Web Advanced

Javascript: Webpack, Babel

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

What is Babel?

Babel is a JavaScript compiler

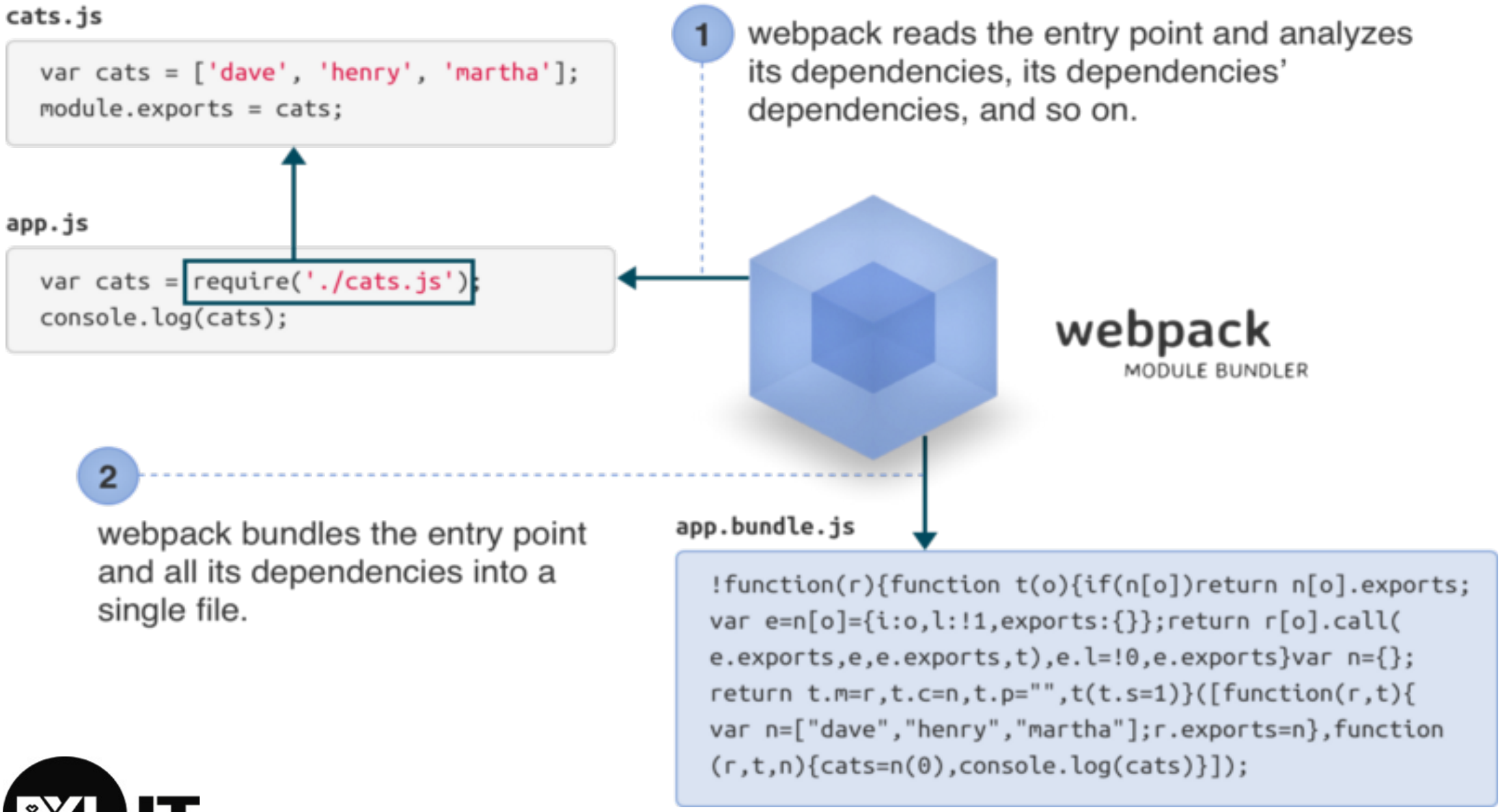
Babel is a toolchain that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments. Here are the main things Babel can do for you:

- Transform syntax
- Polyfill features that are missing in your target environment (through @babel/polyfill)
- Source code transformations (codemods)
- And more! (check out these videos for inspiration)

JavaScript Copy

```
// Babel Input: ES2015 arrow function
[1, 2, 3].map((n) => n + 1);

// Babel Output: ES5 equivalent
[1, 2, 3].map(function(n) {
  return n + 1;
});
```

What is Webpack?



