

INTRODUCTION

DEVELOPING WEB APPLICATIONS
ACIT 2520

INTRODUCTION

- Instructors for this course are: Michał Aibin and Neda Changizi
- Emails: maibin@bcit.ca / nchangizi@bcit.ca
- Office hours - [http://timetables.bcitbusiness.org/
ComputingIndex.aspx](http://timetables.bcitbusiness.org/ComputingIndex.aspx)
- Office hours location - DTC-617
- Private cellphone numbers - +17... (you wish we give them to you)

HOW TO SUCCEED IN 2520?

- Be patient
- Be systematic
- Ask for an opinion, ask for a help, and never give up
- Don't copy-paste the code samples that you find on the Internet
- Don't copy others work

PERFORMANCE EVALUATION

- 20 points for the challenges in the class
- 10 points for the quiz in the class (max 2 points per class)
- 20 points for the homework - 2 homework assignments, 10 points each
- 10 points for the midterm - multiple answer test (minimum 3 points required to pass)
- 20 points final project - in groups of 2-3 people
- 20 points for the final exam - test, at least 5 points at the final exam
- **100 points maximum, 50 points to pass**

GRADES (JUST FOR OVERVIEW)

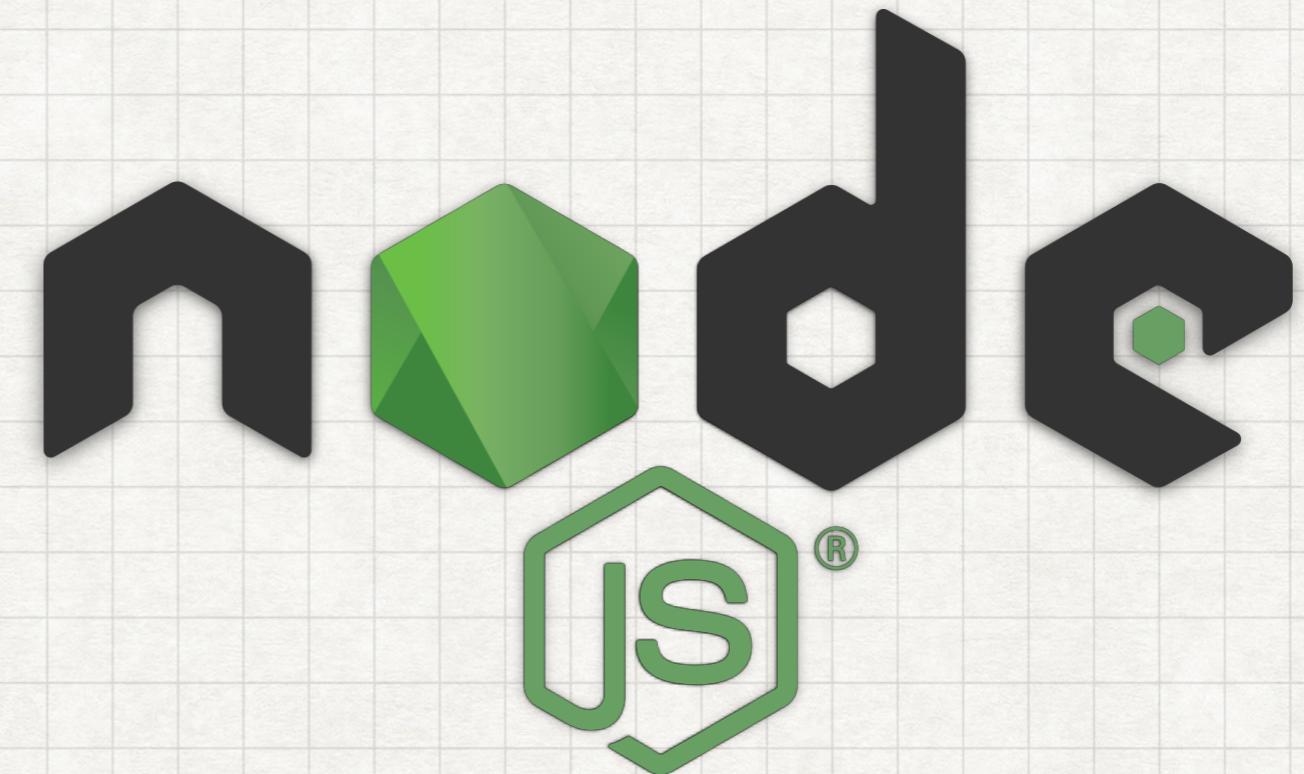
- 100%-95% A+
- 94%-90% A
- 89%-85% A-
- 84%-76% B+
- 75%-70% B
- 69%-65% B-
- 64%-60% C+
- 59%-55% C
- 54%-50% C-
- <50% Failed

