**REACT JS, BASIC JAVASCRIPT QUESTIONS**

1. React Life Cycle **important asked difference of each life cycle methods.**

2. forceUpdate and setState Difference - **assigning a value to state is force update**

3. redux architecture - **elaborately need to explain**

4. virtual dom - explained with react for performance

5. closure function - important with example explained

6. arrow functions - explained in map and functions and scopes

7. ES6 futures - let, promise, arrow functions explained

8. axios, fetchGet,fetchPost - explained about services calls trigger

9. async wait - waiting for promise to be resolved

10. about Webpack build - explained build process

11. about testing tools enzyme - explained test cases

12. keys in reactjs - used for performance

13. reactjs promise - explained about reject and success scenarios

14. React code handling mechanism- i told about jslint

15. Print the diagonal elements of a 2 dimensional array using both multiple loops as well as single loop

Input Array: 1 2 3

4 5 6

7 8 9

Output#1: 1 5 9

Output#2: 3 5 7

16. Check if the sum of any two elements of an array = 15(say) using both multiple loop and single loop

17. Input array#1: 1 4 6 8 5 3 8 (Reverse only the elements in the odd placed, The logic should work fine irrespective of the no. of elements in the array being odd or even)

Output#1: 1 3 6 8 5 4 8

18. For every element is an array, look to all elements to its right and replace it with the the largest element

E.g Input: 6 8 7 9 5 6

Output: 9 9 9 9 6 6

19. Split an array into two sub arrays such that sum are equal

20. Reverse a string without using any inbuilt function or second string

Find out the valid expression - **critical problem need to tell multiple algorithms**

Inputs:

1:(a+b)

const input = "(a+b)";

2:

(a+b))

3:

)(a+b)(

4:

((a+b))

(aaa\*a\*)

(a+(b+c)) ==>

Based on Input 1:

postfix:

ab+

prefix:

+ab

Infix:

a+b

Answer:

counter =0;

for(var i = 0 ; i < input.length ; i++){

if(input[i] == '('){

counter++;

}

else if(input[i] == ')'){

counter --;

}

if(counter < 0){

//invalid

}

}

if(counter ==0){

//valid

}

else{

//invalid

}

//

merging two arrays

sorted Arrays

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

2: [6,7,8,9,10]

Using array methods:

arr1.concat(arr2)

Without using array methods

var arr1len = arr1.length;

var concatArr = arr1;

for(var i = 0 ; i< arr2.length; i++){

concatArr[arr1len++] = arr2[i];

}

**ANGULAR, NODE JS and BASIC JAVASCRIPT QUESTIONS**

* Need to share our work experience

Ans:

Totally i have 4 years of working experience in java, spring boot, and I have 2 years of working experience with angular 2 technology.

With this technology previously i have worked on our company’s ticket tracking system. And recently i have joined with photon.

* Reverse an array without using any array methods. (Note: You shouldn't create new array, reverse given array itself)

Ans:

Refer working example

file:///C:/critical/backup/currently\_doings/interview\_prep/samsung/works/reverse\_wo\_temp\_arr.html

* What is closure? Example for closure.

Ans:

1. Closures are functions with preserved data.
2. When ever we declare the function inside another function. The **inside function is are again recreated** when the outer function is called.

Ex:

Refer working example

file:///C:/critical/backup/currently\_doings/interview\_prep/samsung/works/closures.html

* What is prototypical inheritance?

Ans:

1. Every function has the **prototype** property
2. Each prototype property has the property and function
3. **Inheritance** is also called **prototype chain**

Ex:

file:///C:/critical/backup/currently\_doings/interview\_prep/samsung/works/prototype.html

* What is single ton? Example.
* Array Methods & String Methods.
* What is recursive function? Example.
* Angular Js Vs jQuery
* What is prototypical inheritance?
* $scope Vs $rootscope
* How to communicate between controllers
* Service Vs Factory
* ng route vs UI Router
* Bootstrap based questions like they will draw some design in paper. we have to code using bootstrap
* Jquery Selectors
* Angular life cycle
* $watch vs $digest
* Custom directive
* Isolated scope.
* Promise in Angular
* link vs controller.
* Write program for output:

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

* Find output of the program:

(function() {

console.log(1)

setTimeout(function(){console.log(2)}, 1000);

setTimeout(function(){console.log(3)}, 0);

console.log(4);

})();

* Find output of the program

function a()

{

var a = 2;

function b() {

a++;

console.log(a);

function c() {

a++;

console.log(a);

};

c();

};

b();

console.log(a);

}

a();

**Javascript:**

1. Anagram without using sort function.

2. Timestamp to HH:MM:SS conversion without using date method

3. Swap an element in array which is in any order. For example input array = [1,2,3,4,5,6] and output [1,6,3,4,5,2] OR input [1,2,3,4] and output [1,4,3,2]

**Angular:**

1. $scope explain

2. Promise

3. Yield

4. Without using .then how to return success or error.

First question about current project and followed by basic javascript questions.

I suspect interviewer may ask questions in javascript first before they validate knowledge of Angular.

