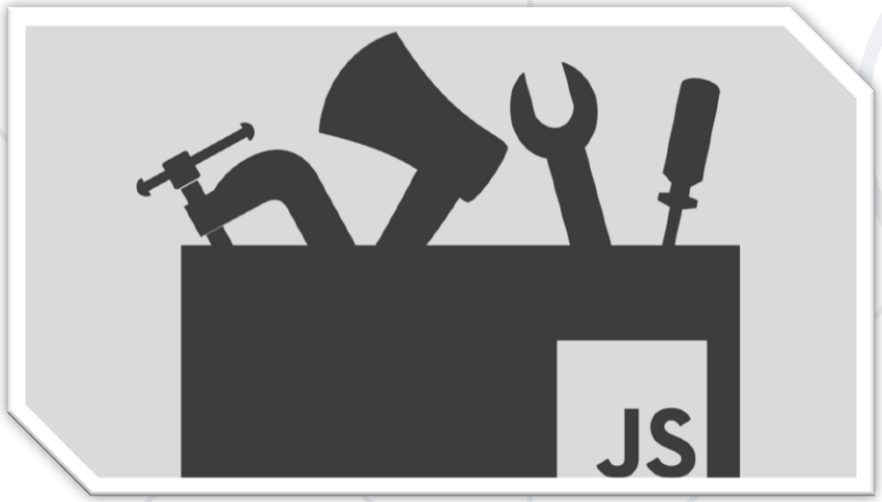
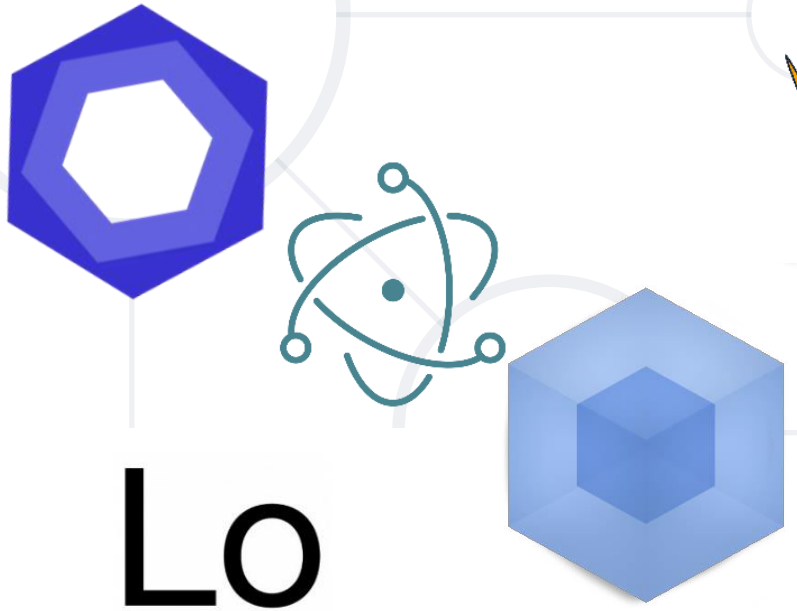
JS Tools and Libraries

Build Tools, ESLint, ElectronJS, Lodash





SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#JSCORE

Why Automate?

SoftUni Foundation

JavaScript does not need to be compiled

…Right?

Modern Development Process

Optimize and Start over

Build tools

SoftUni Foundation

- Build tools help us with the **build process** of our application
  - **Prepare scripts** for web (minify/uglify/concat)
  - **Run** Unit **Tests**
  - **Deploy** to Git, Cloud etc…
- Famous tools are
  - Grunt
  - Gulp
  - Webpack

Grunt

JS Task Runner

Grunt

SoftUni Foundation

- Grunt can be used for various **file operations** from merging two or more files into one to shortening **javascript/css** code
- Installing
  - First install the **Grunt** command line interpreter (**CLI**)

```
npm install grunt-cli -g
```

