

MODULE 2: Node - NodeJS - Introduction

Que 1: What is Node.js? Where can you use it?

Ans: Node.js is an open source programming language. Its cross platform runtime environment used for development of sever side web applications. Node.js applications are written in JavaScript and can be run on wide verity of operating systems. Node.js can be use to build to different types of applications such as a command line application, web application, real time chat application, REST API server etc. it is mainly use to build network program like web servers, similar to PHP, Java, ASP.NET.

Node.js war written and introduce by Ryan Dahl in 2009.

Node.js official web site: <https://nodejs.org>

Node.js on github: <https://github.com/nodejs/node>

Node.js community conference <http://nodeconf.com>

Que 2: Explain callback in Node.js.

Ans: a callback is function which is called when a task is completed, thus helps in preventing any kind of blocking and a callback function allow other code to run in meantime. Callback is called when task get completed and is asynchronous equivalent for a function.

Example:

```
const fs=require('fs')

fs.readFile('text1.txt',function(err,data){

if(err)

    console.log("Err in this file: "+err);

else

    console.log("File Data -- "+ data);

})
```

Que 3: What are the advantages of using promises instead of callbacks?

Ans: Promises are used to handle asynchronous operation in JavaScript. They are easy to mange when dealing with multiple asynchronous operations where callback can create callback hell leading to unmanageable code.

Que 4: What is NPM?

Ans: NPM is Node Package Manager ans installer. It's a software registry. The registry contains over 8,00,000 code package. Npm is free to use. You can download all npm public software packages without any registration or login.

Npm include a CLI(Command line) that can be use to download and install software.

```
C:\>npm install
```

Que 5: What are the modules in Node.js? Explain\

Ans: modules in node.js is a simple or complex functionality organized in single or multiple javascript files which can be reuse throughout the node.js application. Each module in node.js has its own context. So it can not interfere with other module.

There are mane readymade modules are available in market which can be use in a node.js application.

Que 6: Create TODO app using FS module with yargs (command line arguments).

Ans: lets first create a module.

Example:

```
const fs=require('fs');

var log={
    info:function(info){
        console.log("My MSG for infomation"+info);
    },
    warn:function(warn){
        console.log("My MSG for warnning"+warn);
    },
    err:function(err){
        console.log("My MSG for error"+err);
    },
    rdfile:function(rdfile1){
        fs.readFile(rdfile1,function(err,data){
            if (err)
                console.log("Error in this file: "+err);
            else
                console.log("File date -- "+data);
        })
    }
}
```

```
};
```

```
module.exports=log;
```

Make an one program using this module in fs module with yargs.

Example:

```
var m=require('./Module1')

const yargs=require('yargs')(process.argv.slice(2)).argv;
const fs=require('fs');

const filename=yargs.filename, trans=yargs.trans, fdata=yargs.fdata
if(yargs.trans=="open")

  fs.open(yargs.filename,'w',function(err,data){

    if (err)

      console.log("Error in this file: "+err);

  })
else if(yargs.trans=="delete")

  fs.unlink(yargs.filename,function(err){

    if (err)

      console.log("Error in this file: "+err);

    else

      console.log(yargs.filename+" File delete successfully");

  })
else if(yargs.trans=="read")

  fs.readFile(yargs.filename,function(err,data){

    if (err)

      console.log("Error in this file: "+err);

    else

      console.log("File data is -- "+data);

  })
else if(yargs.trans=="write")

  fs.writeFile(yargs.filename,yargs.fdata,function(err,data){

    if (err)

      console.log("Error in this file: "+err);

    else
```

```
fs.readFile(yargs.filename,function(err,data){
  if (err)
    console.log("Error in this file: "+err);
  else
    console.log("File data is -- "+data);
})
})
else if(yargs.trans=="edit")
  fs.appendFile(yargs.filename,yargs.fdata,function(err,data){
    if(err)
      console.log("Error in this file: "+err);
    else
      m.rdfileread(yargs.filename);
  })
else
  console.log("Please give instruction carefully");
```