PXL IT

Installatie

(1) Installeer nodejs
https://nodejs.org/en/download/

(2) Installeer npm
https://www.npmjs.com/get-npm

(beide moeten in je PATH-variabele zitten)

(3) Download de skelet-toepassing van Blackboard
webpack_babel_skeleton.zip

npm

$ npm init

maakt package.json aan

’lege’ beschrijving van het project

$ npm install prompt **$ npm i prompt**

installeert de dependecy prompt in de map node_modules/

beschrijft de dependency in package.json (en package-lock.json)

```
{
  "name": "test_npm",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "prompt": "^1.1.0"
  }
}
```

https://www.npmjs.com/package/prompt

PXL IT

npm

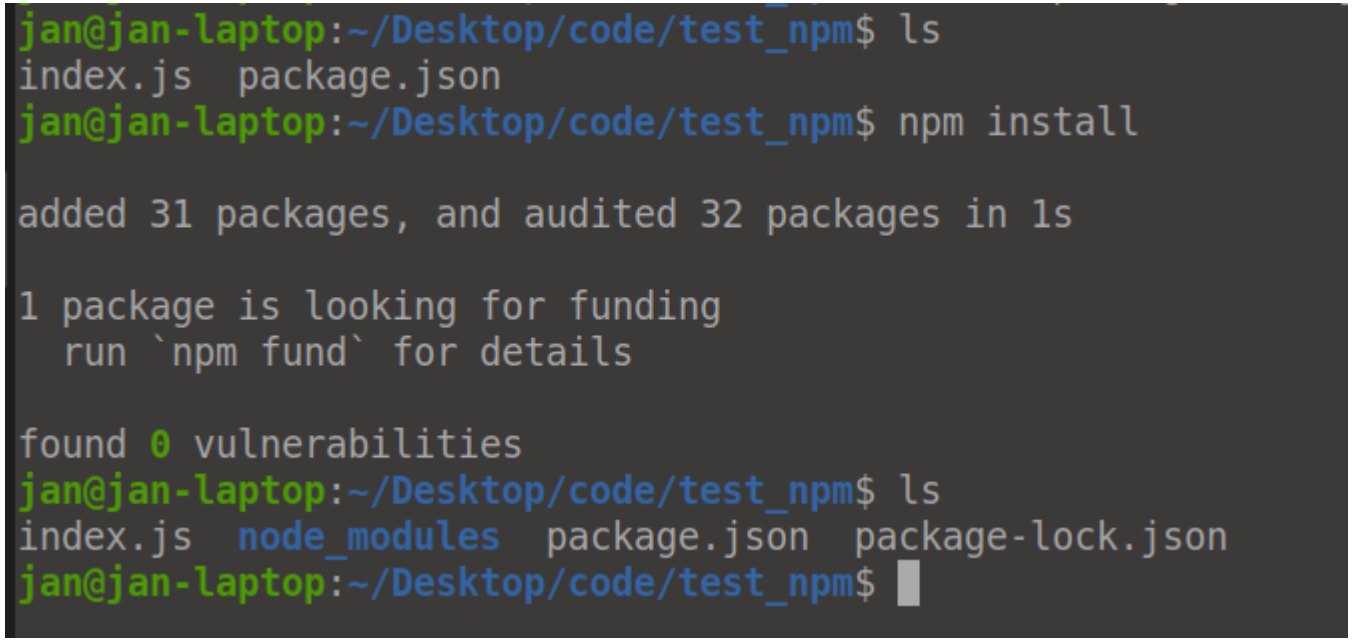
```
1  var prompt = require("prompt");
2  var colors = require("colors/safe");
3
4  prompt.message = colors.rainbow("Question!");
5
6  prompt.start();
7
8  prompt.get({
9    properties: {
10     name: {
11       description: colors.magenta("What is your name?")
12     }
13   }
14 }, function (err, result) {
15   console.log(colors.cyan("You said your name is: " + result.name));
16 });
```


```
jan@jan-laptop:~/Desktop/code/test_npm$ node index.js
Question!: What is your name?:  ok
You said your name is: ok
```

