1.125 – Software Engineering and Architecting

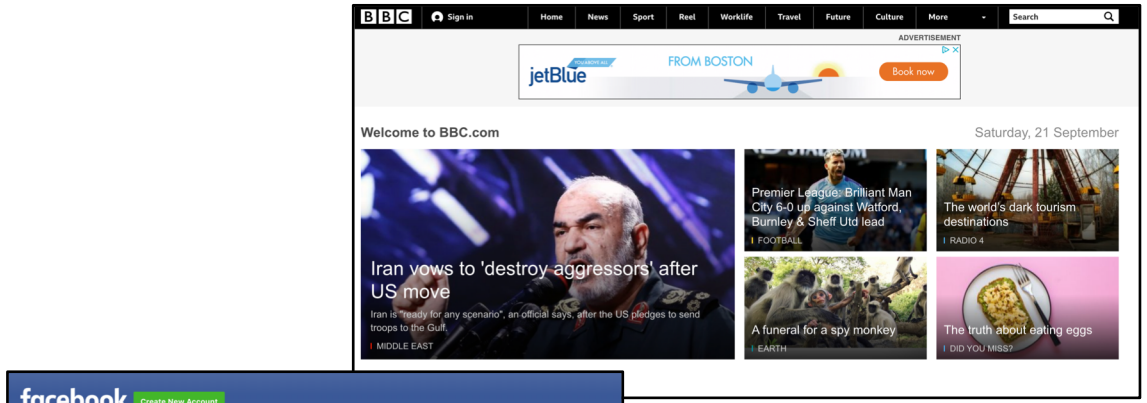
Lectures 5-6

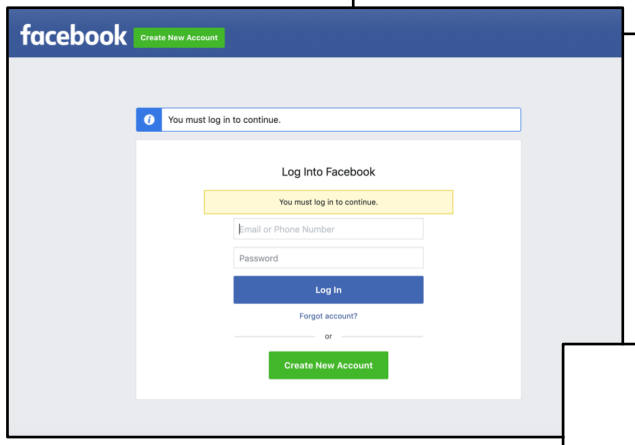
NodeJS and Backend Development

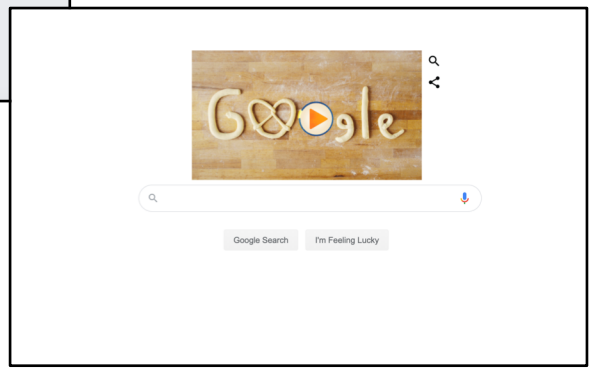
The Web – Our point of view

Front End

Websites/App interfaces







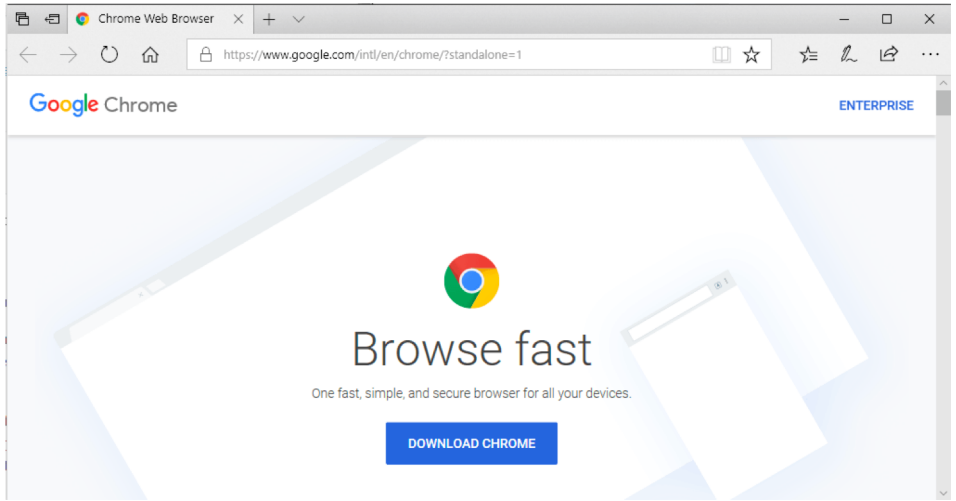
A Modern Web-App

Front End Websites/App interfaces	HTML - Structure CSS - Style JavaScript - Control
Back End Server/Databases	NodeJS - Control MongoDB - Storage The Cloud - Portability