  - Than the Grunt itself

```
npm install grunt --save-dev
```

- You may later download additional **plugins**

GRUNT

The JavaScript Task Runner

Demo

Bundle up Files with Grunt

Gulp

Automate Workflow

Gulp

SoftUni Foundation

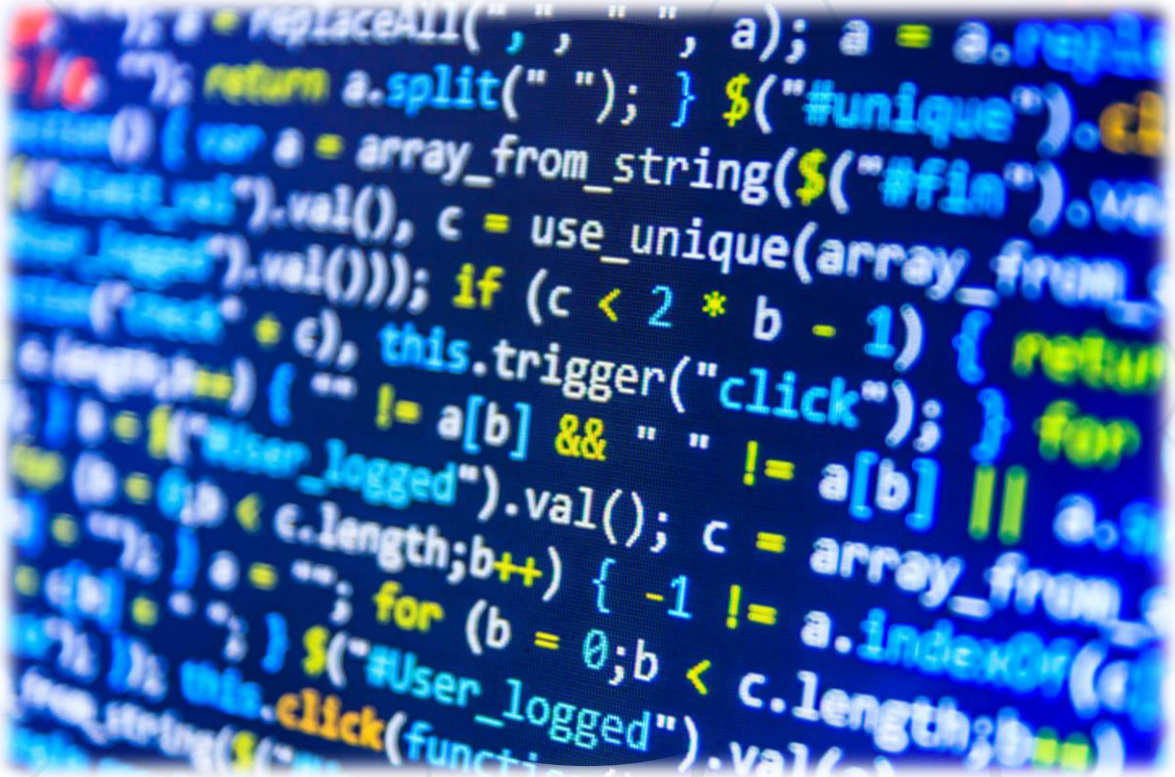
- **Gulp** is basically same as Grunt but **simpler** and **faster**
- Installing
  - First install the **Gulp CLI**

```
npm install gulp-cli -g
```

  - Than the **Gulp** itself

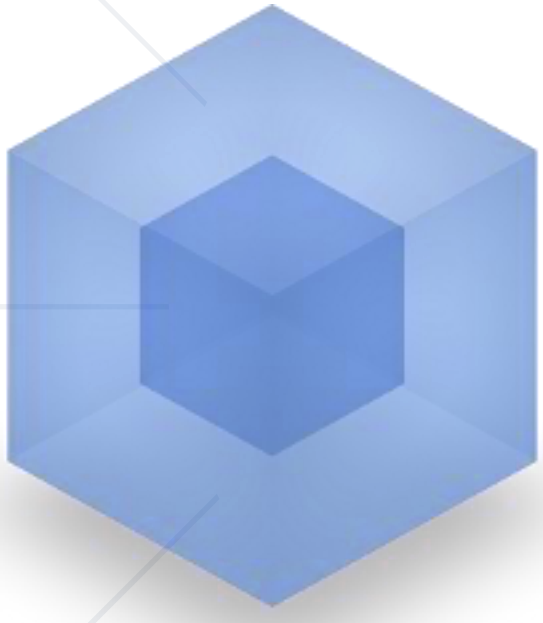
```
npm install gulp --save-dev
```

- You may download additional **plugins** in order to minify/concat files (same as “Grunt”)