npm

Vertrek van bestaand bestand package.json
$ npm install $ npm i

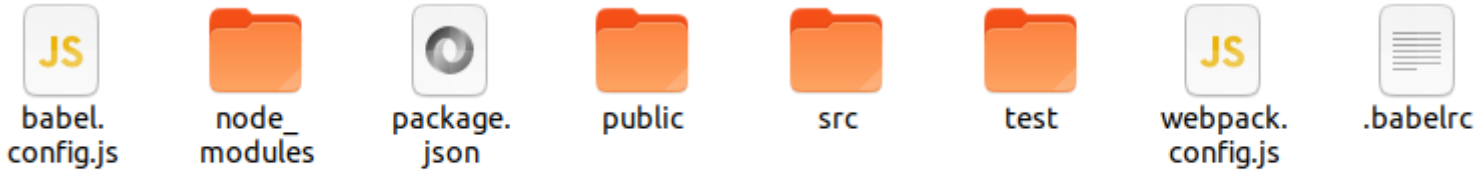
installeert de dependecy prompt in node_modules



In een git-repository: nooit node_modules (.gitignore)

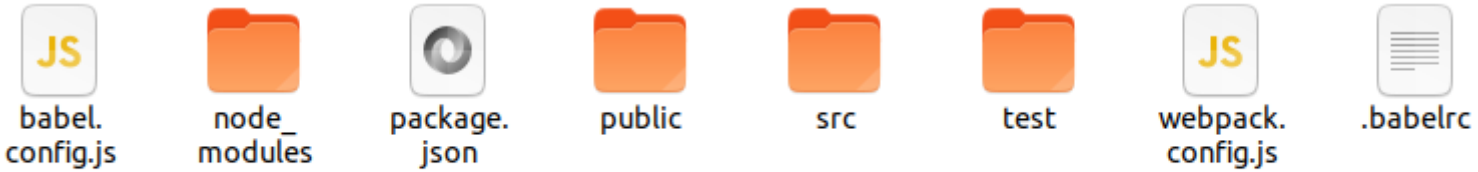
PXL IT

Voorbeeld1



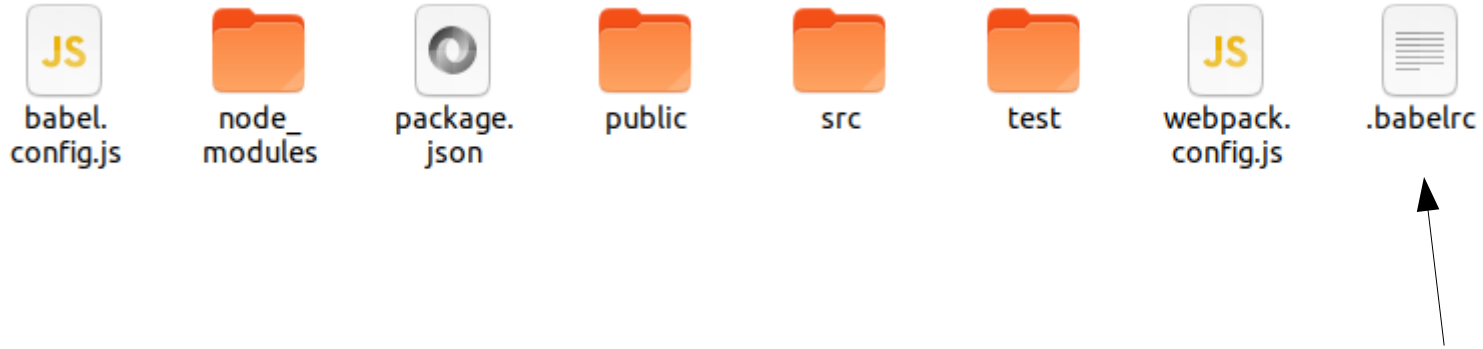

```
{
  "scripts": {
    "build": "webpack",
    "serve": "webpack serve",
    "test": "jest"
  },
  "devDependencies": {
    "@babel/core": "^7.12.9",
    "@babel/preset-env": "^7.16.11",
    "babel-loader": "^8.2.2",
    "core-js": "^3.8.0",
    "jest": "^26.6.3",
    "jest-cli": "^26.6.3",
    "webpack": "^5.8.0",
    "webpack-cli": "^4.2.0",
    "webpack-dev-server": "^4.7.3"
  }
}
```

