Full Stack Developer

Code kata skills

Fundamentals of coding's…

Prepared by,

Kalilur Rahman A R

ServiceNow Developer

```
/*
-------- CODEKATA PROBLEM SOLVING DAY 1 t0 4 ----------
10 - dec- 2022
-----------------
Basics of Zen portal in JS : JS codekata Day 1
variable declaration :
Types  : var, let, const
var fruit = 'Apple'
let guviMentor = 'Kalil'
const studentName = 'Khalidh'


//reassiging of variable but dont use var keyword :
fruit = 'orange'
guviMentor = 'kasheef'
studentName = 'Mohammed Mafaaz' // You cannot change the variable constant it
is immutable
console.log('Fruit : ', fruit)
console.log('Guvi Mentor : ',guviMentor)
console.log('Student name : ',studentName)
o/p :
Fruit :  orange
Guvi Mentor :  kasheef
Student name :  Khalidh
-----------------------------------
Data types :
1. String --> character " "
2. number --> 546541.64564
3. Boolean --> true or false
4. array --> ['string', number, boolen] - It contains all kind of data types in
single container
var fruit = 'Apple';
var bool = true;
var number = 4565465;
console.log(fruit, bool, number)
o/p: Apple true 4565465
```

```
-----------------------------------

Conditions : (If, else, If else)

Simple If condition :

Eg..,

var money = 5

if (money == 5){

    console.log('Chocii is affordable');

} else {

    console.log('No chocolates');

}

o/p : Chocii is affordable

Else if and Nested if condition :

Eg..,

var money = 10

if (money >= 5){ // greater than and greater than or equal to

    console.log('Big Chocii is affordable');

}if (money >= 2){

    console.log('Small chocolates');

} else {

    console.log('No chocaltes');

}

o/p :

Big Chocii is affordable

Small chocolates

-------------------------------

Strict mode :

console.log('Strict mode, Will check with data type :',5 === '5')

o/p :

Strict mode, Will check with data type : false

---------------------------------

Loop : For loop - as of now

Eg..,

// single opertions to run multiple times - you can use var and let but not
const

for(let i = 0; i <= 5; i++){
```

```
    console.log('Iam called : ', i)
}
o/p:
Iam called :  0
Iam called :  1
Iam called :  2
Iam called :  3
Iam called :  4
Iam called :  5
-----------------------------
Array : It is set of many datatypes which is stored in single container

let array = ['kalil', 'kasheef', 'kareem', 'khalidh'];
console.log('Array is :', array)
o/p: Array is : (4) ['kalil', 'kasheef', 'kareem', 'khalidh']

array.push('Mafaaz'); // will add value at the last
console.log('Array is :', array)
o/p : Array is : (5) ['kalil', 'kasheef', 'kareem', 'khalidh', 'Mafaaz']

array.pop('Mafaaz'); // will remove only the last value in array
console.log('Array is :', array)
o/p : Array is : (4) ['kalil', 'kasheef', 'kareem', 'khalidh']

array.unshift('Zaynab'); // adding in first index
console.log('Array is :', array)
o/p : Array is : (5) ['Zaynab', 'kalil', 'kasheef', 'kareem', 'khalidh']

array.shift('Zaynab'); // removing from first index
console.log('Array is :', array)
o/p : Array is : (4) ['kalil', 'kasheef', 'kareem', 'khalidh']

// In Array index will starts with zero - 0
```

```
let array = ['kalil', 'kasheef', 'kareem', 'khalidh'];
console.log('Length of Array :', array.length); o/p : Length of Array : 4
console.log(array[0]); o/p : kalil

for(let i = 0; i < array.length; i++){
    console.log( i +' '+array[i]);
    // Here i is the count of number and array[i] print array
}
o/p :
0 kalil
1 kasheef
2 kareem
3 khalidh

Eg.., --> 2 checking value present in the array if not needs to add
let array = ['kalil', 'kasheef', 'kareem', 'khalidh'];
for(let i = 0; i < array.length; i++){

    if (array[i] == 'Zaynab'){
        console.log(array[i]);
        break;
    } else {
        array.push('Zaynab');
        break;
    }

}
console.log('The New array is :', array);
o/p :
The New array is : (5) ['kalil', 'kasheef', 'kareem', 'khalidh', 'Zaynab']
-----------------------------------------
11 - dec - 2022 :
**************************
Functions : Its a very important thing in Javascript
```

