ALESSANDRO BELLINI
FRONTEND DEVELOPER @ SPRYKER

spryker.com
ilmente.com
@ilmente

WEBPACK

JUST A FEW THINGS BEFORE WE START…

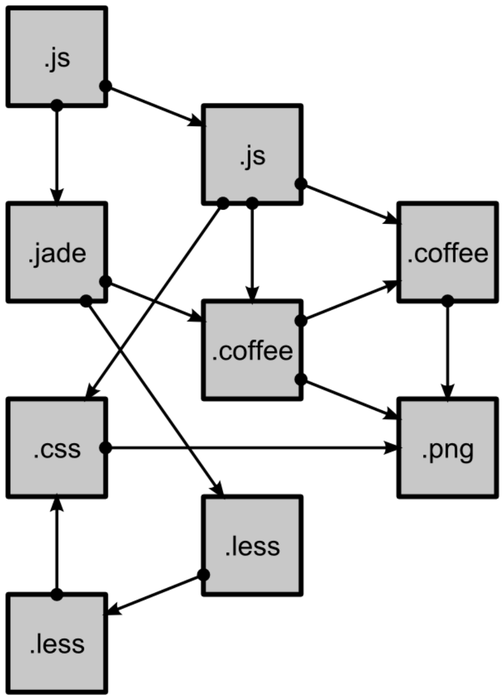
- **thanks** for being here
- my english is **horrible**, so please excuse me
- no, really: it **sucks**! You have no idea…
- I'm not a "guru": this is the result of a my **work**
- I'm here to **share**, not to teach
- fell free to **ask** and comment

AGENDA

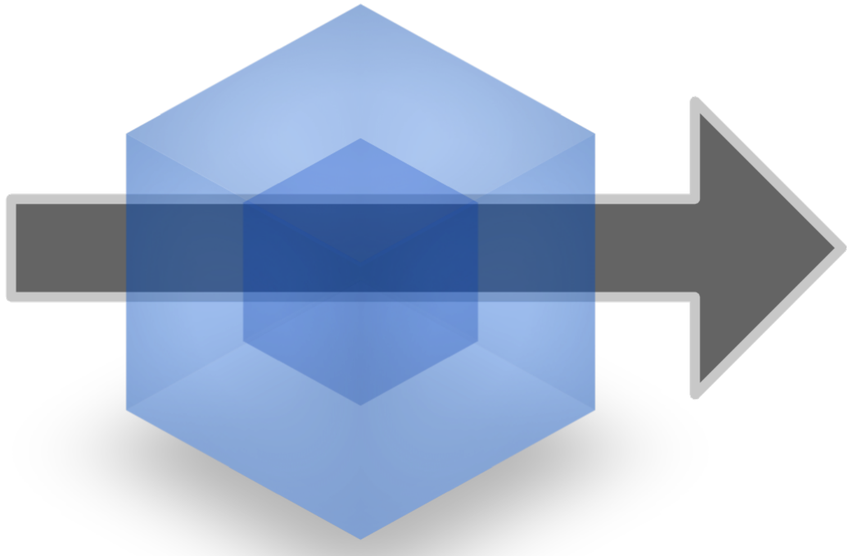
- webpack
  - what it is and how it works
  - the configuration file
  - loaders and plugins
  - code examples

WEBPACK

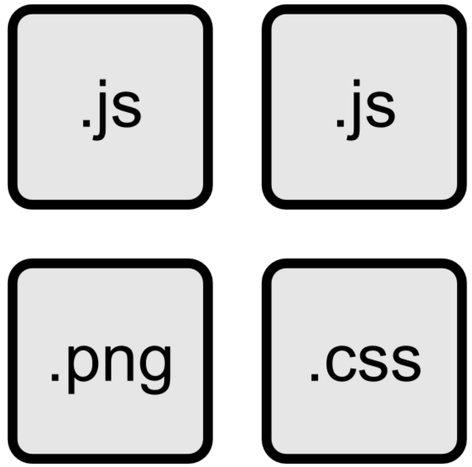
I STARTED WITH THIS



modules
with dependencies



webpack
MODULE BUNDLER



static
assets

THEN I READ

- a flexible module bundler, a tool…
- split the dependency tree into chunks loaded on demand
- keep initial loading time low
- every static asset should be able to be a module
- ability to integrate 3rd-party libraries as modules
- ability to customize nearly every part of the module bundler
- suited for big projects