PXL IT



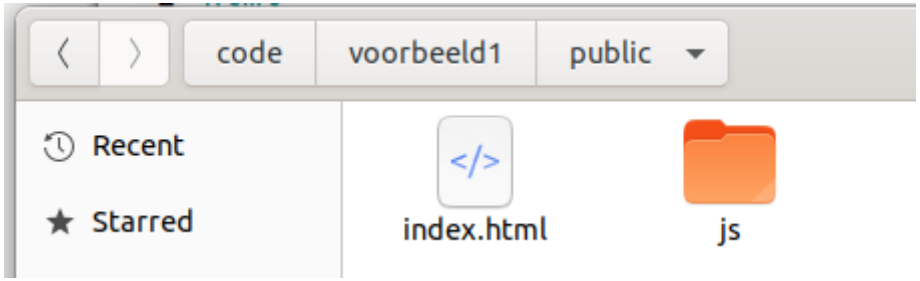

```
const path = require("path");
module.exports = {
  entry: './src/js/app.js',
  mode: 'development',
  output: {
    path: path.resolve(__dirname, "public/js"),
    filename: "bundle.js"
  },
  module: {
    rules: [{
      test: /\.js$/,
      exclude: /node_modules/,
      use: {
        loader: 'babel-loader',
        options: {
          presets: [
            ['@babel/preset-env']
          ]
        }
      }
    }]
  }
}
```

PXL IT



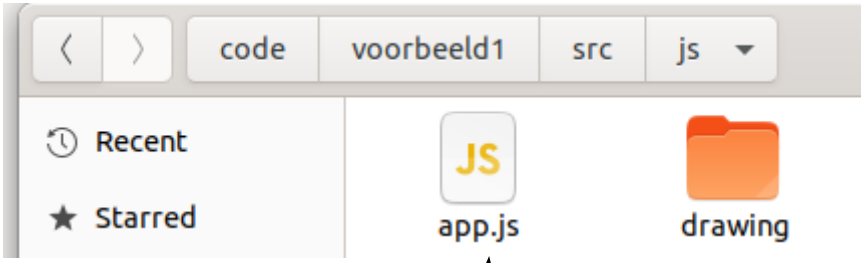

```
{
    "presets": [
      ["@babel/preset-env", {
        "targets": {
          "browsers": ["last 2 versions"]
        }
      }]
    ],
}
```

PXL IT



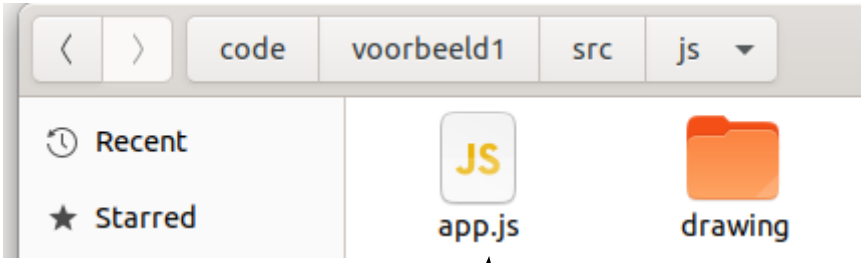

```html
<!DOCTYPE html>
<html>
  <head lang="en">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Webpack Babel Skeleton</title>
  </head>
  <body>
    <div id="output"></div>
    <script src="./js/bundle.js" charset="utf-8"></script>
  </body>
</html>
```

PXL IT



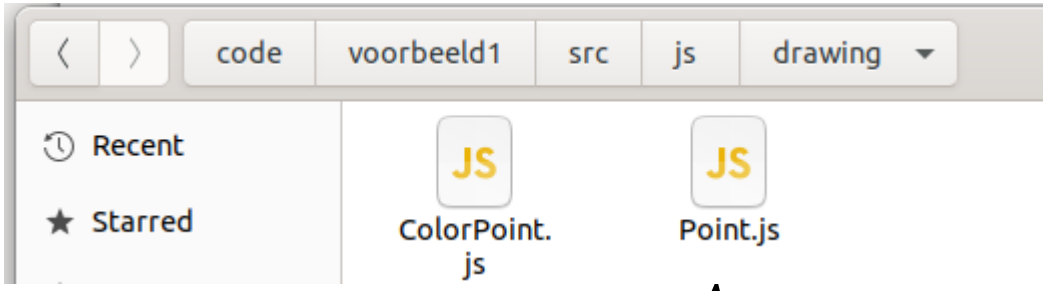

```
"use strict";

import Point from './drawing/Point';
import ColorPoint from './drawing/ColorPoint';

window.addEventListener('load', (event) => {
    let point=new Point(1,2);
    let preElement=document.createElement('pre');
    let textNode=document.createTextNode(point+'\n' );

    let colorPoint=new ColorPoint(1,2,'red');
    let textNode2=document.createTextNode(colorPoint);

    preElement.appendChild(textNode) ;
    preElement.appendChild(textNode2) ;
    document.getElementById('output').appendChild(preElement);
});
```