Eg.., calling a function need to use function() keyword

```
function kalil(){
    console.log('This is kalil');
    console.log('This is B43 batch');
    console.log('This is FSD');
}
kalil();
```

o/p :

```
This is kalil
This is B43 batch
This is FSD
```

Eg..,2

```
function addNumbers(){
    // set of operations
    var num1 = 9;
    var num2 = 98;
    var total = num1 + num2;
    console.log('The total is', total)
}
addNumbers();
```

o/p :

```
The total is 107
```

Eg.3.., --> Functions with arguements:

```
function addNumbers(num1, num2){
    // set of operations
    var sum = num1 + num2;
    console.log('The sum is', sum)
}
addNumbers(45,58);// calling a function then only it will execute the set of operations
```

o/p:

```
The sum is 103
function subNumbers(num1, num2){
    // set of operations
```

```
    var sum = num1 - num2;
    console.log('The sub is', sum)
}
subNumbers(45,58);// calling a function then only it will execute the set of operations
```

o/p:

```
The sub is -13
```

Problem Approach : How to do it?

Eg.,

A car has milege per hour plus speed per hour and fuel capacity and a car how long it can travel

Step 1 : Layman Approach

we know milege is 30 km/ltr, speed is 75 km/hr and fuel capacity is 5 ltr

total km is m * f = 30 * 5 = 150

fuel emptied is 150/75 = 2 hrs this is called Lyaman Approach

step 2 : Think how to do with the code

Eg..,

```
function remainingHoursOfTravel(milege, speed, fuel){
    var totalKilometers = milege * fuel;
    var totalHours = totalKilometers / speed;
    console.log('The total hours is', totalHours, 'Hrs')
}
remainingHoursOfTravel(30, 87, 6)
```

o/p :

```
The total hours is 2.0689655172413794 Hrs
```

// Split and join() - keywords

split - means it will split based on the value inside () and put it on array []

example : if kalil is there, if split('l') => ['ka', 'i', '']

Eg..,

```
const userInput = "k a l i l u r";
console.log('The user input : ',userInput);
```