**Few javascript questions they asked:**

1. Compare two strings to see if they are anagrams.

**Example code:**

function compare (a, b) {

var y = a.split("").sort().join(""),

z = b.split("").sort().join("");

console.log(z === y ? a + " and " + b + " are anagrams!" : a + " and " + b + " are not anagrams." );

}

compare ('abc', 'cba');

2. Write an logic to reverse the array, for example input [-3,5,1,3,2,10] and output [10,2,3,1,5,-3]

**Example code:**

var x = [-3,5,1,3,2,10];

function reverse(a) {

var counter = a.length - 1;

for (var i = 0; i < a.length; i++) {

var temp = a[i];

a[i] = a[counter];

counter--;

}

return a;

}

console.log(reverse(x));

Please refresh basic javascript logical things addition to Angular preparation. Below are few basic questions I listed, but we cannot list all of them since we have many logical questions.

1. Find odd number in javascript

2. Fibonacci sequence algorithm in Javascript

3. Find an element in array.

4. OOPS basics

5. ECMA latest

etc...

1. Angular, Define Digest?

2. Define Angular.module?

3. Write a sample login application using google auth, services, promise, resolve, reject?

4. Define promise?

5. What is digest?

6. Jquery element selector?

7. Click even parent, will it click child?

8. Difference between $scope and $rootscope?

9. Define service, providers and factories?

10. Write sample code for Jquery $.Ajax or Angular $http?

11. Angular 1 / Angular 2 / Router configuration?

Q1.

Input array = [1,2,3,4,5,6];

Output should be this [1,6,3,4,5,2];

Q2. What is NodeJS

Q3. What is $scope

Q4. Whats is Promise

Q5. What Is MVC

Q6. What is Digest

Q7. What is Factory write example

Q8. which purpose using dist folder

Q9. Closure in JS sample program

Q10. What are the features of Node.js

Q11. What is Callback in node.js

Q12. How are JavaScript and ECMA Script related

Q13. how model will work in Angular JS

J**avascript:**

1. function a()

{

var a = 1;

function b() {

a++;

console.log(a);

function c() {

a++;

console.log(a);

};

c();

};

b();

console.log(a);

}

a();

What is the output of this function?

2. (function() {

console.log(1);

setTimeout(function(){console.log(2)}, 1000);

setTimeout(function(){console.log(3)}, 0);

console.log(4);

})();

What is the output of this function?

3. <div class="a1">

<div class="b">

Hello, a

</div>

</div>

<div class="a2">

<div class="b">

Hello, b

</div>

</div>

Write jQuery to change html of class b 2nd occurrence?

4. n \* n

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

How to handle matrix design?

5. Array reverse without using .reverse and introducing new array. Input var arr = [1, 2, 3, 4, 5, 6, 7]; Out put : 7, 6, 5, 4, 3, 2 ,1

**Angular:**

1. Watch

2. Digest

3. Broadcast

4. How to share data between controllers?

5. Diff between scope and rootscope

6. Promise

**1. Write a code for number triangle for number 'n':**

n = 5;

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

say, n=3

1

1 2

1 2 3

3

**Answer**:

function(n){

var b=[];

for (i=1;i<=n;i++){

for(j=1;j<=i;j++{

b.push(j);

console.log(j);

}

console("\n");

}

}

======

**2. What is the output of this function?**

function a()

{

var a = 2;

function b() {

a++;

console.log(a);

function c() {

a++;

console.log(a);

};

c();

};

b();

console.log(a);

}

a();

**Answer: 3,4,4**

----------

**3. what is the output of this function?**

(function() {

console.log(1)

setTimeout(function(){console.log(2)}, 1000);

setTimeout(function(){console.log(3)}, 0);

console.log(4);

})();

**Answer: 1,4,3,2**

----------

**4. Write jquery and change "Hello Kiran" => "Hello Kumar" for the below code:**

<div class="a1">

<div class="b">

Hello Dinesh

</div>

</div>

<div class="a2">

<div class="b">

Hello Kiran

</div>

</div>

**Answer:**

$('.a2 .b').html('Hello Kumar');

====

**5. What is promise?**

**6.Explain two way binding?**

**7. What is closure?(Important)**

**8.How you will share the data between controller?**

**9. What is scope and root scope?**

**10. what is watch in AngularJS and Explain it?**

1. write and call a function to increment value.

add() op:1

add() op:2

add() op:3

Answer :

function add() {

var count=0;

function increaseCount() {

count++;

}

return increaseCount()

}

add();

2. what is Promise. Explain it with example.

Answer :

var app = angular.module("myApp", []);

app.service('getService', function($http, $q)) {

var deffered = $q.deffer();

this.getMyName = function() {

return $http.get('http/....')

.then(function(response){

deffered.resolve(response)

},

function(response, status) {

deffered.reject(response)

});

}

});

3. How will you share data between controllers.

4. Element selector to select all three div which has custom attribute

<div cust-attr="val1">div1</div>

<div cust-attr="val1">div2</div>

<div cust-attr="val1">div3</div>

Answer : $("div:[cust-attr='val1']").

5. sum of any 2 numbers in an array should be 9

var arr [ 1 , 2, 5, 7, 10, 12]

sum = 9

function findIfSum(arr, sum){

var i =0;

var j = arr.length-1;

while(i < j) {

if(arr[i] + arr[j] == sum) {

return true

}

}

}