4)What is Node?

Node came about when developers took javascript to something you could usually run inside of browser and they let it run on your machine as standalone process. This means that we can create application using javascript outside of context of browser. Javacsript previously had a limited feature set, but with nodewe now have feature set that looks much more similar to other languages like Java, Python or PHP. We can create node applications using the javascript syntax that can manipulate your file system, creating and removing folders. They can query db directly, you can even create web servers using node.

Now both node and js that get executed inside of browser, are both running on the same engine. Its called V8 javascript runtime engine. It’s an open source engine that takes javascript code and compiles it into much faster machine code. That is big part of what makes nodejs so blazing fast. Machine level is low level code that your computer can directly without needing to interpret it. Your machine only knows how to run certain types of code, for ex your machine cannot run js or php code without first converting it into something, it does know. Because v8 engine does this so well. Node is just super quick.

V8 engine is written in language called c++. So if you wan to extend node language you don’t write node code, you write c++ code that builds off what v8engie has in place.

Open cmd, type – node and press enter. Now you can execute commands directly here.it is equivalent to executing commands on console in browser. Lets see difference bettwen js running on browser and node. In js on browser we have global object called window, it stores basically everything that you have access to. Essentially every variable that you create lives inside window. We have something similar inside of node, we have global. It stores a lot of same things. You can see methods like setTimeout and setInterval. We have most of things that are defined over inside of window with some exceptions.

Inside our browser I also have access to document. Document stores a refrence to DOM up above. The document shows exactly what I have up here inside of browses viewport. I can make changes to the document down below to update what its showing up above(in bowser). Obviously we do not have HTML document inside of node but w edo have something similar. What we have is called process. Type node , then press enter. Then type process. Now we can view process. here we have lot of information about the specific node process that’s being executed. There is also methods availaible to shut down the current node process. Enter-

process.exit(0)

here we say that things exited without any error. Now we are out of node. So that is difference at high level-

window becomes global. Document becomes process.

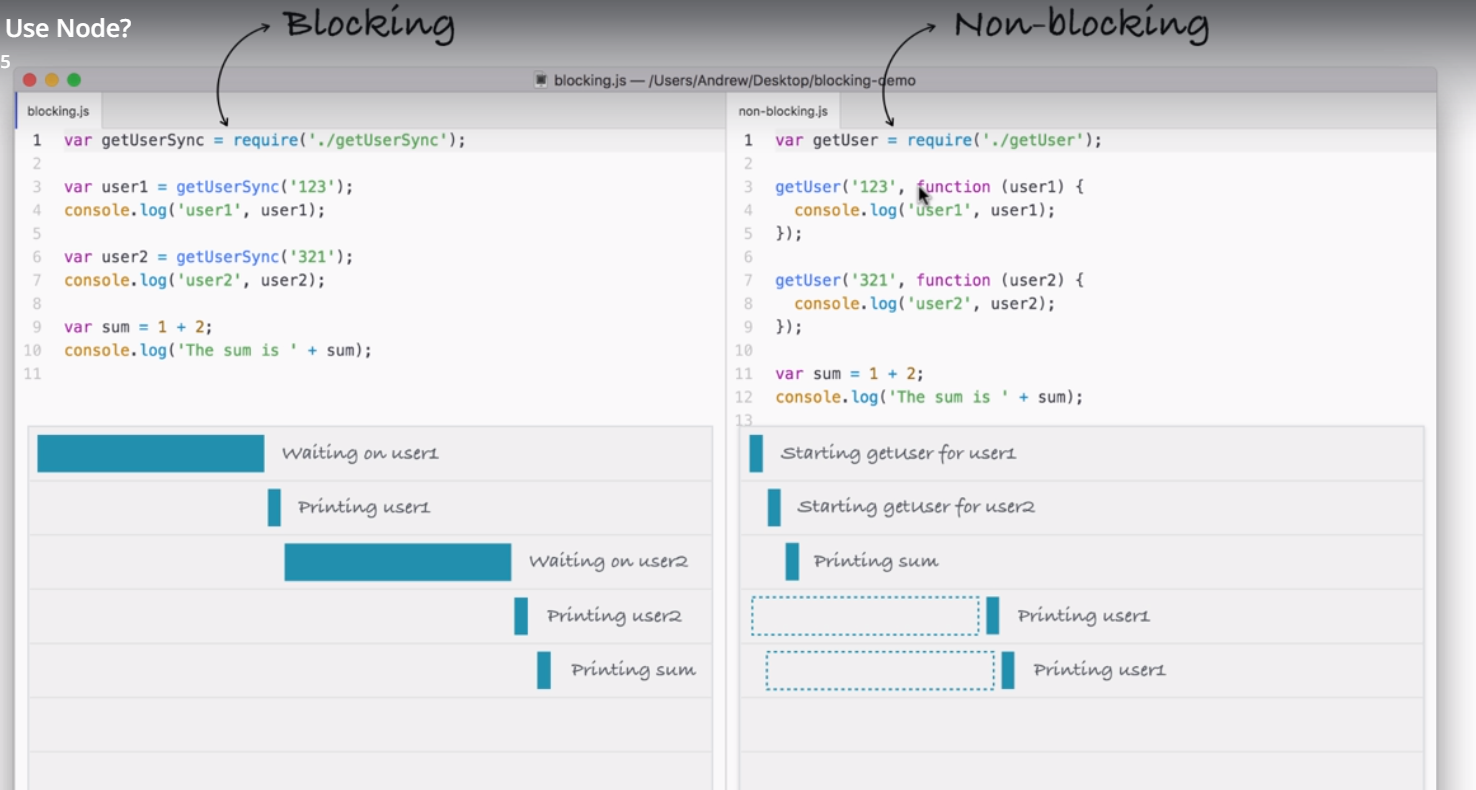
So node is javascript run time that uses V8 engine and when someone ask you what the V8 engine is you can say that the engine is open source javascript engine written in c++ that takes js code and compiles it to machine code. Is is used inside of nodejs and it is used inside of chrome browser.