Demo

Bundle up Files with Gulp



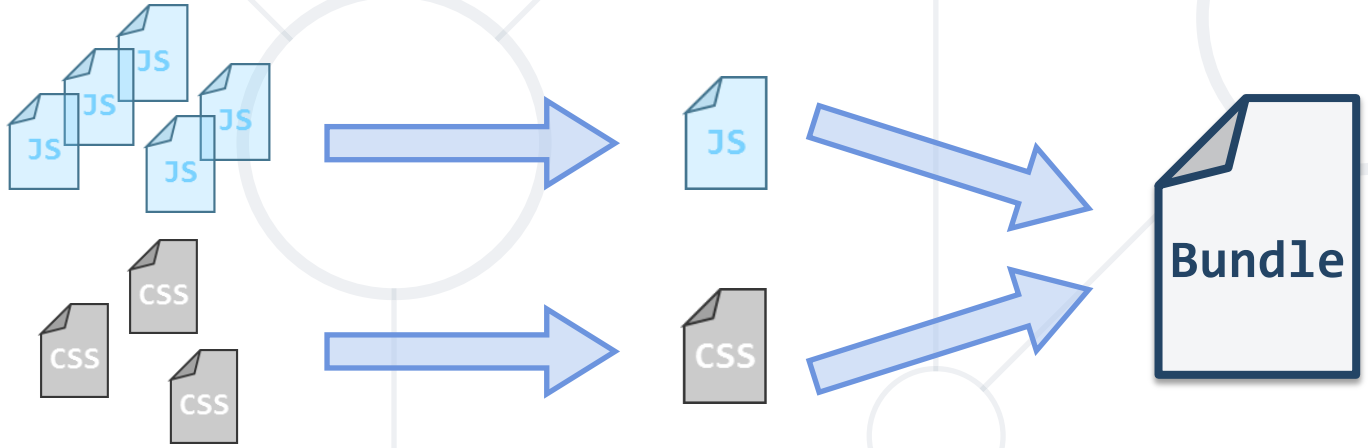
webpack
MODULE BUNDLER

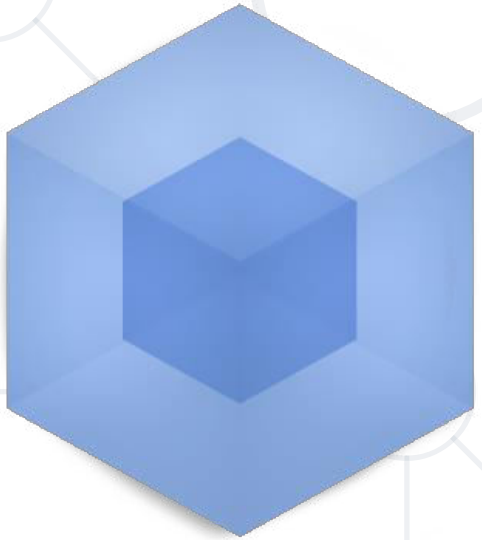
Webpack

A Wicked Smart Module Bundler

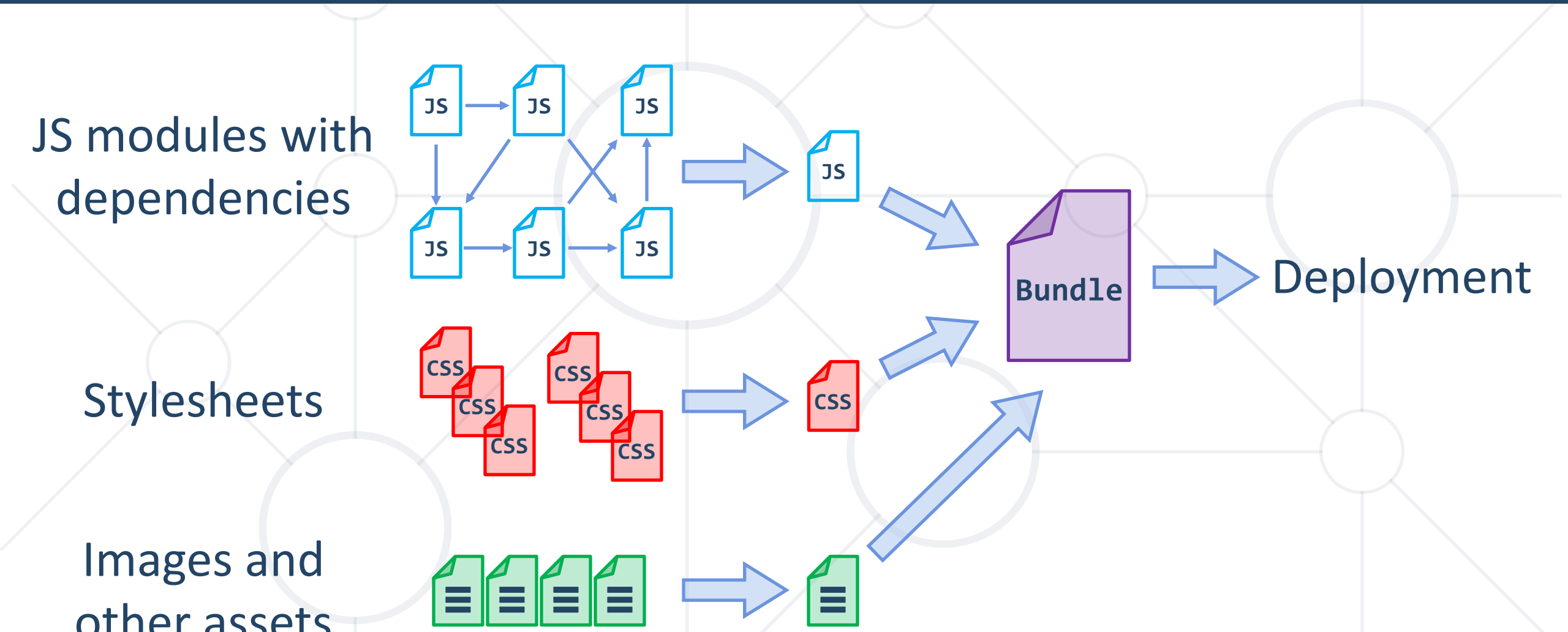
What is Webpack?