I'M NOT ALONE

chshouyu · 10 months ago

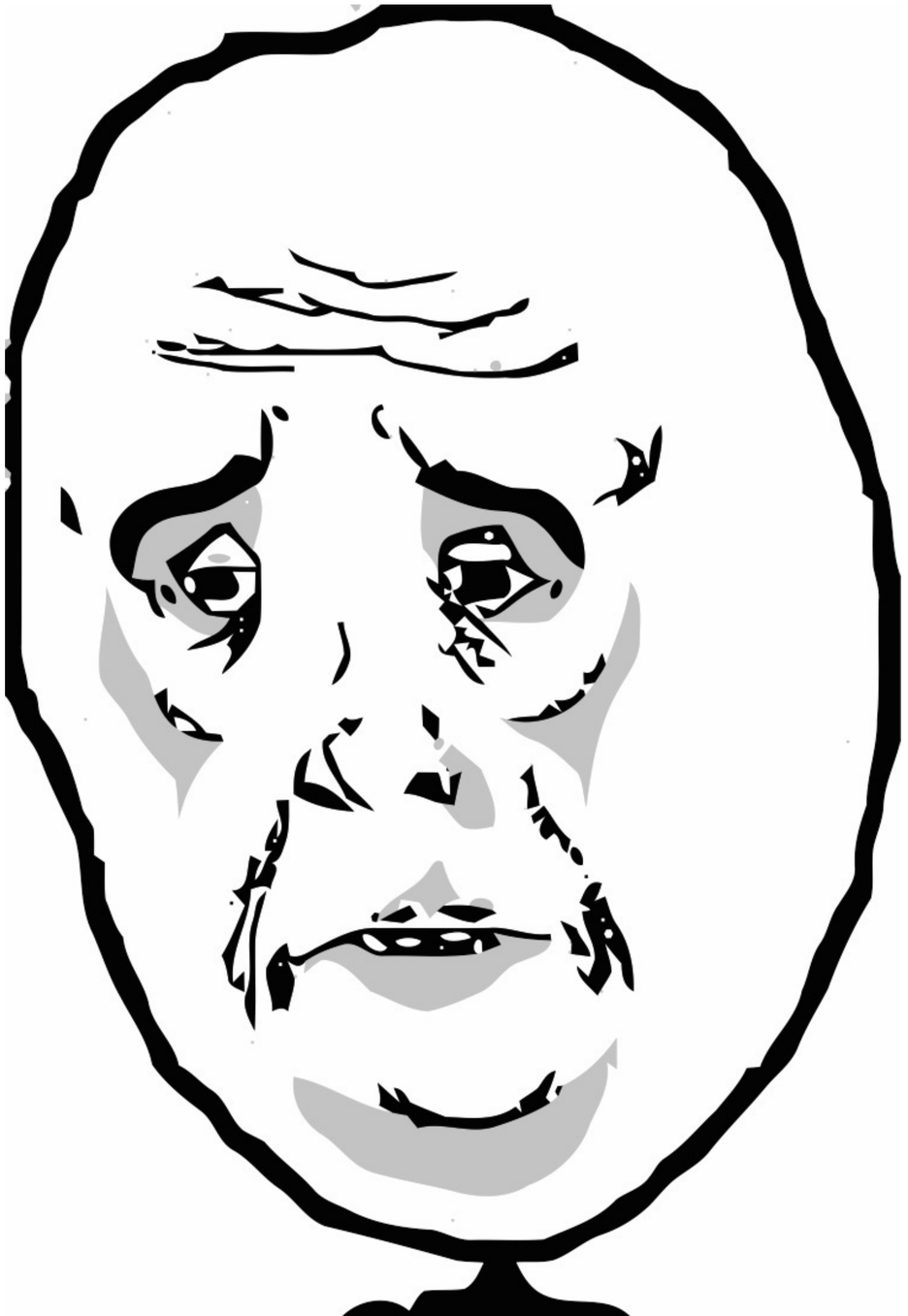
The document is so obscure, I read so many times,but still can not figure out how to use it comprehensively and cleanly.