QUESTIONS?

LET'S START!

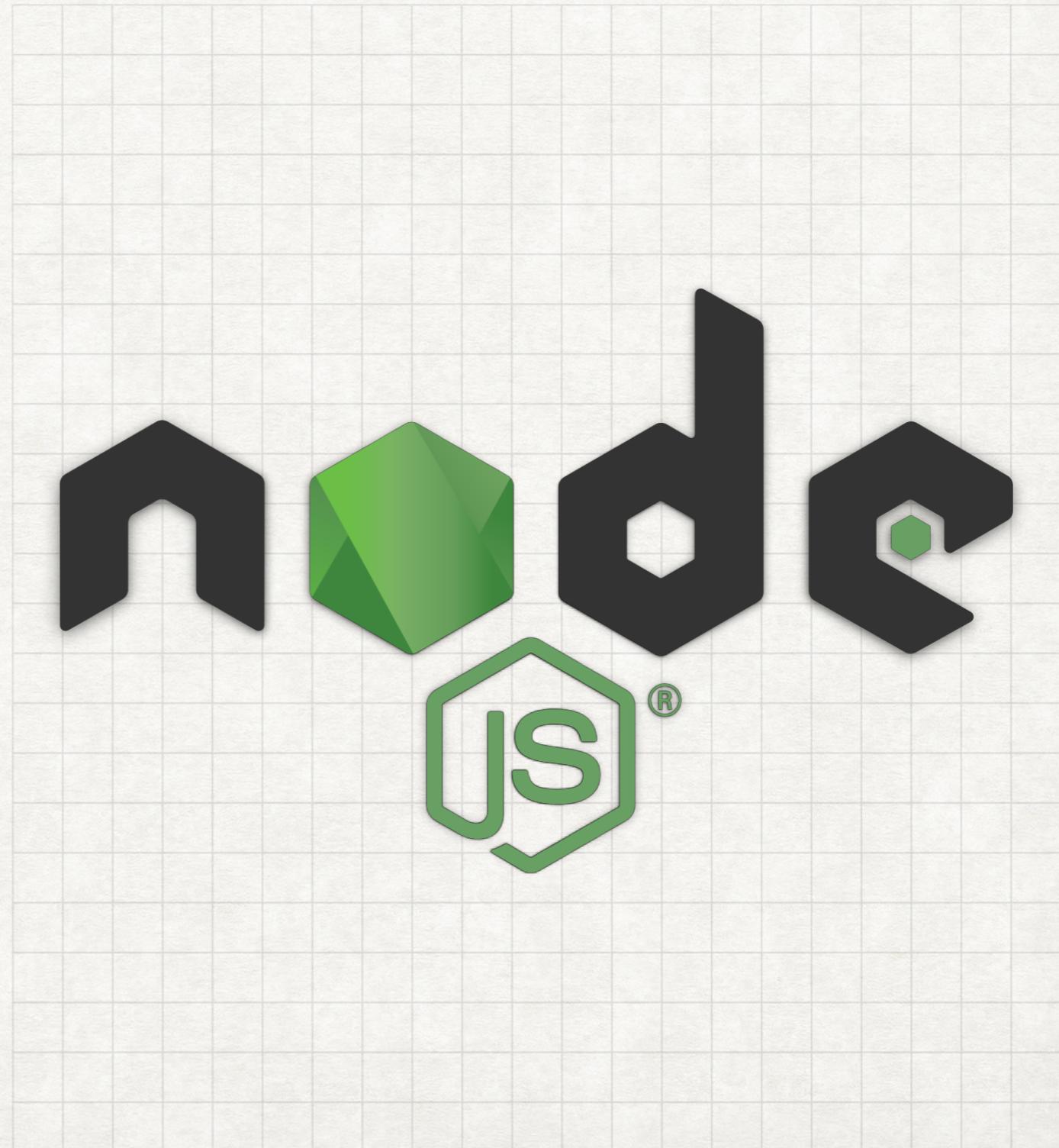
WHAT IS NODE.JS?

