

Node Course Test Questions



NodeJS MongoDB Express Beginner to Intermediate JavaScript

NodeJS Express MongoDB connected together to update database content from Frontend Web Application using JavaScript

NodeJS MongoDB Express Beginner to Intermediate JavaScript	1
Section #1 Introduction to Installing and Setup of Node	2
Assignment Section #1	4
Section #2 JavaScript Fundamentals quick review of JavaScript Coding	6
Assignment Section #2	9
Section #3 JavaScript Asynchronous programming	10
Assignment Section #3	12
Section #4 Node Modules	13
Assignment Section #4	15
Section #5 Node Web Application with Express	16
Assignment Section #5	18
Section #6 MongoDB with Frontend web page interaction using Node and Express	19
Assignment Section #6	21

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>

More Courses content and answers <https://basescripts.com/> Laurence Svekis

Section #1 Introduction to Installing and Setup of Node

#1 What differentiates NodeJS and JavaScript?

- A. JavaScript is a programming Language and Node is an interpreter and environment for JavaScript
- B. JavaScript running engines are included in all browsers and Node uses the same running engine as Google Chrome
- C. both of the above
- D. none of the above

#2 Which is true about NodeJS

- a. It's a powerful framework developed on Chrome's V8 JavaScript engine
- b. Lightweight framework used for creating server-side web applications
- c. Can be used for large-scale application development
- d. All of the Above

#3 What makes NodeJS scalable

- a. the V8 Engine allow it to scale
- b. Server can respond in a non-blocking way using the event mechanism
- c. Node is Open Source providing excellent modules
- d. Not having to wait for an API to return data

#4 What are the types of API functions in NodeJS?

- a. Asynchronous, non-blocking functions and Synchronous, blocking functions
- b. Response and Request
- c. Request and Return
- d. None of the above

#5 How do you enter REPL environment within your terminal

- a. Type REPL

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

- b. Type node
- c. Type node REPL
- d. None of the above

#6 what will be output in the terminal when you type
\$ node app.js Laurence Svekis
with the code below

```
const first = process.argv[1];  
const second = process.argv[3];  
console.log('hi ' + process.argv[0] + second);
```

- a. hi /usr/local/bin/nodeSvekis
- b. hi Laurence
- c. hi Svekis
- d. none of the above.

#7 What does the -g flag represent

- a. -g is the global install flag
- b. -g is the directory
- c. -g is the dependency
- d. None of the above

#8 Which line of code will show the following in the terminal

(index)	a	b
0	1	2
1	5	10

- A. console.table([{a:1,b:2}, {a:5,b:10}]);
- B. console.grid([{a:1,b:2}, {a:5,b:10}]);
- C. console.log([{a:1,b:2}, {a:5,b:10}]);
- D. None of the above

#9 To install Nodemon globally what is the syntax using the npm

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

- A. sudo npm install -g nodemon
- B. node npm install -global nodemon
- C. npm install nodemon -global
- D. node install nodemon

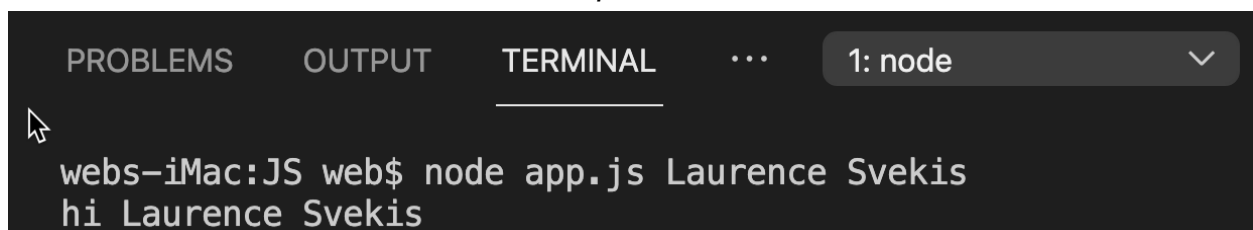
#10 What will the below code do? What will be output into the console?

```
for(let x=0;x<100;x++){  
  console.log(x);  
  if(x>9){  
    process.exit();  
  }  
}
```

- A. values 0 to 10
- B. value of 0
- C. value of 10
- D. value of 9
- E. values of 1 to 10
- F. values of 0 to 9

Assignment Section #1

1. Open the REPL mode and list the process object. Return the `process.ppid` and the `pid` into the terminal.
2. Run the node script with 2 arguments: your first name and lastname. Create a node application and output `process.argv[2]` `process.argv[3]` into the terminal like the example below.



```
PROBLEMS OUTPUT TERMINAL ... 1: node  
webs-iMac:JS web$ node app.js Laurence Svekis  
hi Laurence Svekis
```

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

Section #2 JavaScript Fundamentals quick review of JavaScript Coding

#1 Which of the below are JavaScript Data Types?

