Node Introduction

hello world, REPL, NVM

Lesson Plan

- About node and JS
- NVM
- REPL
- Hello World
- Javascript - variable declaration, and data types

Summary of this lesson can be found here:

https://beta.nerdeez.com/articles/node/introduction

What is Node.JS

- Javascript runtime environment
- Combines Chrome V8 JS engine alone with event loop on a c++ project which allows to run javascript outside of the browser
- Node is cross platform and runs on windows, linux and mac.
- With node we can now do what we can do with other programming language
  - Create Desktop applications
  - Create Server application for dynamic web pages
- Node strong suit is for creating server applications
- Node is event based
- Node is really easy to learn and has very good performance in I/O tasks
  - file system, networking, db query...
- Node is less performant on CPU intensive tasks
  - image processing, cryptography…
- Another strong suit of node is how easy it is to run async tasks.

What is Javascript

- Programming language
- Object Oriented language
- Single Threaded*
- Event driven
- Dynamic
- Interpreted
- In node the v8 engine is in charge of transforming JS commands to machine action
- Really easy to use async code

What is the Event Loop

- Node is an event driven platform
- events are async
- The event loops orchestras the async events
- event loop is in charge of sending actions to the c++ part
- the c++ part is in charge of sending the action either to the kernel or deal with it in one of the threads, after action returns it sends the response to the event loop
- the event loop is in charge to run the callback

Install node

- Verify the install by typing:
  - **> node -v**

REPL - Node CLI

- sometime referred to as node cli
- Read Eval Print Loop
- we can send a js command to it, and it will use node to evaluate that command.
- To run:
  - **> node**
- To run a file
  - **> node <js file>**
- We will use this method to run our js programs.

REPL - more commands

- tab - autocomplete and suggestions
- **.editor -** enter multiline code
- **.save <filename>**

Hello world

- lets create a node program that does the following
  - define 3 variables with hello world strings
  - print those variables to the console

Javascript break - variables, types, functions

- lets go over the js code in the hello world program we just typed.

IDE and Debugging

- Recommended IDE (**jetbrains - webstorm, pycharm**)
  - (Authors opinion)
- In this course the code examples will be either with the cli or with Visual Studio Code (free to use)
- Lets try and add a breakpoint in our hello world program using visual studio code

NVM

- It’s important to maintain a consistent node version on a project between all team members (WHY?)
- NVM is used to manage multiple node environments
- We can place a file called **.nvmrc** specifying the version of node
  - v9.1.0
- **> nvm use**
  - will download and install the proper environment based on the **.nvmrc** file
- Recommended to automatically set the terminal to use **nvm use** if there is an **.nvmrc** file in the folder.

Student EX

- Install node
- install nvm
- add the bash script to automatically do **nvm use** if the **.nvmrc** exists
- install IDE of your choice
- create a js script which does the following
  - create a variable holding array of strings
  - iterate over that array and print every string in the array
- run the script
- try and add a breakpoint to that string

Summary

- First lesson and we are already running js scripts with node.
- Next lesson we are going to write more complex code with node and understand about the module system in node