- One of the newest tools, combining **build** steps and **bundling**
  - Supports dependency management
  - Can load any 3rd party library as a module
  - Comes with it's own development server
  - Code splitting and loading as needed





Webpack Build Process



Installation and CLI

- Install **Webpack** via **npm**

```
npm install webpack -g
```

- Install the Webpack **command line interface**

```
npm install webpack-cli -g
```

- Install the add-on **development server**

```
npm install webpack-dev-server -g
```

Automation with Config Files

- Create a **webpack.config.js** file to automate your build
  - Configuration is in JSON format

```
module.exports = {
  entry: "./entry.js",
  output: {
    filename: "bundle.js"
  }
};
```

Starting module

Final output file

- When running **webpack** from the terminal, it uses this config

Enable Watch Mode

- Webpack can **watch** for file changes and **rebuild** the bundle
  - Add argument or change your **config file** to enable **hot reloading**

```
webpack --watch
```

```
module.exports = {
  entry: … , output: … ,
  watch: true
};
```

Web Server with Watch Mode

- Run the **development server** from the terminal

```
webpack-dev-server
```

  - Change your **config file** to enable **hot reloading**

```
module.exports = {
  entry: … , output: … ,
  devServer: {
    publicPath: "/dist/",
    watchContentBase: true,
  }
};
```

Processing Files with Loaders and Preloaders

SoftUni Foundation

- **Loaders** apply transformations to files
  - Can be downloaded with **npm** and configured in the **main config**

```
module.exports = {
  entry: "./entry.js",
  output: {filename: "bundle.js"},
  module: {
    loaders: [ … ]
  }
};
```

Loader format

```
{
  test: /\.jsx$/,
  exclude: /node_modules/,
  loader: "babel-loader"
}
```

- **Preloaders** are the same, they just run **before** any loaders

Scripts and Production Build

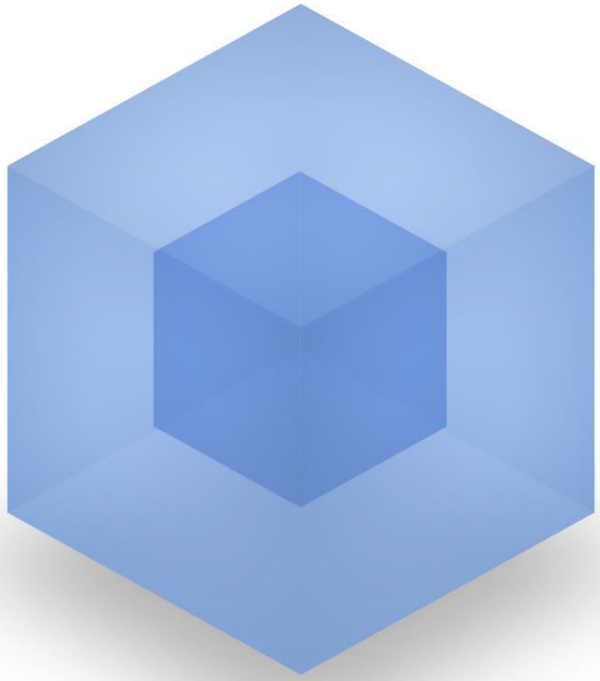
SoftUni Foundation

- Webpack uses **npm scripts** for further automation
  - Add to your **package.json** file

```
scripts: {
  "start": "webpack-dev-server"
};
```