69 ^ | v · Reply · Share ›

drumaddict71 ↗ chshouyu · 5 months ago

Thought I was the only one...! I came here from angular 2.It's just a module bundler,it's just a tool,yet it feels like rocket science with this bloated and all-over-the-place documentation

15 ^ | v · Reply · Share ›

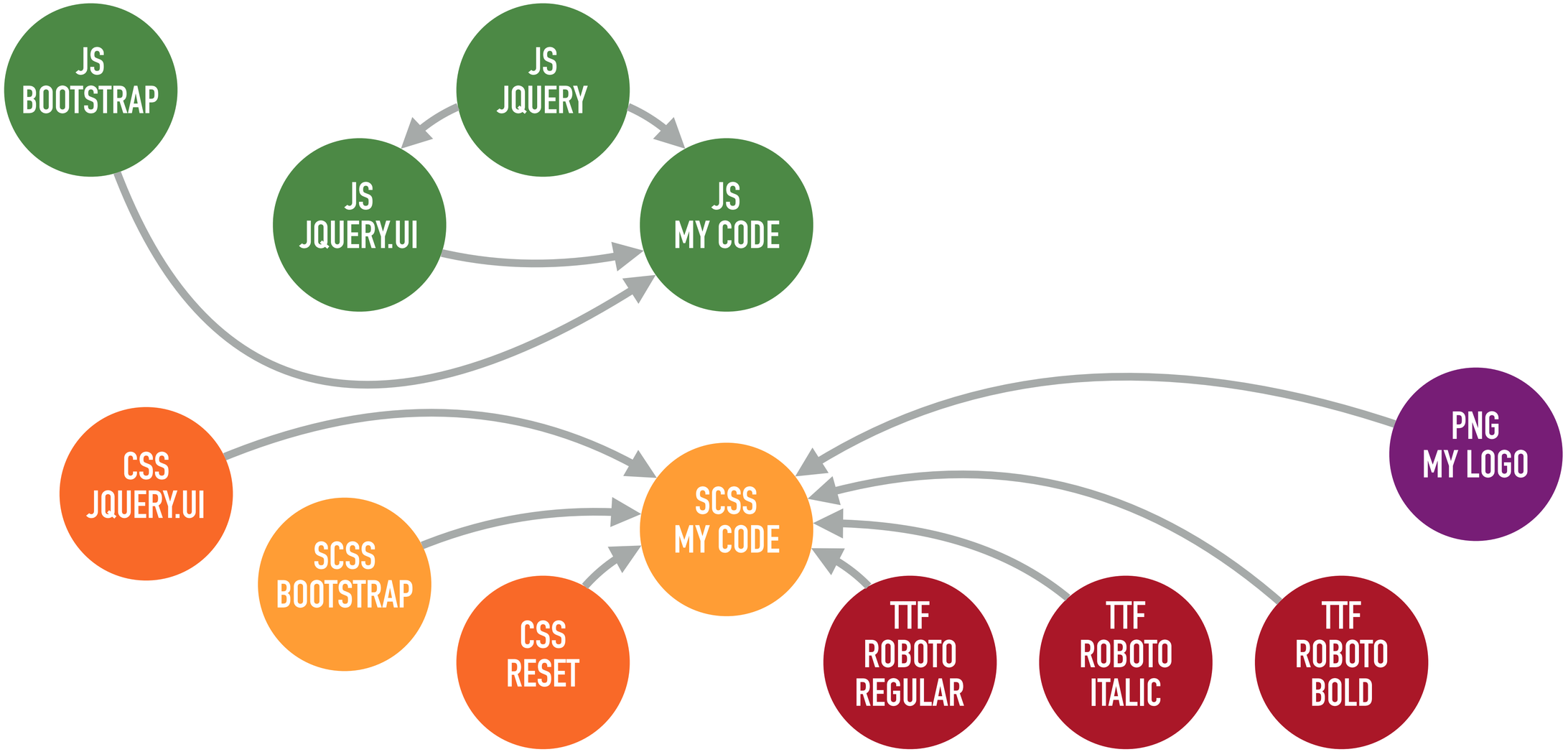


WEBPACK