One Language to rule them all

2008 Google creates a new browser from the ground up.

The JavaScript engine, V8 is extremely fast at running JavaScript



2009 Ryan Dahl has the idea to put the V8 engine on the server-side.

NodeJS was born.

One Language to rule them all

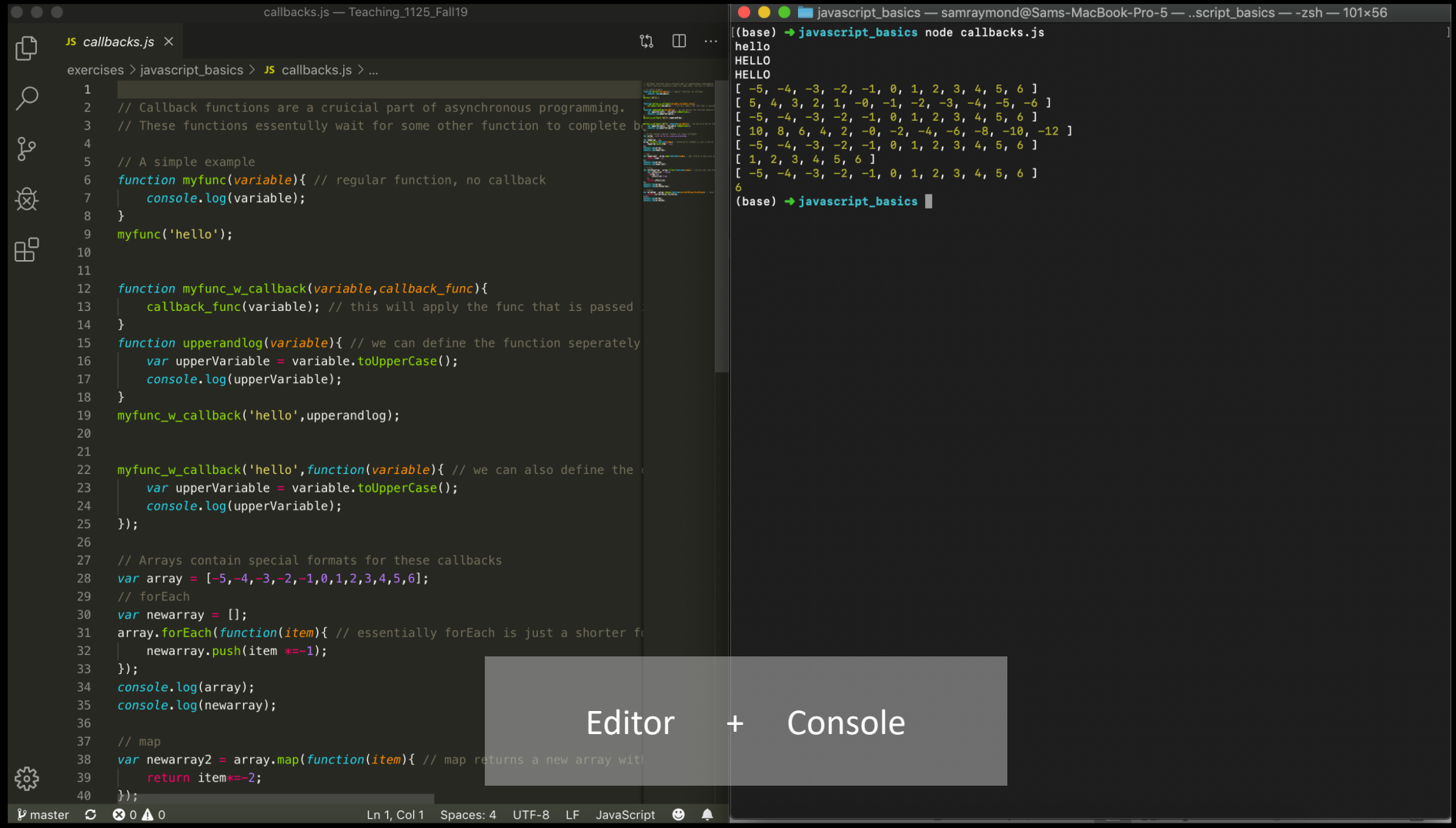
Why is it a big deal?