- Framework based on JavaScript to build web applications
- Uses V8 Engine from Chrome



HOW TO START?

- First, we need to install Node.JS in our system (assuming that you don't have one)
- The complete guide is available in the official docs: <https://nodejs.org/en/download/>
- We are going to the latest, BETA version of Node.JS to support all the newest features (at the time of writing this tutorial, it was 10.x)



BREAKPOINT

HOW TO START?

- Once installed, we need to verify that everything went well, by running the command below

```
[MacBook-Pro-Micha-2:~ Michal$ node -v  
v10.2.1  
MacBook-Pro-Micha-2:~ Michal$ ]
```

RUN IN THE BROWSER

- We can run our commands in the browser or in the command line, by simply writing:

`node`

- Let's write a simple `console.log()` message

- Next let's check the usage of: `process.exit(0);`

```
[MacBook-Pro-Micha-2:~ Michal$ node -v
v10.2.1
[MacBook-Pro-Micha-2:~ Michal$ node
[> console.log("Test!");
Test!
undefined
[> process.exit(0);
```

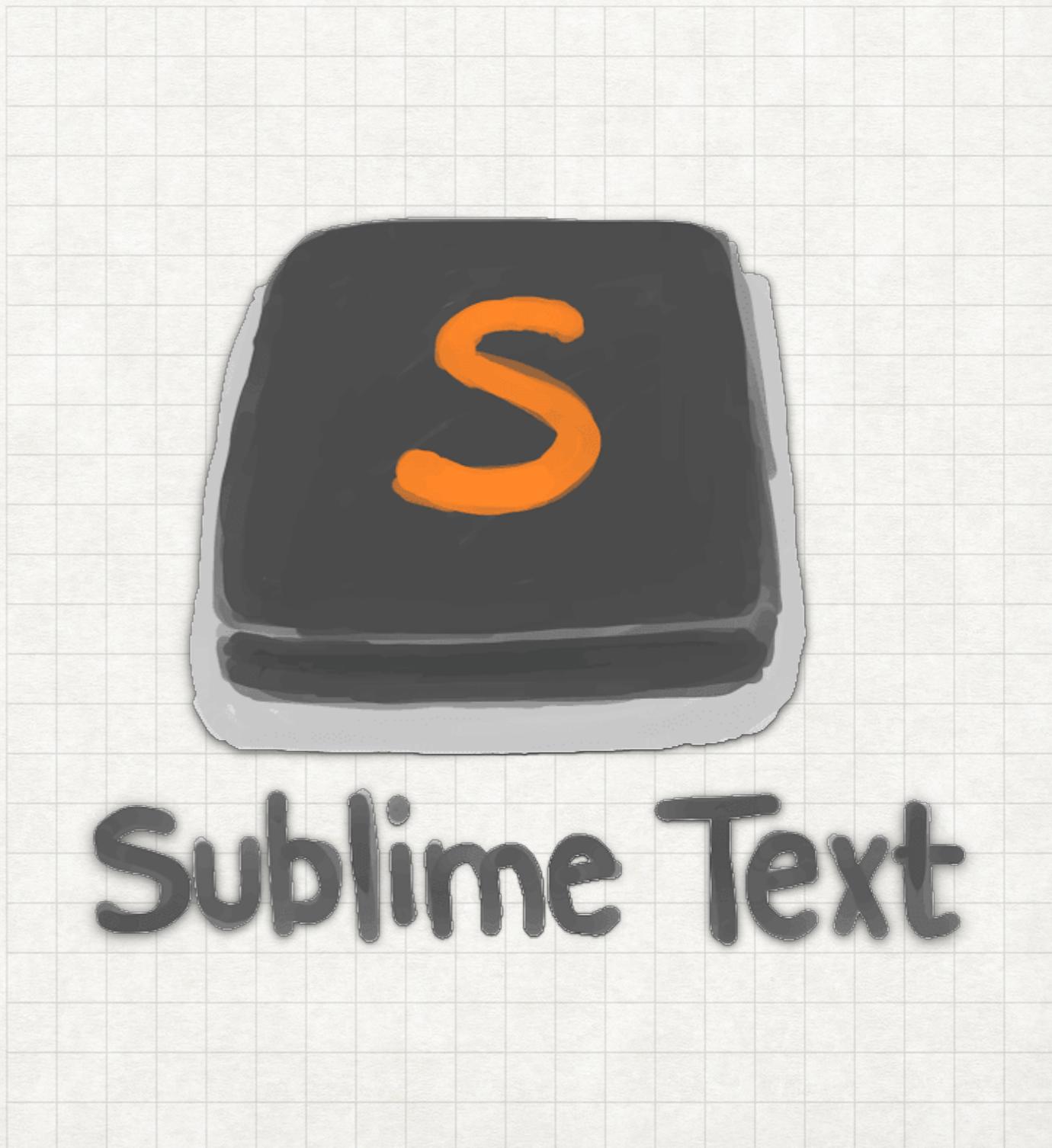
BREAKPOINT

WHY NODE.JS IS IN DEMAND?

- It's a prime backend language used in the web development
- It's an event-driven, non-blocking I/O language - it's a crucial feature for fast development as the connection between files, databases and APIs are done in a super quick fashion, non-blocking other requests from users
- It scales much better than any other backend language
- It's not the only good language for the backend development (i.e. we have also Java Enterprise with Spring Framework, PHP with Symfony/CodeIgniter),
but it's definitely one of the best
- For the code examples, use: <https://www.npmjs.com/>