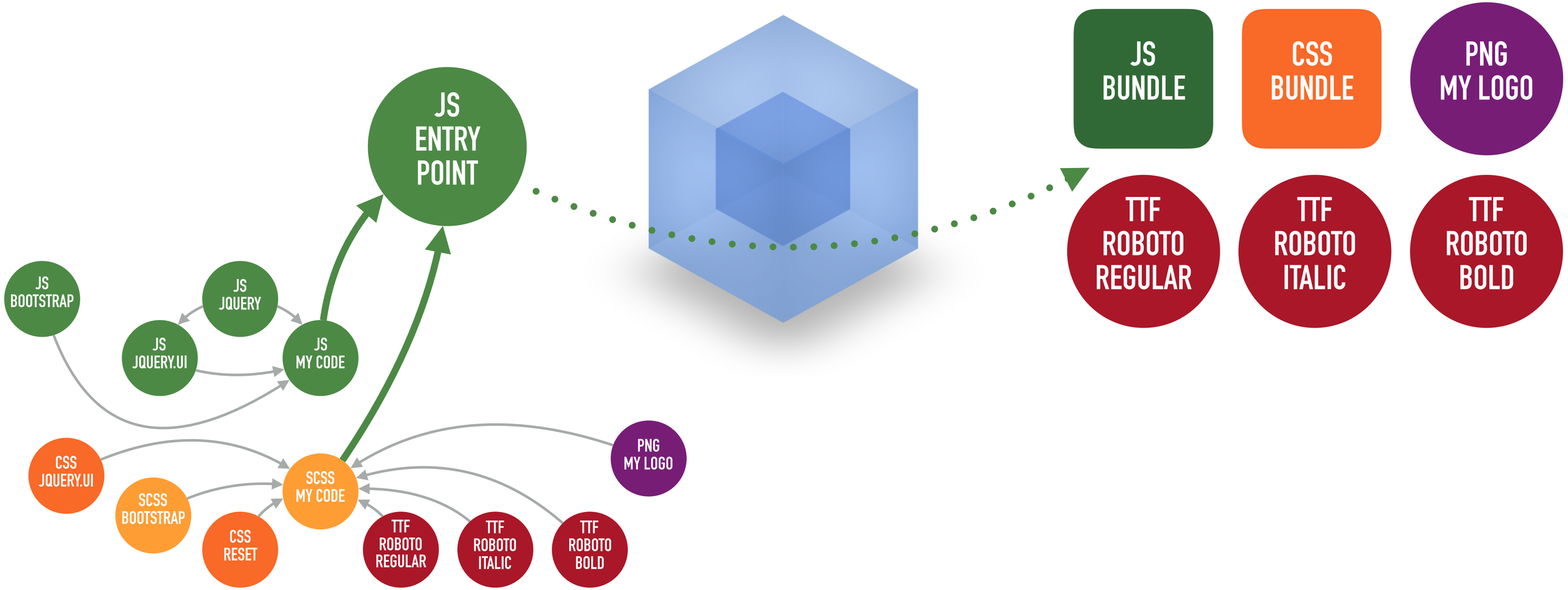
FIRST THINGS FIRST

- javascript is the king here
- webpack relies on CommonJS / AMD standards to define dependencies and export values
- every js module has its own scope
- webpack goes beyond: everything (*.js or not: .html, .jade, .css, .scss, .png, .ttf…*) can be a dependency

MY SIMPLE JQUERY PROJECT!