1 Language. Front end. Back end. Full Stack.

NodeJS Install

https://nodejs.org/en/

A soon-to-be-familiar sight



In Class Activity

1. Create a new repo on your GitHub called: *nodeExamples*
2. Clone this empty repo to your computer
3. Create a new file with the filename: *nodeExamples1.js*
4. Solve the following problems:
    1. Hash-Pyramid: create a function that will take as an input the height of a half pyramid and print out to the node console a pyramid of hashtag signs.

    e.g. pyramid(4) would yield the following output.

    #

    ##

    ###

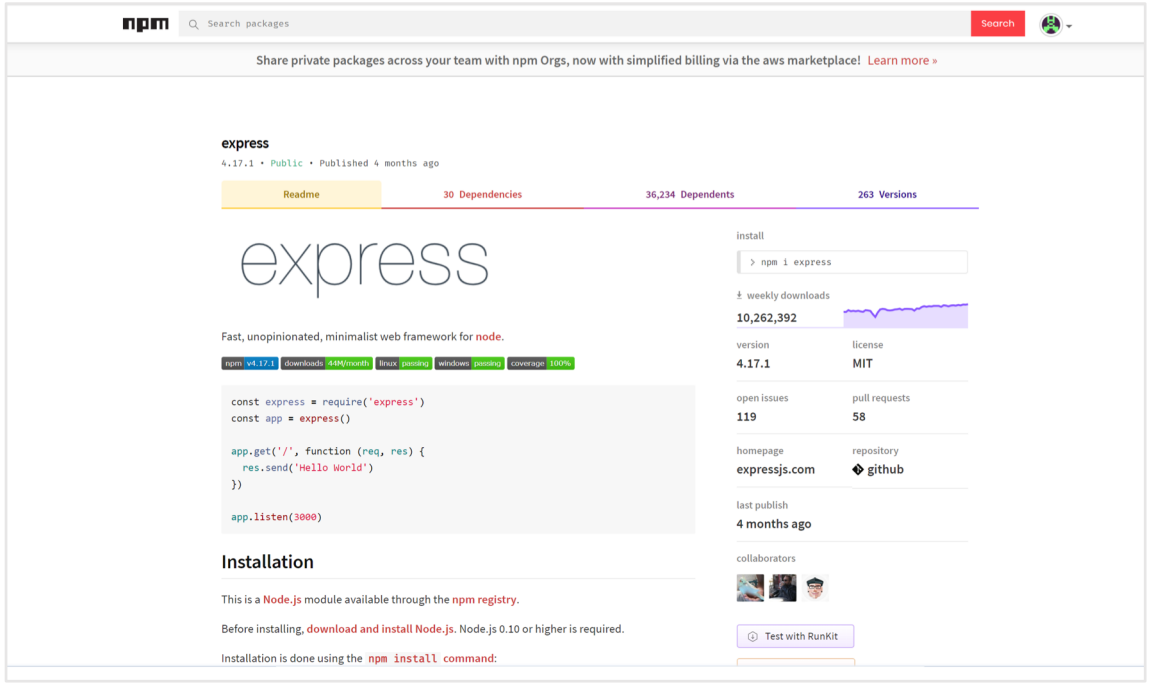
    ####

5. After you have finished, update your github repo

The Node Package Manger

Downloads Last Week

13 BILLION… can you think of anything else that happened 13 Billion times in the last week?



What a typical homepage for a library looks like

package.json

Lists the packages your project depends on

Specifies versions of a package that your project can use using semantic versioning rules

Makes your build reproducible, and therefore easier to share with other developers

```
{
  "name": "expressdockerimage",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {

    "express": "^4.16.4"
  }
}
```

Inside a package.json file

require()

How do we use these packages?

They are available as objects with internal functions (like the console.log()) after we add the ‘require()’ function

```
1  var express = require('express');
2
3  var app = express();
4
5
6  app.get('/',function(request,response){
7      response.sendFile(path.join(__dirname + '/index.html'));
8  });
9
```

Your First Web Server

Create a new repo on GitHub – Use the Node git ignore

Use npm init to create a new node package

Install express and path with npm

Create the following files:
index.html
frontend.js
server.js

Upload the files to GitHub.