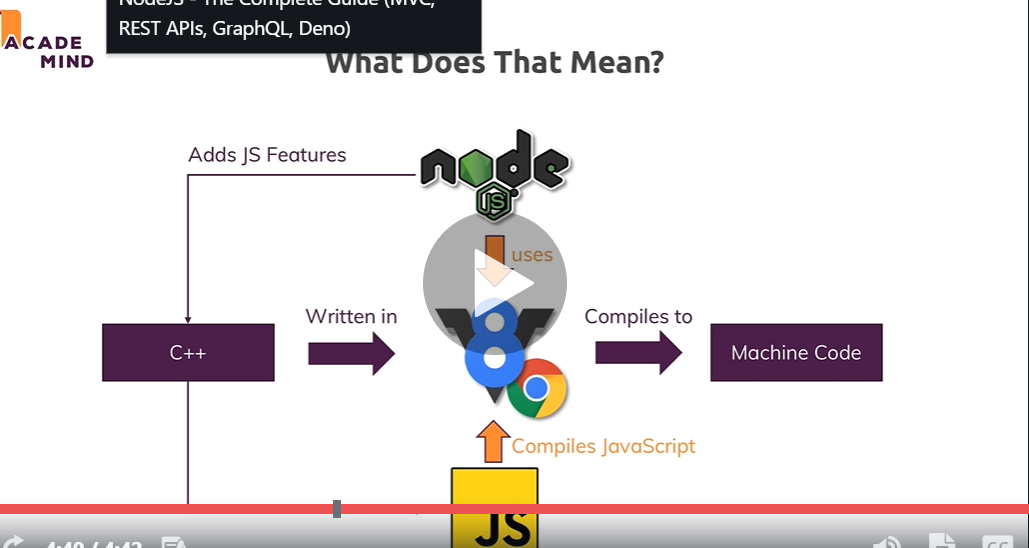
**Lecture 02**

**What is Nodejs?**

* Nodejs is a javascript runtime and now what does this mean? You know javascript, it's a programming language you typically use in the browser to manipulate your dom, to manipulate the page which was loaded in the browser.
* Javascript is a language that runs in the browser that allows you to interact with the page after it was loaded and it therefore is a crucial part when it comes to building interactive user interfaces in the browser
* However javascript is not limited to that. Nodejs is a different version of javascript you could say, it is basically built on javascript, it adds some features to it, is not capable of doing some other things you can do with javascript in the browser, so it basically takes javascript and puts it into a different environment.
* it basically allows you to run javascript not just in the browser but anywhere else like a normal programming language.
* we can use nodejs to run javascript outside of the browser, that is the core takeaway
* nodejs uses v8 and v8 simply is the name of the javascript engine built by Google that runs javascript in the browser
* Well it simply means that engine takes javascript code, the code running in your browser then or in node's case which builds up on v8, also the nodejs javascript code, it takes that javascript code and compiles it to machine code and this is what your browser does too, what v8 does in your browser. It does take your javascript code and compile it to machine code because that is the code that runs ultimately on your computer and that can be handled efficiently.
* v8 itself is written in C++
* but nodejs basically takes that v8 codebase which is written in C++ and adds certain features like for example working with your local file system, opening files, reading files, deleting files, these are all things which are not possible in the browser, you can't access your local filesystem in the browser for security reasons
* It allows you to run javascript on your computer and it adds useful functionalities to the javascript engine so that you can do more useful stuff there than you can do with browser side javascript. Now one important note maybe on this point also is that of course some features are also taken away. In the browser you use javascript to interact with the document object model, so with the html elements on your page, if you just execute a javascript file directly, you of course have no attached page and therefore these features are missing



**Lecture - 04**

**Installing Nodejs and Creating First App**

* require() 🡪 used to import modules.

**Lecture - 05**

**Understanding the Role and Usage of Node.js**

* . Now on the server, we typically do tasks that we can't or don't want to do from inside the browser for performance or security reasons. We connect to databases for example to fetch and store data. We do user authentication which we obviously can only do on a place the user can't access to make it more secure and avoid it being hacked. We do it for input validation to see if a user entered a correct e-mail address, the browser can always be tricked, users can even edit their browser side code. You can open the developer tools and start working on that page you're on but the server is of course sheltered from that, the user can't access it
* nodejs is not limited to running code on a server, it's a javascript runtime and you even saw a first demo which did not do anything where we needed a browser right, we didn't spin up a server there, we didn't do anything which we would have reached through a browser.
* you also often use nodejs for other code, for example for local utility scripts or build tools. If you worked with let's say react or angular or vue or anything of that kind, you actually used nodejs indirectly a lot for all the build processes these languages or frameworks needed because nodejs is a great tool for writing utility scripts. You have access to the file system so you can write and read and manipulate files and this allows you to do a lot of utility stuff on your computer that is never exposed to the public
* So you use it to run a server and actually and that is an important difference to PHP for example, with nodejs you don't just write the code that is running on your server, you also write the server yourself, so the code that takes the incoming requests and routes them to your well other code. In PHP, you have extra tools like apache or nginx which run the servers which listen to incoming requests and then execute your php code, here nodejs does both. . It does that listening and it then also does whatever you want to do in your code
* we therefore also use it to run all our business logic, so not just to listen to incoming requests but to then work with the requests data, work with files, work with databases
* we also handle the response side not just incoming requests, you will also learn how you use nodejs to send back data to your clients, be that html pages, html pages with dynamic content or data only in the format of json or xml or even files.
* Alternatives to nodejs would be things like Python, also with frameworks like flask or Django or PHP with frameworks like laravel maybe or standalone vanilla PHP of course and more, asp.net, Ruby on Rails, all that stuff, these basically are all replacements for nodejs or nodejs can be a replacement for them and there is no clear winner
* . All these languages are capable of doing the same kind of stuff and of course they differ in some technical regards but in general, it's great to have that broad variety
* The huge advantage or one huge advantage of nodejs is that it uses Javascript, a language which you need so much in modern web development for all the frontend, for some build tools and if you then can also use it on the server side, you don't need to learn a bunch of different languages, you can use one and the same language and then use that for your server side code too.
* , it's also a highly performant and popular language

