


Explain and Reflect:

- Explain the differences between Java and JavaScript + node. Topics you could include:
 - that Java is a compiled language and JavaScript a scripted language
 - Java is both a language and a platform
 - General differences in language features.
 - Blocking vs. non-blocking
- *Java code is compiled to java bytecode on build. It is not run as it appears the screen. The bytecode can run on any machine that has a JVM to process the bytecode. This is why Oracle says “write once, run everywhere” because they have build JVM’s to almost all types of machines. Some other programming languages also compiles down to java bytecode (Kotlin, scala...)*
- *JavaScript, run in browsers and is dealt with in different ways from browser to browser. i.e. chrome uses a V8 c++ engine to produce machine code from JavaScript code. (vanilla)JavaScript is a less strict language and doesn’t have typecheck, it is also possible to mutate objects at any time, such as add properties to existing objects. JavaScript also does not offer compile time errors.*
- *Blocking refers to operations that block further execution until that operation finishes while non-blocking refers to code that doesn’t block execution. Or as [Node.js docs](#) puts it, blocking is when the execution of additional JavaScript in the Node.js process must wait until a non-JavaScript operation completes.*
- *Blocking methods execute synchronously while non-blocking methods execute asynchronously.*
- Explain generally about node.js, when it “makes sense” and *npm*, and how it “fits” into the node ecosystem.
- *Node.js is a JavaScript runtime engine outside the browser, it’s built on the chrome V8 engine. Therefore it makes sense to use for server side javascript app’s. Node.js ships with the “npm” which helps downloading and including external code, to use for i.e. password hashing, or a server like express.*
- Explain about the Event Loop in JavaScript, including terms like; blocking, non-blocking, event loop, callback queue and "other" API's. Make sure to include why this is relevant for us as developers.
- *The node.js event loop system is a way to make a single threaded engine run as it was multithreaded. An event loop is looking for events in the event queue, and dispatches the events to the handlers (e.g. a .then()) in the program. This makes asynchronous code possible.*
- What does it mean if a method in node's API's ends with **xxxxxxSync**?
 - *It means that the method is not async and therefore would block the js-event-queue*
 -
-  Explain the terms JavaScript Engine (name at least one) and JavaScript Runtime Environment (name at least two)

- *chrome uses a V8 c++ engine to produce machine code from JavaScript code. The engine works inside an environment (eg node / chrome) which provides features to your scripts that can be used at runtime, for example utility libs or APIs.*
 - Explain (some) of the purposes with the tools *Babel* and *WebPack* and how they differ from each other
 - *Webpack can pack a big frontend app into few “condensed” files, while doing that the js code will be minified, and other options are available as well e.g. obfuscating (unreadable code), uglify etc.*
 - *Babel is a transpiler, that is compatible with all modern versions of ES and will translate into old wide compatible ES code supported by most browsers.*
- When using create-react-app both webpack and babel will be configured for you, just run the command: npm run build, and it will build an optimized version of your app in a “build directory”*