- Number
- String
- Array
- Value
- Boolean
- Object
- Undefined

#2 What is === operator?

- Conditional operator both the operands have the same value
- Strict equality operator both operands must have the same value without having to convert the datatype.
- Equality operator both the operands must have the same datatype
- None of the above.

#3 Which of the below are loop structures in JavaScript.

- for loop
- Do until
- Break and Continue
- while loop
- Do-While Loops
- For Break

#4 What is the result of this statement $7 * 2 + 10 - 1 + "1" + 0$

- "2310"
- 24
- "24"
- none of the above

#5 What would you see as the value when a variable used in the code doesn't exist?

- undefined

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

- null
- error message
- none of the above

#6 List 2 ways to return an object value. myObj[0]

- myObj.name
- myObj[0]['name']
- myObj["name"]
- myObj.['name']

#7 This method removes the last element from an array and returns the value

- pop()
- push()
- shift()
- unshift()

#8 This method adds a new element to the end of an array

- pop()
- push()
- shift()
- unshift()

#9 This method removes the first array element

- pop()
- push()
- shift()
- unshift()

#10 This method adds a new element to the start of an array

- pop()
- push()
- shift()
- unshift()

#11 How would you get each key next to each value from this object in the console?

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

```
const myObj = {  
  first : "Laurence",  
  last : "Svekis",  
  city : "Toronto"  
}
```

- for(let key =0;key<length;key++){console.log(myObj[key]) }
- myObj.loop({console.log(myObj.key)})
- for (key in myObj){ console.log(key,myObj[key]); }
- none of the above.

#12 Create a function to return a random number with parameters of min and max as arguments in the function.

- function ranNum(max) { return math.floor(Math.random() * max);}
- function ranNum(max) { return math.floor(Math.random() * (max-min));}
- function ranNum(min,max){ return math.floor(Math.random()*max)+min; }
- None of the above

#13 What is a promise in JavaScript

- Promise is a function that will run the code after the operation is ready
- Returns a block of code once the value is provided to the API
- Promise object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.
- None of the above

#14 Which keywords are used to handle exceptions in JavaScript?

- Try, Catch, Done
- Put, Get, Post
- Try, Catch, Finally
- None of the above

#15 Which loop structure will run the block at least once?

- While Loop
- do...while loop
- For Loop
- For Object Loop

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

- None of the Above

Assignment Section #2

1. *Create an array that can hold a list of your friends.*
2. *Use a JavaScript function to add a new friend as an object including first name and last name to the friends array.*
3. *Add more friends to your array so that you have several*
4. *Create a loop to output all the friends into the console listing their first and last name.*

Section #3 JavaScript Asynchronous programming

#1 How does NodeJS work?

- Node JS is a coding language
- Node JS uses a global object to run code
- Node.js works on the Google v8
- None of the Above.

#2 Is Node a Single Thread process

- True
- False

#3 Typically what is the first Argument passed to a NodeJS callback handler?

- Results
- Response
- Error
- Return
- None of the Above

#4 What is REPL (READ, EVAL, PRINT, LOOP) used for?

- Used to debug your application
- easy creation of Command Line Interface applications.
- Used for creating Files
- None of the above

#5 What order will the output be in the console?

```
console.log(1);
setTimeout(() => {
  console.log(2);
}, 0);
console.log(3);
```

- 1 3 2
- 1 2 3
- 2 1 3
- 3 1 2
- None of the above

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

#6 What order will the output be in the console?

```
console.log(1);
setImmediate(() => {
  console.log(2);
});
setTimeout(() => {
  console.log(3);
}, 0);
console.log(4);
```

- 2 4 3 1
- 1 2 3 4
- 1 3 2 4
- None of the above

#7 What does the event loop do?

- Stores and listens for Events
- Runs in the background to handle loops
- Allows NodeJS to perform non-blocking I/O operations
- None of the Above

#8 What does the event loop solve?

- Node.js overcomes the problem of blocking of I/O operations by using the event loop instead of threads.
- Solves the problem of capturing new event triggers
- In Node.js it performs the thread operations
- None of the Above

#9 What are the 3 states of Promises

- ready, result, return
- waiting, done, error
- pending, fulfilled, rejected
- get, post, delete
- None of the above

#10 Do you need to use require() to get the global object

- No you only need to return() it

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>

More Courses content and answers <https://basescripts.com/> Laurence Svekis

- No its always available
- Yes you need to request() the module
- Yes you need to import the package
- None of the Above

#11 What does EventEmitter do? Select all that apply?

