Introduction to Node.js

By Matthew Honour

What exactly is Node.js?

- Node.js is a runtime environment for JavaScript.
- Node.js is not a language.

What is a Runtime Environment?

- A runtime environment is simply an environment that a programming language can be executed in.
- Until 2009 the only runtime environment available for JavaScript was in the browser.

How do browsers run JavaScript?

- Browsers run JS using their respective JS Engine.
- Some example include:
  - Chrome: V8 Engine
  - Firefox: SpiderMonkey
  - Microsoft Edge: Chakra

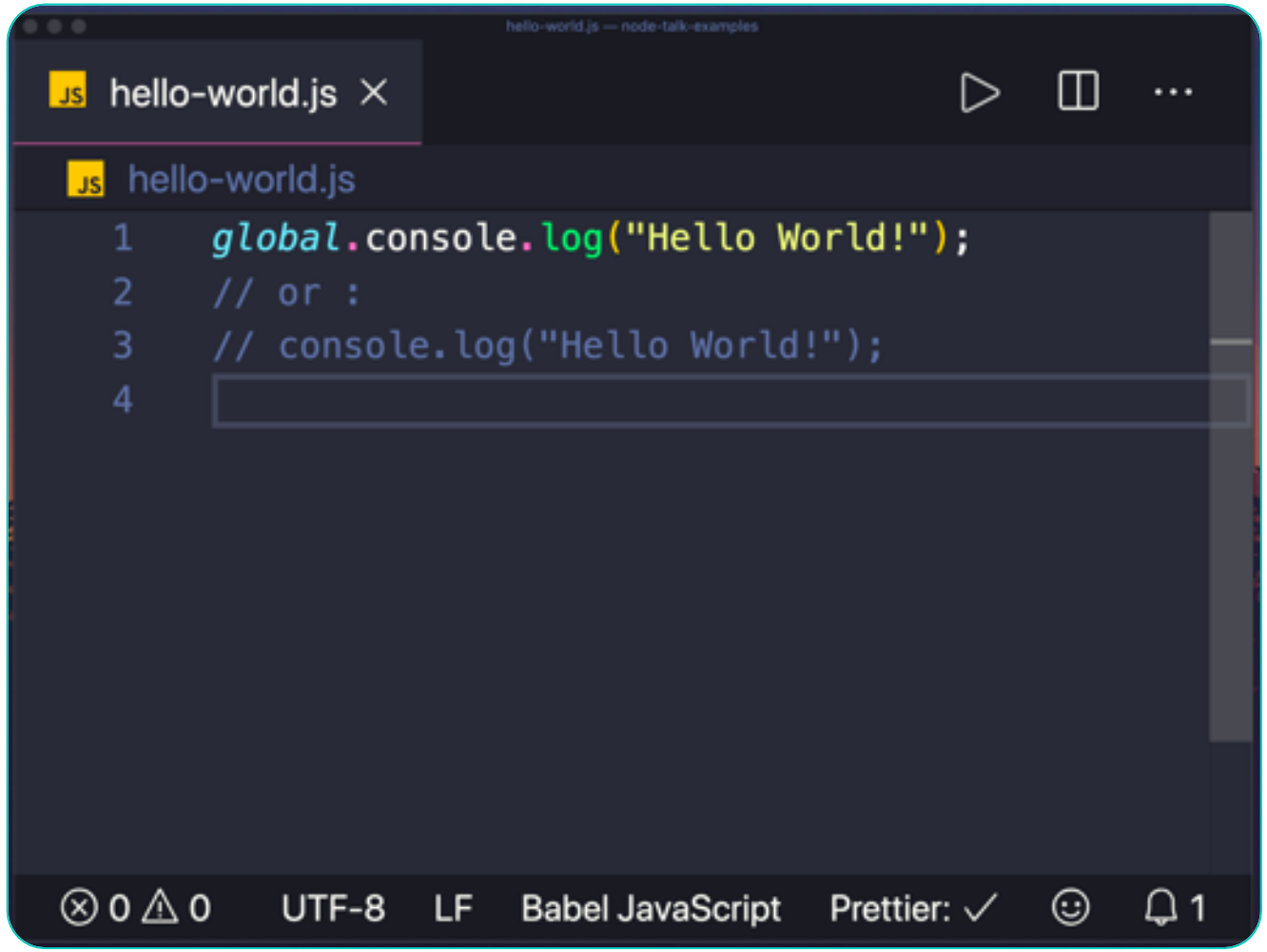
How does Node.js run JavaScript?

