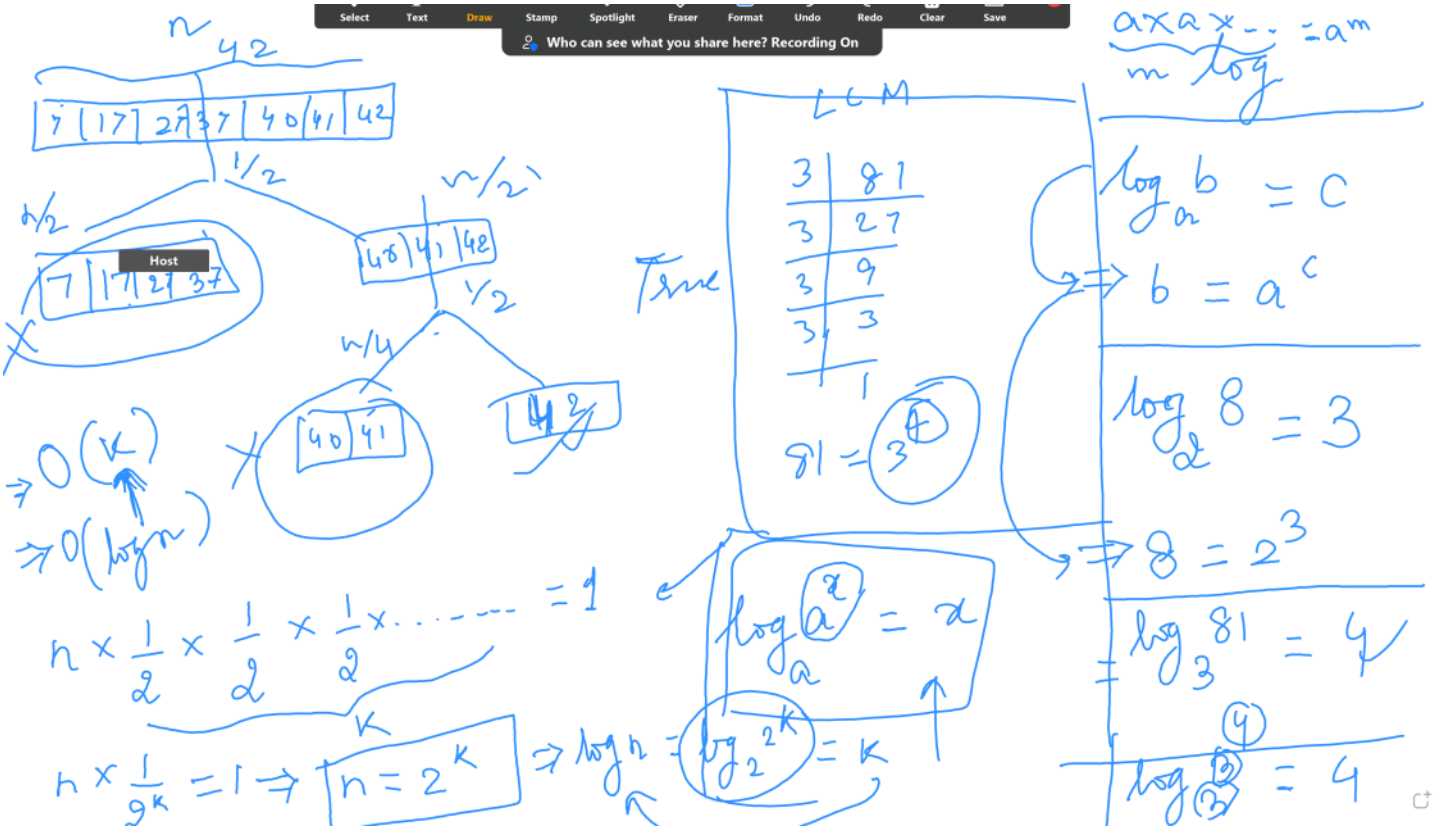
---------- SEARCHING ----------

1. Linear Search: - It says you have to search linearly means we are checking every indexes or in some sequence.



2. Binary search




Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Host
LCM
Time
axax... = am
log
$\log_a b = c \Rightarrow b = a^c$
$\log_2 8 = 3 \Rightarrow 8 = 2^3$
$\log_3 81 = 4$
$\log_a a^x = x$
$n \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \ldots = 1$
$n \times \frac{1}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow \log n = \log_2 2^k = k$
$\Rightarrow O(k)$
$\Rightarrow O(\log n)$