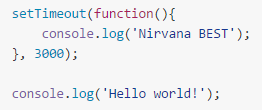
**Node.js** is **single-threaded** and **asynchronous**



Hello world first.. then the other..

Program certainly does NOT STOP AND WAIT..

In Node..

Program starts to execute setTimeout... it will **tell it to wait for 3 sec..**

Node.js now has a **TASK THAT WILL BE RUN AFTER 3 SEC**..

Continues running.. after 3 sec.. will refceive **interrupt** and when it receive this it will run it..

**Node.js** will **ALWAYS CONTINUE TO RUN.**

You define **TASKS** and **CALLBACK** but node.js will always continue to run.

You can try using setTImeout or a callback.. but node.js will not stop... **BECAUSE IT IS SINGLE-THREADED**..

Let‘s assume we have single threaded and NOT asyncrhnous.. **WHEN READING FROM DATABASE FOR ONE CLIENT, ALL OTHERS WAIT...**

Node.js can handle this because it is asynchronous:

* Node SENDS REQUEST TO DB
* While it waits: HANDLES OTHER CLIENTS REQUESTS
* When DB finished.. Node gets INTERRUPT... receives the data and gives the first client that data..

To write HTTP service.. var http = require(‚http‘);