Node uses Chrome's 'V8 JavaScript Engine' to execute JavaScript Code.

What else does Node.js do?

- Node.js adds several useful libraries that developers can use to write server-side applications. Some of these libraries include:
  - Path – working with file paths
  - Os – working with the Operating System
  - Http – used to make web servers and transfer data over http.
  - Fs – working with the file system of the machine.

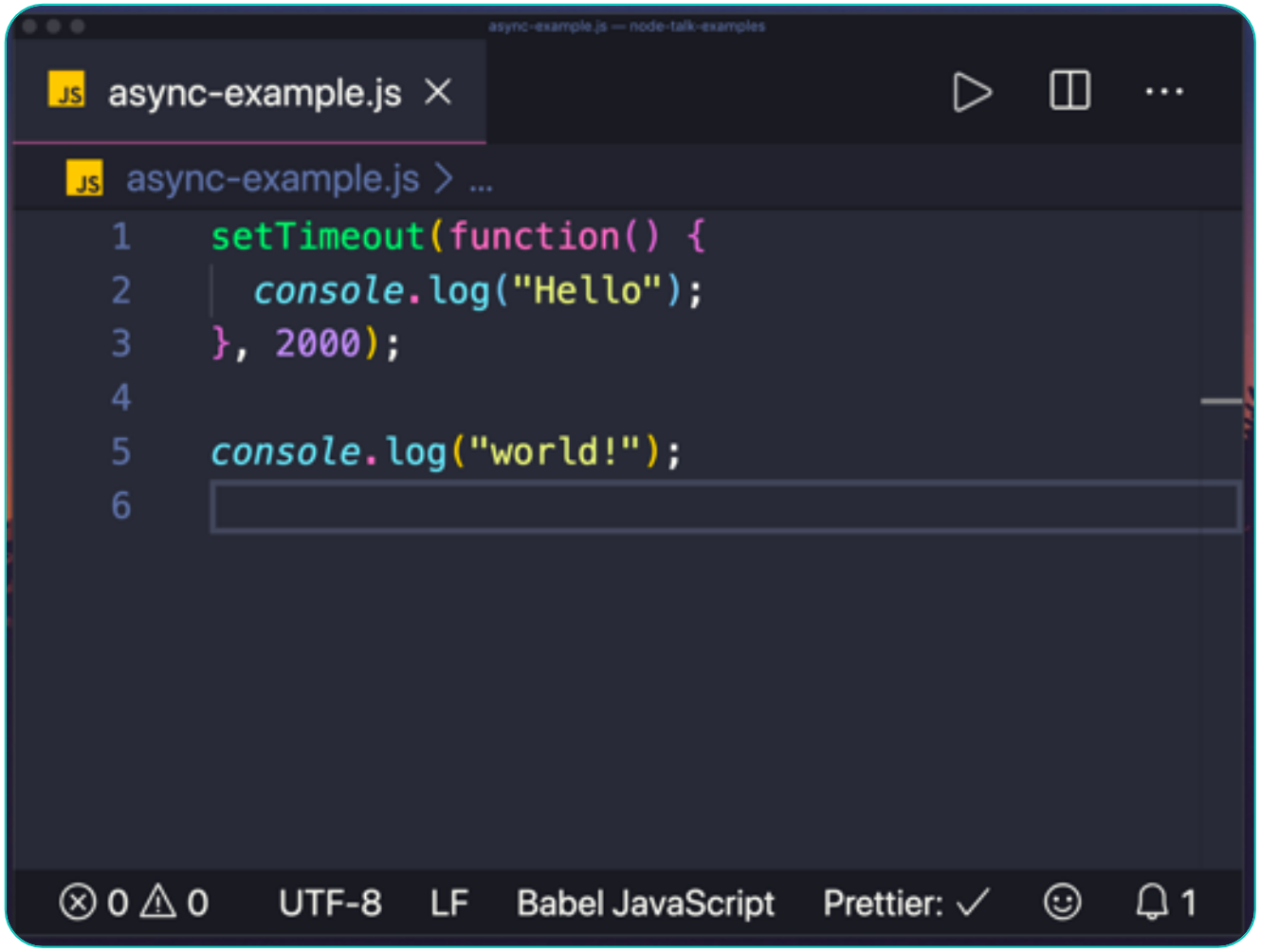
Let's start with a classic Hello World!