MY... JQUERY PROJECT!



“ASSETS CRAWLER AND BUNDLER”

- **javascript ES5.1**
  require(‘module-name’)
  require(‘../path/to/file’)
- **javascript ES6**
  import ‘module-name’
  import ‘../path/to/file’
  import {function} from ‘../path/to/file’
- **javascript mixed**
  require(‘./style.css’)
  require(‘../path/to/pattern.jpg’)
- **css**
  url(‘../fonts/font-name.ttf')
  url(‘../img/logo.png’)
- **sass**
  @import ‘module-style’

webpack searches for assets and puts them all together in one (or more) **bundle** file (when possible), **taking care of dependencies tree**

I DON'T KNOW WHAT **RESOURCE** YOU ARE.
[. . .] BUT WHAT I DO HAVE ARE A VERY PARTICULAR **SET OF SKILLS**.
[. . .] I WILL LOOK FOR YOU, I WILL FIND YOU AND I WILL **BUILD** YOU.

webpack

HOW IT WORKS

- npm install webpack [-g | - -save-dev]
- you can use it programmatically (node.js, gulp) or with CLI (when global)
- it's based on a node.js **configuration file** (a javascript object)

```
module.exports = {...}
```

CONFIGURATION FILE

- **entry** (string, array, object): application entry points
- **resolve.root** (string, array): absolute path(s) where to search for your modules
- **module.loaders** (array): list of automatically applied loaders
- **output.path** (string): the output absolute path
- **output.filename** (string): specifies the name of each output file on disk
- **plugins** (array): list of automatically applied plugins
- **devtool** (string): a developer tool to enhance debugging
- **watch** (boolean): enable watch mode

complete list on https://webpack.github.io/docs/configuration.html

LOADERS

- from the site:
  transformations applied on a resource file of your app: they are functions (running in node.js) that take the source of a resource file as the parameter and return the new source
- you can use loaders to tell webpack how to load sass, coffeescript or react jsx

more info on https://webpack.github.io/docs/how-to-write-a-loader.html

PLUGINS

- from the site:
  they add functionality typically related to bundles in webpack
- so… what exactly are they doing?
- you can use plugins to change or manipulate a resource, or the related build output [*bundle*]
- you can trigger a plugin during all the build process (from locating a single resource to saving the output [*bundle*]), extending webpack functionality

more info on https://webpack.github.io/docs/how-to-write-a-plugin.html

IN A NUTSHELL

- you have a resource to process with web pack
  - if you want to tell webpack *how to load* that specific type of resource: **use loaders**
  - if you want to *change how* this resource (or the relative output) *is built:* **use plugins**

LET'S TAKE A CLOSER LOOK

- configuration file structure
- watchers, devtool and source maps
- assets management and optimisation
- environment: development and production
- tests

JS

WEBPACK VS THE WORLD

- grunt: https://webpack.github.io/docs/usage-with-grunt.html
- gulp: https://webpack.github.io/docs/usage-with-gulp.html
- bower: https://webpack.github.io/docs/usage-with-bower.html
- karma: https://webpack.github.io/docs/usage-with-karma.html

there is no battle here:
webpack **is not a replacement**, but a tool that **can be integrated**

ANY QUESTIONS?

me, hoping you'll be merciful

webpack.github.io/docs

github.com/ilmente/webpack-devtalk

THAT'S ALL, FOLKS
THANKS