o/p:

```
The user input :  k a l i l u r
const output = userInput.split(' ')
console.log('The output is :', output);
o/p:
The output is : (7) ['k', 'a', 'l', 'i', 'l', 'u', 'r']
const joinOutput = output.join(',');
console.log('The join output is :', joinOutput)
o/p:
The join output is : k,a,l,i,l,u,r
-------------------------------------------------
// strings keyword method
const userInput2 = 'Rahman'
let concatValue = userInput.concat(' '+userInput2); //concatinations
console.log('The concatenate is :', concatValue)
o/p :
The concatenate is : k a l i l u r Rahman
//replace : will replace the word from which to what
let para = 'Hey there you failed the test'
let replaceValue = para.replace('failed', 'passed');
console.log('The replaced word :', replaceValue)
o/p:
The replaced word : Hey there you passed the test
//parseInt : will convert the string into number note: only number
//Method 1
let num = '5';
let num2 = 5;
let total = parseInt(num) + num2;
console.log('The total value is :', total)
o/p:
The total value is : 10
//Method 2:
console.log(+'6'+ 8); - o/p : 14
console.log(8 + (+ '9')); - o/p : 17
--------------------------------
```

```
sort() - Asc to Desc or Desc to Asc :
let samArr = [8, 4, 55 ,62 ,4 ,2 ,4]
let sortArr = samArr.sort();
console.log('The sorted array - default (Ascending order):', sortArr)
o/p :
The sorted array - default (Ascending order): (7) [2, 4, 4, 4, 55, 62, 8]
let reversed = sortArr.reverse()
console.log('The sorted array - (Descending order):', reversed)
o/p:
The sorted array - (Descending order): (7) [8, 62, 55, 4, 4, 4, 2]

//sorting strings and alphabet
let samAlphabet = ['kalil', 'zaynab', 'hubby', 'wife']
let sortAplphabet = samAlphabet.sort()
console.log('The string sorting - (Ascending) :', sortAplphabet)
o/p:
The string sorting - (Ascending) : (4) ['hubby', 'kalil', 'wife', 'zaynab']
let revAlpha = sortAplphabet.reverse()
console.log('The string sorting - (Descending) :', revAlpha)
o/p:
The string sorting - (Descending) : (4) ['zaynab', 'wife', 'kalil', 'hubby']
//chaining or Dot walking
let chsamArr = [8, 4, 55 ,62 ,4 ,2 ,4]
let chsortArr = samArr.sort().reverse();
console.log('The sorted array - (Chaining : Descending order):', chsortArr)
o/p:
The sorted array - (Chaining : Descending order): (7) [8, 62, 55, 4, 4, 4, 2]
---------------------------------
// Sample code :
var nameInput = 'kal ilur';
var splitValue = nameInput.split(' ').join('');
for( let i = 0; i < splitValue.length; i++ ){
    console.log(i, splitValue[i])
}
```

```
o/p:
0 'k'
1 'a'
2 'l'
3 'i'
4 'l'
5 'u'
6 'r'
-----------------------
// Sample code :
const input = "30 5 75";
const myArr = input.split(' ')
const output = parseInt(myArr[0] * myArr[1] / myArr[2])
console.log('The output is :',output)
o/p:
The output is : 2
------------ or ------------
const input = "30 5 42";
const myArr = input.split(' ')
let milage = myArr[0]
let fuel = myArr[1]
let speed = myArr[2]

let km = milage * fuel;
let totalHours = km / speed;
console.log('The hours travel :', totalHours)
console.log('The hours travel :', totalHours.toFixed(3)) ***
// toFixed : will reduce the decimal points based on the value
o/p :
The hours travel : 3.5714285714285716
The hours travel : 3.571
-----------------------------------------
****** 17-Dec-2022 ********
-----------------------------------
```

```
Note : parseInt() and unary operator are same

console.log('Without decimal :',parseInt(52.53))

o/p : Without decimal : 52

console.log('With decimal :',parseFloat(52.53))

o/p : With decimal : 52.53

// Instead of above code can use unary operator

console.log('With decimal :', +(52.53))

o/p : With decimal : 52.53

trim() : --> It will remove the whitspaces in the front and back side of the
string

-----------------------------------

// Map function()

let inp = '1 23 3 4 5'

let a = inp.split(' ').map(Number);

console.log(a)

o/p : [1, 23, 3, 4, 5]

// The Best method to sort() or to use sort()

//This case will work for more than 10

const arr = [1,30,4,21,10000];

let b = arr.sort();

console.log('This wont work some cases :',b)

o/p : This wont work some cases : (5) [1, 10000, 21, 30, 4]

//If it is more then 10 means

arr.sort((a,b) =>{

    if(a > b){

        return 1;

    } else if (a < b) {

        return -1;

    } else {

        return 0;

    }

})

console.log('The corrected sorting :', arr);

o/p : The corrected sorting : (5) [1, 4, 21, 30, 10000]

-------------------
```

Codekata : Array 14:

Ramesh is a student and wants to find out if there is any other student

in his class who has got the same marks as his, in maths.

Help him to find out.

Input Description:

First line contains the number of students in the class

followed by Ramesh’s mark. Second line contains

the marks of all students in the class.

Output Description:

Index of student who got mark same as Ramesh’s mark. If no such mark exists, return -1.