Asynchronous by Default

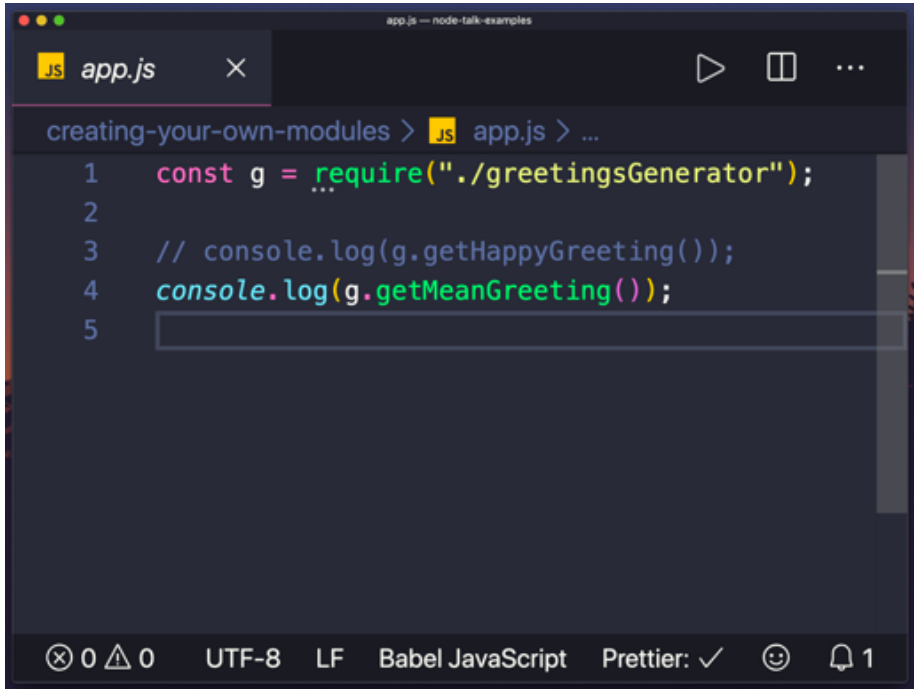
- All JavaScript code executed using Node.js is asynchronous by default.
- This means that Node.js never waits for a line of code to be executed...
- … Instead, it proceeds on to the next line of code while the previous line is being executed in the background.
- This makes Node.js perfect for programs that contain a lot of I/O operations (e.g. reading and writing to a database).