- EventEmitter classes can use the `eventEmitter.on()` function to attach event listeners to an event.
- The EventEmitter module makes possible the communication and interaction between objects in Node.
- EventEmitter provides properties like `on` and `emit`.
- The `on` property is used to bind a function with the event and `emit` property is used to fire an event.
- EventEmitters can be used to trigger new events to run blocks of code.

#12 What is the output order in the terminal?

```
setImmediate(() => {
  console.log(1);
})
```

```
process.nextTick(() => {
  console.log(2);
})
```

```
console.log(3);
```

- 3 2 1
- 1 2 3
- 3 1 2

Assignment Section #3

Setup a custom event and emit the event into the terminal

1. *require the events module*
2. *setup a new events emitter*
3. *create a function the outputs a value into the terminal*
4. *Setup custom event emitter to listen for value. Add the function which will run the block of code. `eventEmitter.on(value,function);`*
5. *using emit select the value. `eventEmitter.emit(value);`*

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>

More Courses content and answers <https://basescripts.com/> Laurence Svekis

Section #4 Node Modules

#1 Which statement below is true about modules in NodeJS? Select all that apply?

- Each module in NodeJS has its own context and cannot interfere with other modules.
- You cannot create your own module
- Modules are like libraries that you can include into your application.
- None of the above

#2 Which of the below are examples of Builtin Core Common Modules in NodeJS?

- http , fs , path , dns
- url , fs , string
- fs , console , ftp
- util , fs , xml

#3 What is the keyword to make properties and methods available outside the module file?

- output
- return
- request
- exports
- response

#4 What is the command line to download a package called 'lsvekis' from npm

- node install lsvekis
- npm -g lsvekis
- npm install lsvekis
- none of the above

#5 What is the command line to get help with npm?

- npm help
- node help
- node npm help
- help -npm
- none of the above

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

#6 What is the fs module responsible for?

- fs module is responsible for all the asynchronous or synchronous file I/O operations
- fs module is responsible for file reading and file paths
- fs module is responsible for checking file errors
- none of the above

#7 How can you create a package json file in the directory?

- mkdir package.json
- mkfile package.json
- npm init -y
- node package.json
- none of the above

#8 How do you convert a buffer to a string value?

- buffer.from(str)
- Buffer.str();
- buffer.toString();
- none of the above

#9 What will be the output in the terminal if there is no value for val?

```
try {  
  console.log(val);  
}  
catch (err){  
  console.log('ERROR');  
}
```

- undefined
- undefined Error
- ReferenceError: val is not defined
- ERROR
- none of the above

#10 What line of code is missing to be able to use the code in this file as a module?

```
const person = {
```

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

```
first : 'Laurence',  
last : 'Svekis'  
}
```

- `module.export = person;`
- `export.person = person;`
- `export.person;`
- `exports.person = person;`

Assignment Section #4

Create a local module to use within your node application that holds some data and contains a function that can return results from that data.

1. create a new js file save it as data.js
2. create 2 variables in the file first and last for your first name and last name
3. create a function that can output and return an argument and concatenate the value of your first and last to the end of the string.
4. Export the output function
5. Require the data.js file as a module
6. Use the module function pass in a string value 'Hi, ' and see if it returns your first and last name from the variables in the data.js file

Section #5 Node Web Application with Express

#1 What is express

- Quick module to improve app speed
- Library for Node that adds new web pages
- Routing module for Node
- Fast minimalist web framework for Node.js
- none of the above

#2 Which are the arguments that are available to an Express JS route handler function

- return, callback, Next
- Prev, Next, Last, Done
- request, response, Next
- None of the above

#3 Which statement will serve files from a static folder named public

- `app.use(__dirname+'/public');`
- `app.path(__dirname+'/public');`
- `express.static(__dirname+'/public');`
- `app.use(express.static(__dirname+'/public'));`

#4 How do you Get Variables In Express.js In the Get Method?

- you cannot get the values only in post method
- In the response object as query object
- in the request object using `req.query`
- none of the above

#5 What will be output in the terminal when you go to the page <http://localhost:8080/?id=100> with the following code?

```
const express = require('express');
const app = express();
app.get('/', (req, res)=>{
  console.log(req.query);
  res.send('id: ' + req.query.id);
});
app.listen(8080);
```

- query

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

- id = 100
- { id: '100' }
- 100
- none of the above

#6. What does CRUD stand for

- Copy Read Upload Delete Methods
- Create Response Upload Delete Methods
- Create Read Update Delete Methods
- none of the above

#7 What does the body-parser module do?