5)why should I use node

These lines are from nodejs website-

**Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem,**[**npm**](https://www.npmjs.com/)**, is the largest ecosystem of open source libraries in the world.**

i/o is reading or writing to database.it is short form of input output. This is communication from your node application to other things inside of internet of things. This can be changing files on file system, database read/write, making http request to separate web server. IO takes time. Non blocking IO is great. That means when one user is requesting a URL from google other user can be requesting database file. they can be requesting all sorts of things without preventing anyone from getting some work done. d/w blocking and non blocking software development-



Using non-blocking model we can still do stuff like printing the sum without having to wait for our database to respond. Ths is big difference b/w 2. In blocking everything happens in order and in non-blocking we strat vents attaching callbacks and these callbacks get fired later. We are still printing user 1 and user2 , but we are doing it when data comes back. But data does’nt come back right away. Inside of nodejs event loop attaches a listerner for event to finish in this case. When that event finishes that callback is called.

Imagine this was web server instead of current example, that would mean if a web server comes in looking to query the databse , we can’t process other users request without creating separate thread. Nodjs is single threaded, which means your application runs on single thread. But since we have non blocking IO that is not a problem. In blocking conectext we hand handle 2 request on two separate thereads. But that does’nt scale well because for each request we have to beef up the cpu and ram resources that we are using for application. And this sucks because those threads are just sitting idle, just because we can spin up other threads we should not. We are wasting resources that are doing nothing. here instead of wasting resources by creating multiple threads, we are doing everything on one request. When request comes in, the I/O is non blocking.

This explains the statement-

Node.js' package ecosystem, [npm](https://www.npmjs.com/), is the largest ecosystem of open source libraries in the world.

**Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.**

So by using non blocking io we were able to cut down the time our application took by half.this non blocking IO makes our apps super quick. That is where lightweight and efficient comes into play. Last statement is-

**Node.js' package ecosystem,**[**npm**](https://www.npmjs.com/)**, is the largest ecosystem of open source libraries in the world.**

This is cherry on top. The community, the people every day developing new libraries that solve common problems in your nodejs applications, thibgs like validating objects, creating servers and serving up content live using sockets. There are libraries alreday built for all of those. You do not have to worry about these, this means you can focus on things speciific to your application.

Go to npm site. If your problem is generic, chances are that someone has already solved it. Lets say you want to validate a object that name property exists and it has id of length 3. then google it- npm object valiadator.

This is what node developers do, they take advantage of fantastic community of developers.

I am not saying that you should not use blocking languages like python, php. I just showed the power of non blocking. Languages like python have thibgs like library twisted that aims to add non blocking. Feature to python though big problem is all third party libaaries are still written using blocking code. So you are kimited to which libraries you can use. Since node js was built non blocking from gorund up, every single library on npm is non blocking.