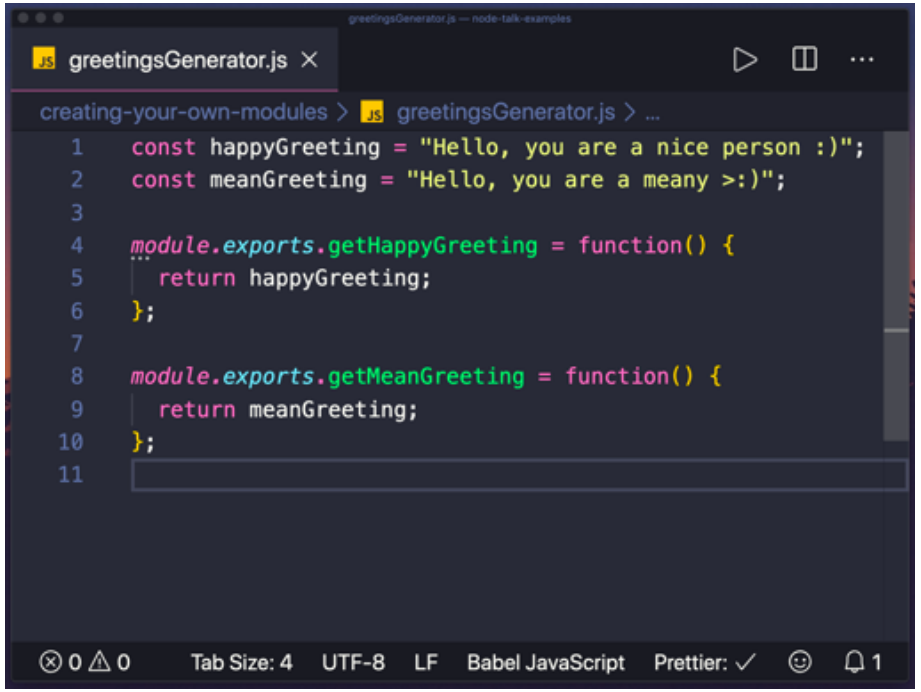
Async Example



Modules in Node.js

- In Node.js code is organised into modules.
- Each JS file within a node project is considered a module and it is the way that Node.js deals with organising code.
- There are three types of module in Node.js...



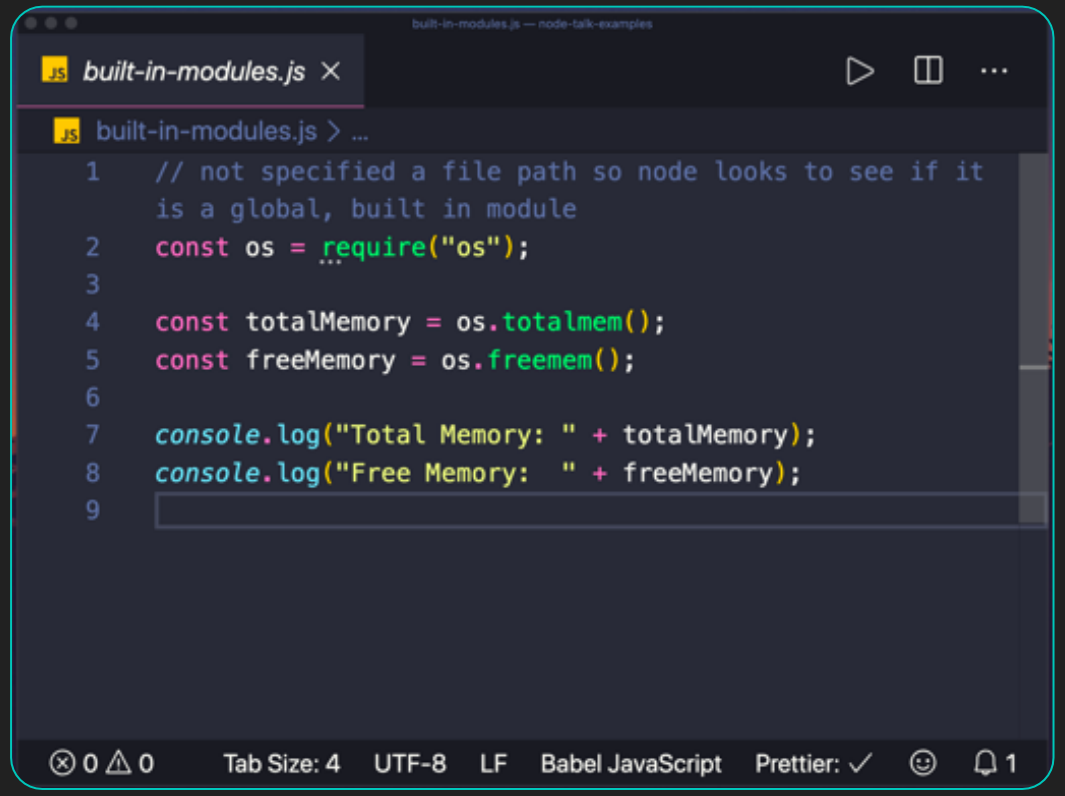


...Your own modules

Each File that you make is a module and can be used within other files.