- Returns the body of the request and outputs the values
- Parse incoming request bodies in a middleware before your handlers
- Sends handler information to the body of the request
- none of the above

#8 What does the body-parser module do?

- Is a core part of Express
- Is needed to read parameters
- Is only needed for JSON data
- Parse incoming request bodies in a middleware before your handlers

#9 What are core features of Express framework?

- Dynamically renders HTML pages
- Allows for setup of middleware to respond to HTTP requests
- Provides a routing table that can work with HTTP method and URLs
- All of the above

#10 Which of following are valid form of route path? Select more than one if applicable.

- `app.get('/info',(req,res)=>{ res.send('<h1>Info</h1>'); })`
- `app.get('/ab?cd',(req,res)=>{ res.send('<h1>ABC</h1>'); })`
- `app.get('/ab*de',(req,res)=>{ res.send('<h1>AB__DE</h1>'); })`
- `app.get('/z/,(req,res)=>{ res.send('<h1>ZZZZZZZZ</h1>'); })`

Assignment Section #5

Setup a frontend application that can send data to the backend. Use of both Get and Post methods to different routes in Express.

1. Create your express server run the express server
2. Add body-parser module middleware
3. Setup app.get route to return the value in the request query. Try within your browser to ensure you can get the value of id and send it as output to the page.
4. Setup app.post use the request body object req.body and return as JSON back to the client side
5. Setup the frontend code with a web page that contains a form that can Post to the node route for post. Add 2 input fields and a submit button sending the data to node express.

Section #6 MongoDB with Frontend web page interaction using Node and Express

#1 What is the syntax for the mongo database client in node

- `const MongoClient = require('MongoClient');`
- `const MongoClient = require('mongodb');`
- `const MongoClient = require('mongodb').MongoClient;`
- `const MongoClient = require('mongo').MongoDB;`
- none of the above

#2 What will the following code do if the url is the correct path with key to the database

```
MongoClient.connect(url,(err,db)=>{  
  if(err) throw err;  
  console.log('Connected');  
  db.close();  
})
```

- Update the last item in the database
- End the database session with close
- Show connected if a successful connection is made
- Throw an error and output the error
- None of the above

#3 Where are the captured values populated regarding route parameters?

- `req.body.value`
- `req.params`
- `res.body`
- `res.params`

#4 How can we create chainable route handlers for a route path in the ExpressJS app?

- `app.route()`
- `apps.routes()`
- `app.routes()`
- `app.routing()`
- `app.router()`

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

#5 How do you set the content type of the response to JSON?

- `res.JSON.stringify(data);`
- `res.json().data;`
- `res.contentType('application/json');`
- `res.contentType('JSON');`
- none of the above

#6 What is the syntax to install mongoDB in your application?

- `sudo npm mongodb -g`
- `node npm mongodb`
- `node -install mongodb`
- `npm -g mongodb`
- `sudo npm i mongodb`

#7 How would you connect to a database called people and select a collection called friends?

- `db('people').collections('friends');`
- `mongodb('people').collection('friends');`
- `client.db('people').collection('friends');`
- `mongodb.client('people').table('friends');`
- none of the above

#8 What is the syntax to create a collection in MongoDB?

- `client.db().create(name,opts);`
- `db.create.collection(name);`
- `db.createCollection(name,options)`
- `db.collection.create(name);`
- none of the above

#9 What is the syntax to drop a collection in MongoDB?

- `client.db().drop(name);`
- `db.drop.collection(name);`
- `db.collection.drop();`
- `db.collection.delete(name);`
- none of the above

#10 Which line of code is missing in the below example to make Express work

Test Questions Based on course at <https://www.udemy.com/course/nodejs-mongodb-express>
More Courses content and answers <https://basescripts.com/> Laurence Svekis

```
const express = require('express');
app.get('/', (req, res) => {
  res.send('Hello World!');
});
app.listen(8080, () => {
  console.log('ready');
});
```

- `const app = express();`
- `app.router();`
- `express.app();`
- `let app = router();`
- none of the above

Assignment Section #6

Creating a fully functional node application that uses Express to run a web application. Connect to a MongoDB and using Node allow the frontend web application to access the Database items, full CRUD interaction.

1. *Create a web application using Express - output html file from public folder*
2. *Setup a MongoDB in the cloud prep to connect using NodeJS*
3. *Create and test insert into the database*
4. *Setup Frontend web application*
5. *Add DOM event listeners to trigger requests*
6. *Generate visual response as HTML from the Node response object*
7. *Update and create interactive list of database content CRUD*
8. *Add HTML updates to send requests to the database.*