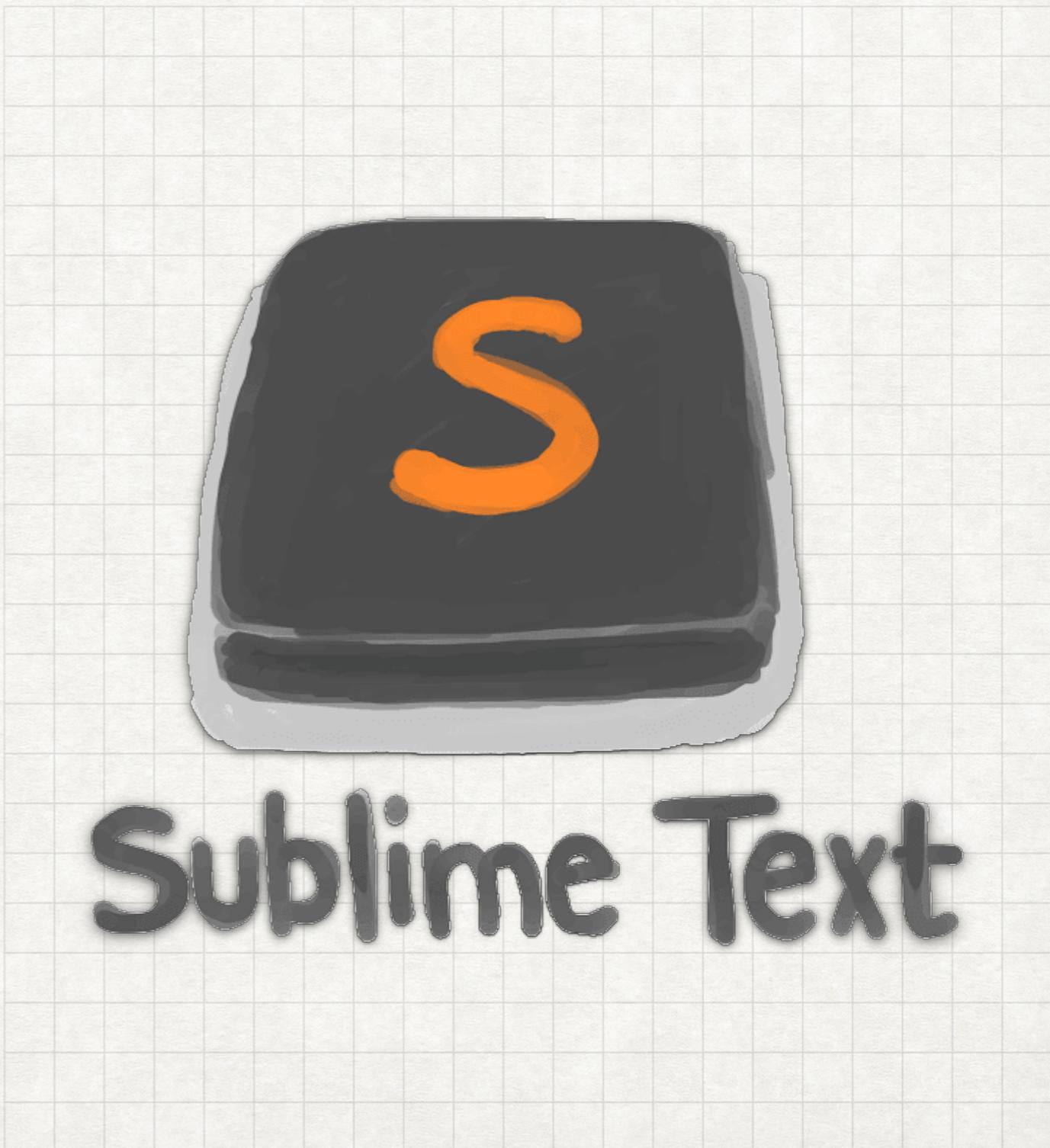
IDE

- We are also going to install an IDE
- Personally, I am going to use Sublime Text 3:
<https://www.sublimetext.com>
(pro-tip: you don't need to buy the license, just always cancel when it asks you for it)
- You can also use Atom:
<https://ide.atom.io/>
- Or even Vim:
<https://github.com/vim/vim>
(but then you'll be officially considered as a crazy person)