PXL IT



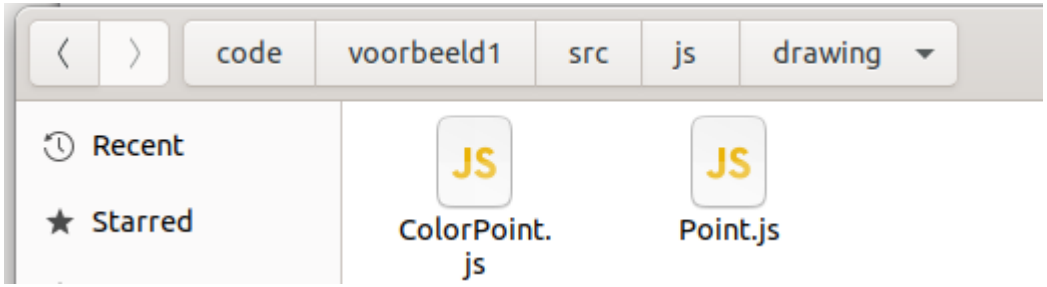

```
"use strict";

import Point from './drawing/Point';
import ColorPoint from './drawing/ColorPoint';

window.addEventListener('load', (event) => {
    let point=new Point(1,2);
    let preElement=document.createElement('pre');
    let textNode=document.createTextNode(point+'\n' );

    let colorPoint=new ColorPoint(1,2,'red');
    let textNode2=document.createTextNode(colorPoint);

    preElement.appendChild(textNode) ;
    preElement.appendChild(textNode2) ;
    document.getElementById('output').appendChild(preElement);
});
```

PXL IT




```
1  "use strict";
2
3  export default class Point{
4      constructor(x,y){
5          if (typeof x != 'number' || typeof y !='number') {
6              throw new Error("Parameter is not a number!");
7          }
8          this._x=x;
9          this._y=y;
10     }
11
12     get x(){
13         return this._x;
14     }
15     get y(){
16         return this._y;
17     }
18
19     toString() {
20         return `(${this._x},${this._y})`;
21     }
22 }
```

PXL IT




```
1  "use strict";
2
3  import Point from './Point';
4
5  export default class ColorPoint extends Point {
6      constructor(x, y, color) {
7          super(x, y);
8          this._color = color;
9      }
10     toString() {
11         return `${super.toString()} in ${this._color}`;
12     }
13 }
```

PXL IT

Alle modules downloaden:

$ npm install

babel, webpack, jest worden in node_modules/ geplaatst

Bundle maken

$ npm run build

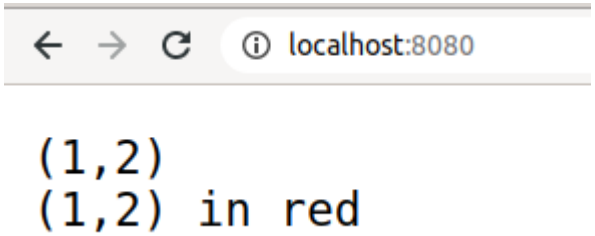
script met naam build (uit package.json) wordt uitgevoerd

```
"scripts": {
  "build": "webpack",
  "serve": "webpack serve",
  "test": "jest"
},
```

in public/js/ wordt bundle.js gemaakt

Server starten

$ npm run serve



PXL IT

Oefening1

(1) Vertrek van voorbeeld1 in webpack_babel_jest_skeleton.zip op Blackboard. Maak de klasse Line. Een Line bestaat uit 2 Point-obecten (point1 & point2).
Maak een constructor die 2 argumenten heeft. Als een van deze argumenten geen Point is wordt een Error opgeworpen Anders worden de argumenten toegekend aan point1 en point2.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/instanceof
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/throw
Voorzie getters voor point1 en point2 en toString.

~~Maak gebruik van Jest om constructor, getters en toString te testen.~~