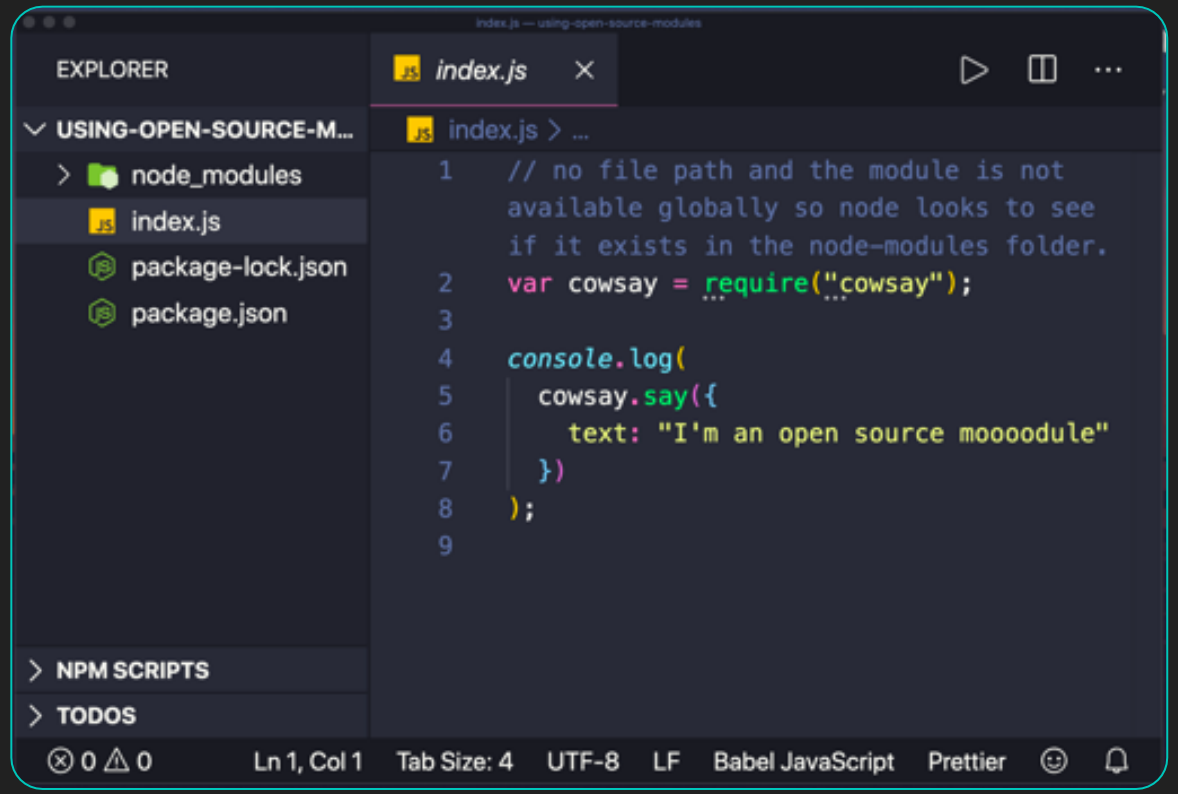
...Node.js Core Modules

- These are the modules that are globally available in a Node.js application.
- ...We mentioned some earlier like: path, os, fs and http.



… Open Source Modules

- Lucky for us developers there are thousands of modules already created for Node.js.
- These are stored in a repository and made available via the 'Node Package Manager' (npm).
- Npm is a command line tool that you can use to add open source packages to your Node.js project.
- Here it is: https://www.npmjs.com/



When to use Node.js?

- As aforementioned, Node is good for I/O operations like accessing databases.
- Due to its asynchronous nature, it can deal with many client requests without making each user wait until the previous request is resolved.
- This makes Node.js perfect for creating RESTful APIs.

When not to use Node.js?

- Node is single threaded.
- This means that for any tasks that require multi-threading to make full use of the CPU:
  - e.g. Anything that we ever did with Marius last year...

  Use another language :D! I hear C++ is a good one !

To Conclude...

- Node.js is the best :D

- Ps. I accept tips!

Any Questions?