IDE

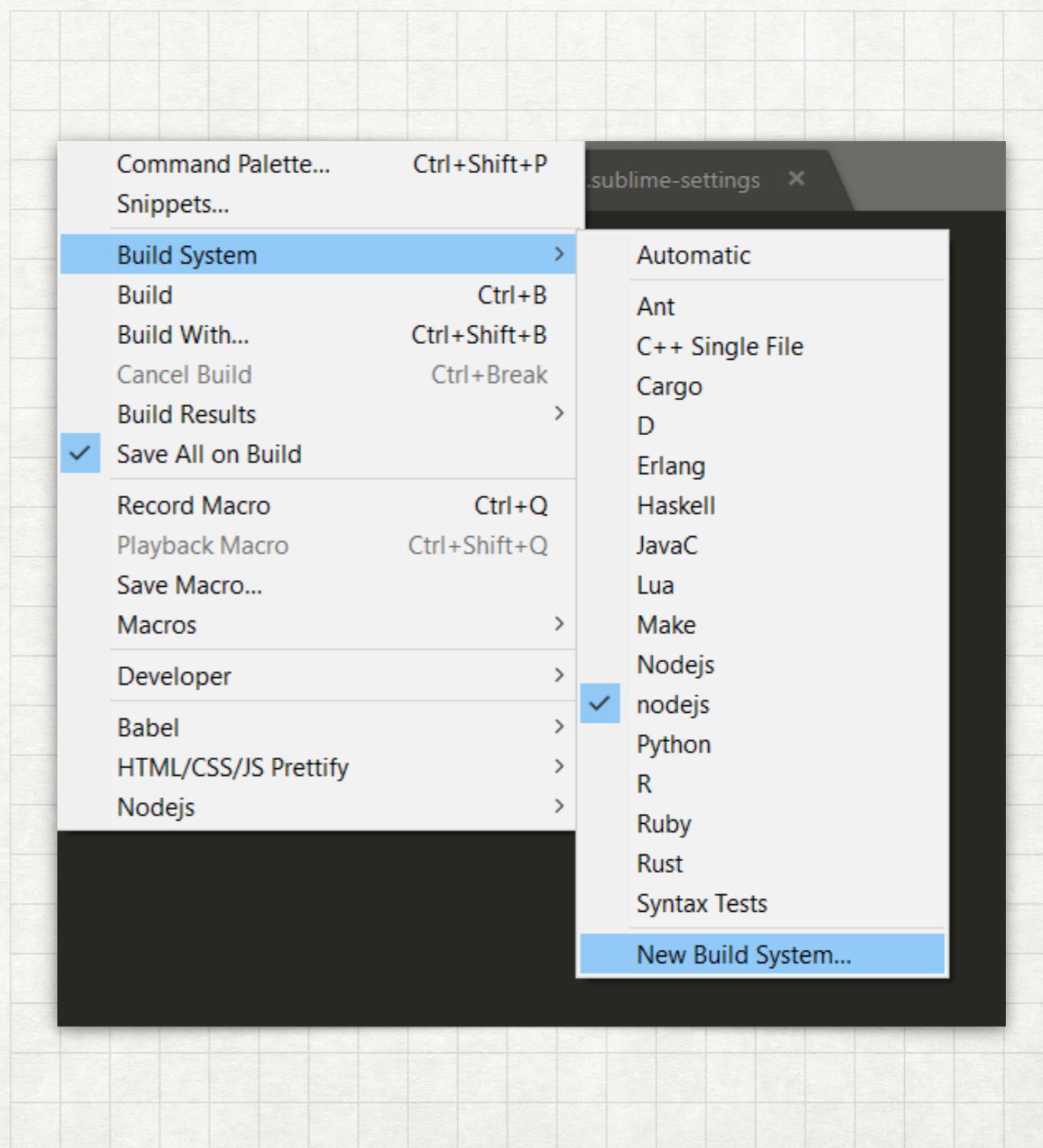
- We also can add additional features to Sublime as: autocomplete and/or formatter
- Let's open Sublime and press:
Preferences/Package Control
- Then choose: *Install Package*
- You can install those two:
nodejs; HTML-CSS-JS Prettify



IDE (OPTIONAL)

- One more step if your *nodejs* plugin doesn't work - a custom setup for Node build directly from Sublime
- Follow the image on the right
- Then paste this:

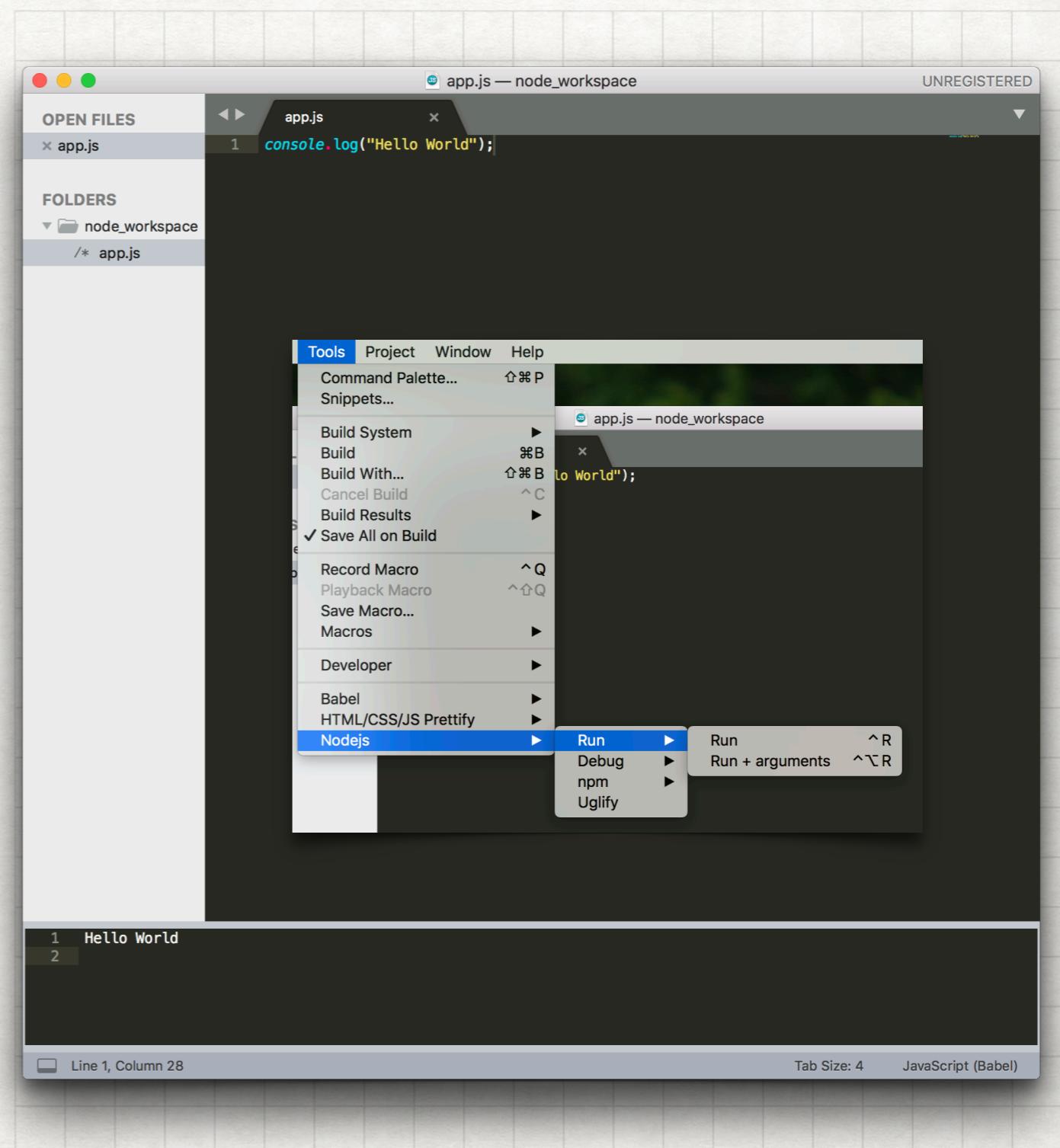
```
{  
  "cmd": ["C:/Program Files/  
  nodejs/node.exe", "$file"],  
  "selector" : "source.js"  
  
}  
(you may want to format it in the  
destination file)
```



BREAKPOINT

HELLO WORLD!

- Time for the first program!
- A super simple Hello World!
- Let's create a workspace
(I called mine:
node_workspace)
- Then inside the folder please
create an *app.js* file and write
console.log("Hello World");
- Next, press *Run* in a Nodejs
plugin and see the result



The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with 'OPEN FILES' and 'FOLDERS' sections. The 'OPEN FILES' section shows an 'app.js' file with the code: `1 console.log("Hello World");`. The 'FOLDERS' section shows a folder named 'node_workspace' containing an 'app.js' file. The main editor area has a dark background with light-colored text. The 'Tools' menu is open, and the 'Nodejs' submenu is selected, showing options: Run, Debug, npm, and Uglify. In the bottom right corner, there's a terminal window displaying the output: `1 Hello World`. At the bottom of the screen, there are status bars for 'Line 1, Column 28', 'Tab Size: 4', and 'JavaScript (Babel)'.

QUESTIONS?

MICHAŁ AIBIN, NEDA CHANGIZI