Benjamin Zhuang - Week 2 Day 4

1. What is Node.js?
    a. node .js is a single-threaded JavaScript runtime environment built of Chrome’s V8 engine. It’s capable of executing JavaScript code.
2. What are some differences between Node.js and JavaScript?
    a. Node.js is a runtime environment and JavaScript is a programming language.
    b. Node is used for backend development, while js can be used for anything including frontend development, and other general purpose programs.
    c. Node does not have built-in Web APIs, such as fetch(), XHR, storages.
3. What is npm?
    a. It’s node package manager, which is a CLI for accessing and installing packages from online packages created by other developers.
4. What is CommonJS? How does it differ from ES Modules?
    a. Commonjs is the project to implement a module ecosystem for JavaScript outside of the web browser.
    b. It uses the require() method to import modules, synchronously.
    c. It uses module.exports to export objects for the current module.
5. How do we import files into other files in Node.js?
    a. We use the require() method.
6. How can you make the server automatically restart when files are modified?
    a. We use the nodemon package or, potentially, some other packages.
7. What are some global objects in Node.js?
    a. The “global” object.
    b. Console
    c. process
8. Explain how the Node.js architecture is event-driven in terms of event emitters and other core modules.
    a. Event emitter is a module that facilitates the communication between objects. Basically, event emitters are a way to trigger events and event handlers are implemented by callback functions.
    b. For example in fs.writeFile, it takes in two arguments, the first is the path to the file that we’re writing, and the second is the callback function to act upon some event finished. In this case, the event is finished opening the file and is ready for reading/writing it. Now, we’re ready to make some changes to the file using the logic that we’ve defined in the callback function.
9. What are streams? In Node.js, what are the different kinds of streams?

   a. Streams are used to process large amounts of data by processing a smaller chunk at a time.
   b. There are 4 types of streams in node,
      i. Readable - can only read data from a particular data source.
      ii. Writable - can only write data to a particular destination/
      iii. Duplex - allows both read & write operations, however, the read & write operations are independent from each other. That said, one cannot read what's written to the destination with a duplex stream.
      iv. Transform - a duplex stream such that its readable and writable streams are connected by some transforms. So you can perform some read and then write data by transforming it, such as reading encrypted text and writing plain texts.

10. What is the difference between fs module's readFile and createReadStream?
    a. readFile loads the entire file into memory, while createReadStreams reads the entire file but only loads a chunk size of the file into memory.
11. What are the different timing or scheduling functions in Node.js?
    a. setImmediate, which schedules a function to be executed right after the I/O event callback from the event loop/
12. How does the event loop work in Node.js?
    a. The event loop handles asynchronous code by taking advantages from the kernel.
    b. Inside of the event loop, it contains a couple phases for each event loop iteration.
       i. **Timers**. The event loop will start by executing the timers and put the callback functions into the task queue. Those callbacks will only be added to the call stack once their timer countdown has been reached.
       ii. **Pending callbacks**. Execute the I/O callbacks that were deferred from the previous event loop iteration.
       iii. **Idle, prepare**. These are only used internally.
       iv. **Poll**. Retrieve new I/O events and executes their callback functions except close callbacks, callbacks set by timers, and setImmediate.
       v. **Check**. setImmediate callbacks are invoked here.
       vi. **Close callbacks**. Executes some close callbacks, such as socket.on('close', …).
    c. It also has a micro task queue that is higher priority. We schedule callbacks by process.nextTick(callback), which will be executed right after the current loop iteration completes.
13. How do you manage multiple node versions?
    a. Use the "n" npm package or use nvm.
14. Explain the request & response cycle.

    a. Browser sends GET a request to DNS.
    b. DNS resolves the server IP address and sends it back to the browser.
    c. Browser sends an HTTP request to the IP address.
    d. Server receives the request, processes it, and sends back information to the client’s IP address.
    e. The browser receives the request, parse the HTML file and render it.

15. Explain serialization vs deserialization.
    a. Serialization is converting an object to a stream of bytes.
    b. Deserialization is converting a serialized stream of bytes back to an object.
16. What is Express.js?
    a. Express js is a lightweight backend framework to build RESTful APIs.
17. How do you build a basic server in Node.js using express?
    a. We set up some middleware to transform data between the endpoints.
    b. We set up some routes for the client to make requests to and use some callbacks to process the request and send back responses.
    c. We specify the static folder for serving our static files.
    d. We start listening on a port indefinitely for any client requests.
18. What is middleware?
    a. Middleware is a software that facilitates communication between different applications.
    b. In Express, a middleware is some sort of request handler that has access to both the incoming requests and outgoing responses.
19. What are ports? Which one is reserved for HTTP and HTTPS?
    a. Ports at software level are used to identify connections with specific processes or applications/services. 80 is reserved for HTTP, 443 for HTTPS.
    b. Remember ports are ranged from 0 to 2^16 - 1 (65535).
20. What is the difference between response.send() and response.write() in express?
    a. Response.send sends back data to the client and can only be called once.
    b. Response.write appends some data to the body of the response that is going to be sent back to the client before res.end().
21. Explain query strings vs route parameters.
    a. Query strings is an object containing all of the query parameters key value mappings.
    b. Route parameters is an object that contains the mapped key value pairs of the route parameters in an URL.