Asked interview

Interview architect

Ecommerce site

n/w traffic handle how users are more

black Friday how handle the users

global soft

observable and promise diff syntaxes tell

data binding

life cycle hooks

directives

didmount func diff

node js

diff in callback and callback hell

control flow function

roles and responsibility

tell me about yourself

sonata fullstack

observable and promises diff

ngoninit and constructor diff

rest and spread operator

life cycle hooks of angular

es6 features

Union

Diff structure and classes

Diff oops and rdbms

Angular

Javascript pro xx0

Persistent project manager

1 critical situation

Budgeting how to do

Resource management

Poor performer and client issue

Scenarios given of p.mgt

Describe imp project

Involvement of your in project

New gen soft

Why left so many companies

Where do you see in next 3 years

15 min only int took

Cursor asked

Factorial of number recursive function

All function need to do in javascript

angular

full desc resume

unifocus

factorial of number

sum of two no

sum(2,4) sum (2),(4)

map and foreach difference in array

hoisting

prototyping ex

object destructuring

spread and rest operator

php

mysql

transactions in mysql

interface and abstract class diff

cron job

update db from one db

oops concepts

join on and and condition

solid principle of php

(like abstract class)

laravel

batchservice

aws services what used

decorators in angular

php which version

standard class in php

whole interview( project worked, technology know, involvement in current prj

why so many companies in 10 years

int record for reference

interview what you see

birla soft 6th oct

clusters

javascripts

event bubbling

event emitter in node and angular

tell me abt your self

nodejs and angular

event handling in node

foreach and map difference

document and window difference

proc link

how to handle n/w load in ecommerce site

how to remove port no in url

javascript prog of array

how to get it

prj mgt

what is imp in project

why prj mgt is good for you.

Node works parllely or once execute code

n/w load how to handle

map and foreach diff in js

life cycle hooks

entry pointi n angular

how module.export work

exit codes in node

constructor and ngoninit

design pattern

subject and behavior subject diff

brisilio

how to handle scrum team

crisp

how to increase performance of team

what thing you do not want in team

syncrasy tech lead

asked abt jwt token

maches token or

large data how to optimize

technical documents need to make

apis error 400&500 and what is 429

2 people took interview

happy minds php developer

reverse routing

service container in laravel

yaml files

prevent default in jquery

define and constant diff

var-dump

interface and abstract class difference

copy function

mysql functions

views

mysql query

triggers

javascript

higher order func

computability python or javascript

reactjs rating

routing how work in react

settimeout,setimmediate,callback func which output first

Answer

**setTimeout** is simply like calling the function after delay has finished. Whenever a function is called it is not executed immediately, but queued so that it is executed after all the executing and currently queued eventhandlers finish first. setTimeout(,0) essentially means execute after all current functions in the present queue get executed. No guarantees can be made about how long it could take.

**setImmediate** is similar in this regard except that it doesn't use queue of functions. It checks queue of I/O eventhandlers. If all I/O events in the current snapshot are processed, it executes the callback. It queues them immediately after the last I/O handler somewhat like process.nextTick. So it is faster.

Basically, a function which takes another function as an argument or returns a function is known as a higher order function. Let's deep dive a bit to see both types of implementation, that is: Passing a function as an argument to another function. Returning a function from another function.09-Jun-2022