Sample Input :

2 10

1 2

Sample Output :

-1

code:

```
let arr = userInput[0].split(' ');
let n = +arr[0];
let mark = +arr[1];
//map(Number) - will convert as a number from array
let res = userInput[1].split(' ').map(Number);
let flag = false;
let answer = 0;
for (let i = 0; i < n; i++){
    if (res[i] === mark){
        answer = i;
        flag = true;
        break;
    }
}
if (flag){
    console.log(answer);
} else {
    console.log('-1');
```

```
}
```

---

codekata : array 4

You are given with an array. For each element present in the array

your task is to print the next smallest than that number.

If it is not smallest print -1

Input Description:

You are given a number ‘n’ representing size of array.

And n space separated numbers.

Output Description:

Print the next smallest number present in array and -1 if no smallest is present

Sample Input :

7

10 7 9 3 2 1 15

Sample Output :

7 3 3 2 1 -1 -1

code :

```
let size = 7
let arr = [10, 7, 9, 3, 2, 1, 15];
let res = [];
for(let i = 0; i < size; i++){
  let flag = false;
    for (let j = i + 1; j < size; j++){
        if(arr[j] < arr[i]){
            res.push(arr[j]);
            flag = true;
            break;
        }
    }
    if(flag === false){
        res.push(-1);
    }
}
console.log(res.join(' '));
```

---

codekata : companies(2)

Given a string and a number K, change every kth character to uppercase from beginning in string.

Sample Testcase :

INPUT

string 2

OUTPUT

sTrInG

code :

```
let str = 'string';
let number = 2;
let newStr = str.split('');
console.log(newStr);
o/p: ['s', 't', 'r', 'i', 'n', 'g']
if(number === 0){
    console.log(str);
    o/p : ----------
} else {
    for(let i = number - 1; i<newStr.length; i+=number){
        // i+=number : i = i + number, if number is 2 then 2 - 1 is 1
        // so it will take 1 as 1 + 2(number) = 3, see the below pic
        console.log(newStr[i]) o/p : t, i, g
        newStr[i] = newStr[i].toUpperCase();
        console.log(newStr[i]) o/p : T, I, G
    }
}
console.log(newStr.join(""))
```

o/p :

sTrInG

.....

Code optimization for above code:

```
let str = 'string';
let number = 2;
```

```
let newStr = str.split('');
console.log(newStr);
if (number != 0){
    for(let i = number - 1; i<newStr.length; i+=number){
        console.log(newStr[i])
        newStr[i] = newStr[i].toUpperCase();
        console.log(newStr[i])
        }
}
console.log(newStr.join(""))
```

Checking for palindrome :

```
let str = 'Oyo';
let checkValue = str.toLowerCase();
console.log(checkValue)
```

o/p : oyo

```
let finalStr = checkValue.split("");
let reverseValue = finalStr.reverse();
console.log(reverseValue)
```

o/p : ['o', 'y', 'o']

```
let joinValue = reverseValue.join("");
console.log(joinValue)
```

o/p : oyo

```
if (checkValue === joinValue){
    console.log("Yes");
} else {
    console.log("No")
}
```

Exponential or finding the power of

Math.power is professional way in Js script

```
let a = 2
```

```
let b = 12
let powerOfBase = Math.pow(a, b);
console.log(powerOfBase)
o/p : 4096
```

Company level question : codekata problem 11 in companies

Given a number N print a right angled traingle structure with the starting level as single 1 and every immediate proceeding level with 2 more additional ones than the previous level .Repeat the pattern for N levels.

Input Size : N <= 1000

Sample Testcase :

INPUT

3

OUTPUT

1

1 1 1

1 1 1 1 1

code :

```
let inputs = 3;
let output = "";
for(let i = 0; i<inputs; i++){
    for(let j = 1; j <= i; j++){
        output += "1" + " " +  "1" + " "
    }
    output += "1"
    output += "\n"
}
console.log(output)
```

Note :

In the above program "for" inside the "for" loop it is called Nested for loops.

steps :

1. if i = 0 then checks with j if it is not same then break the loop

2. if i = 1 checks with j in this case it is same then will come inside the loop and print the o/p : 1 1 and print next o/p and concatenate it then o/p : 1 1 1 and it goes towards new line

Note : In Nested For loop first for loop iteration is 2nd and the second for loop will run twice

*/