ES6 class review

NPM

node.js, npm, webpack, babel

Create React App

src/ vs public/

.gitignore public/

1.
Node.JS

Node.JS
About

JavaScript environment that works without web browser!

- developed in 2009 by Ryan Dahl
- cross-platform
- open source & free
- async (operacje wejścia-wyjścia nie blokują wątków)
- fast (uses JavaScript engine - V8)
- allows to use JavaScript server side
- npm (node package manager)

Node.JS
Installation

https://nodejs.org/en/download/package-manager/

```
$ sudo apt-get install -y nodejs

$ node -v
$ npm -v
```

Node.JS
Quickstart

We can use node directly in terminal by typing:

```
$ node
```

We can type `ctrl + c` to quit.

We can also make simple script and run it:

```
$ nano simple.js
$ node simple.js
```

Task 0

Make a simple script that console.logs 1, 2, 3 … 10 and run it through node.js

Node.JS

Simple HTTP server using Node.JS *

```
const http = require('http');
const fs = require('fs')
const index = fs.readFileSync('./index.html')

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer(function(req, res) {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/html');
  res.end(index);
});

server.listen(port, hostname, function() {
  console.log('Server running at http://' + hostname + ':' + port + '/');
});
```

2.
NPM

NPM

About - node package manager

It is the **default package manage**r for the JavaScript runtime environment **Node.JS**. It consists of a command line client, also called npm, and an online database of public and paid-for private **packages**, called the npm registry.

- *Wikipedia*

NPM
Idea of packages/modules

Main idea behind NPM was to:

- create way to easy publish and install libraries
- create a way do manage dependencies
- create a way to make package versions meaningful
- create a file describing the project and its dependencies

NPM
Initialization of a project

We can init project typing:

```
$ npm init
```

It will create a package.json file through a creator.

This file contains our project meta-data and its dependencies.

Task 1

Init a npm project in new directory.
Check the package.json file.

Make an index.html there.

NPM
Initialization of a project

```
/* ./package.json */
{
  "name": "sample-app",
  "version": "1.0.0",
  "description": "node.js, npm, webpack sample",
  "main": "index.js",
  "scripts": {
    "echo-test": "echo 'test'"
  },
  "author": "Foo Bar",
  "license": "ISC"
}
```

Creates package.json

```
$ npm init
```

NPM
Initialization of a project

In most cases we don’t edit package.json manually. NPM stores there the dependencies that we can install in our project.

If we want to install e.g. jQuery we can type:

```
$ npm install jquery
```

NPM
Initialization of a project

NPM creates folder node_modules where all installed modules are stored. This folder is local (only visible in this folder).

We can install packages globally, mostly a command line programs that can do some stuff for us (surge is a node package like this).

We can type:

```
$ npm install --global http-server
$ npm install -g http-server
```

NPM
http-server package

http-server is super simple package that creates a HTTP server that serves content of folder that we are in!

We can type:

```
$ http-server
```

... and that’s all :)

Task 2

Install jQuery in your project.
Try to find it JS file and include
it to your HTML file.

NPM

npm scripts

In **package.json** file we have a property called **scripts**. We can define own scripts there and call them, by typing:

```
$ npm run <script_name>
```

There are two shortcuts:

`$ npm start` is the same as `$ npm run start`
`$ npm test` is the same as `$ npm run test`

Task 3

Make an npm script that runs http-server when user types npm start.

NPM
Installing packages

If we want to share our project we don’t need to share the dependencies - there are public - everyone can download it themselves, so we don’t want and **shouldn’t add node_modules to our version control systems!**

Ofc course no one want to install the dependencies manually. We can type **npm install** and npm will handle the installation from informations stored in **package.json, so we want to add it to our VCS!**

Task 4

Delete node_modules folder.
Check that site doesn't works - missing jQuery!
Run npm install and check again.

NPM
Package versioning

Behind npm is a system called semantic versioning (semver) that sets some rules about naming packages versions.

“A version is made up of three parts: X,Y,Z where those are major, minor and patch versions respectively. An example would be 1.2.3, or major version 1, minor version 2, patch 3.
A change in patch represents a bugfix that doesn’t break anything. A change in minor version represents a new functionality that doesn’t break anything. A change in major version represents a large change that breaks compatibility. If users don’t adapt to a major version change, stuff won’t work.”

- quote from an article on Medium

3.
Build tols

Webpack
About

Webpack is probably the most popular module bundler for JavaScript development.

It creates bundles - combined, minified and optimized files that we can use for production.

Webpack also can:

- use transpiler and transpile our code
- load CSS and other sources into JS
- split code and serves it async when app needs it
- remove dead code (three shake)

Babel
About

Babel is an transpiler that do the process of transpilation - changing code that was written in newer versions of JavaScript into older, that is readable by all browsers.

We can code in ES6 or higher, or even use some functions that aren’t yet in ECMAScript specifications!

See it in action: http://babeljs.io/repl

Other tools*

- Gulp.js
- Grunt.js
- Rollup.js
- Parcel.js

4.
create-react-app

create-react-app
What is it?

create-react-app is an **npm package** that provides to us console command, that we can use to bootstrap whole project structure and **babel + webpack** configurations for developing in React !

This is an official Facebook tool - https://github.com/facebookincubator/create-react-app

Task 5

Install globally npm package create-react-app

create-react-app
Usage

Simply type it in the console:

```
$ create-react-app <folder_name>
```

or if we want to create app in folder that we are in:

```
$ create-react-app .
```

create-react-app
Commands

create-react-app provides us some useful scripts, most important are:

`$ npm start` - runs development server

`$ npm run build` - makes production build

More can be found - https://github.com/facebookincubator/create-react-app

create-react-app
App folder structure

App builded by create-react-app has folder structure like:

- **build** - here will be bundled files after `npm run build`
- **node_modules** - npm packages / dependencies
- **public** - public assets like index.html, favicon.ico etc.
- **src** - our code that will be transpiled and bundled by webpack
- **package.json** - all dependencies and pre-configured scripts
- **.gitignore** - already prepared .gitignore - we can add .idea to this file
- **README.md** - tons of instuctions how to use this tool

create-react-app
Start working

An entry file (where JS begins to execute) for **create-react-app** is:

`src/index.js`

We can delete rest of files in src and, so they won't disturb us and go on.

For exercises we can delete all `index.js` content.

Task 5

Make new directory
Init new GIT repo here
Run create-react-app
Try to start development server
Delete all files from src/ except index.js - delete it's content