- To **minify** the bundle for deploy, run webpack with **–p** argument
- You can specify a different config file for production

```
webpack --config webpack-production.config.js -p
```



Live Demo

Automated build with WebPack

Lodash

High-Performance Object Operations

Lodash

SoftUni Foundation

- Utility **library** that extends working with **objects**, **arrays** etc.

```
let _ = require('lodash');

_.findLast([1, 2, 3, 4], (n) => {
  return n % 2 === 0
}) // 4
```

- Main goal is **performance**.
- Reduces boilerplate and inconsistent code.

Example – Random Number

SoftUni Foundation

```
// Naive utility method
function getRandomNumber(min, max){
 return
   Math.floor(Math.random() * (max - min + 1)) + min;
 }
getRandomNumber(15, 20);
```

VS

```
let _ = require('lodash');

_.random(15, 20);
```

Example – Removing Property

SoftUni Foundation

```
Object.prototype.remove = function(arr) {
  let that = this;
  arr.forEach((key) => delete(that[key]));
};
let objA = {"name": "Pesho", "car": "ford", "age": 23};
objA.remove(['car', 'age']); // {"name": "Pesho"}
```

VS

```
let _ = require('lodash');

objA = _.omit(objA, ['car', 'age']); // {"name": "Pesho"}
```

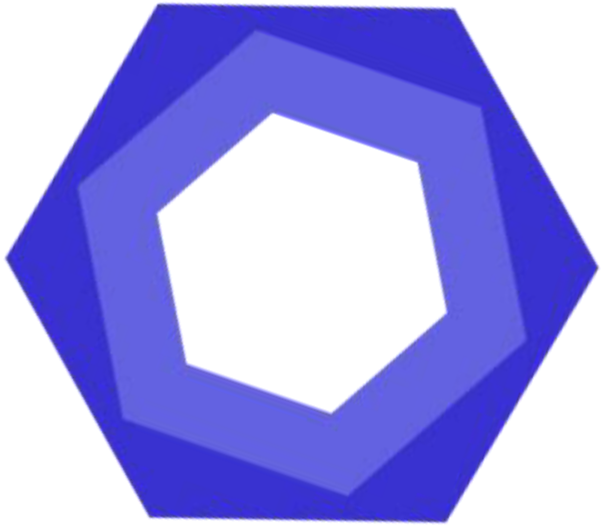
Example – Deep Cloning


```
let objA = { "name": "Stamat" };
// Normal method? Too long.
// You can try with JSON.stringify(JSON.parse(objA)).
// But it won't include methods.
```

VS

```
let _ = require('lodash');

let objB = _.cloneDeep(objA);
 objB === objA // false
```



ESLint

Pluggable Linting

ESLint

SoftUni Foundation

- ESLint is a **tool** which keeps track of code for inconsistent content (**typos**, **formatting** etc)
  - Main purpose is to **reduce bugs** and make the **code** more **readable**
  - It's done by following **rules** which can be configured

```
{
  "rules": {
    "semi": ["error", "always"],
    "quotes": ["error", "double"] }
}
```

Demo

Configuring ESLint

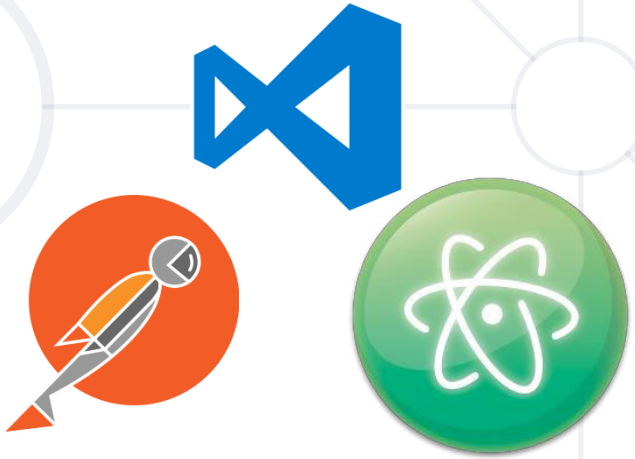
Electron.js

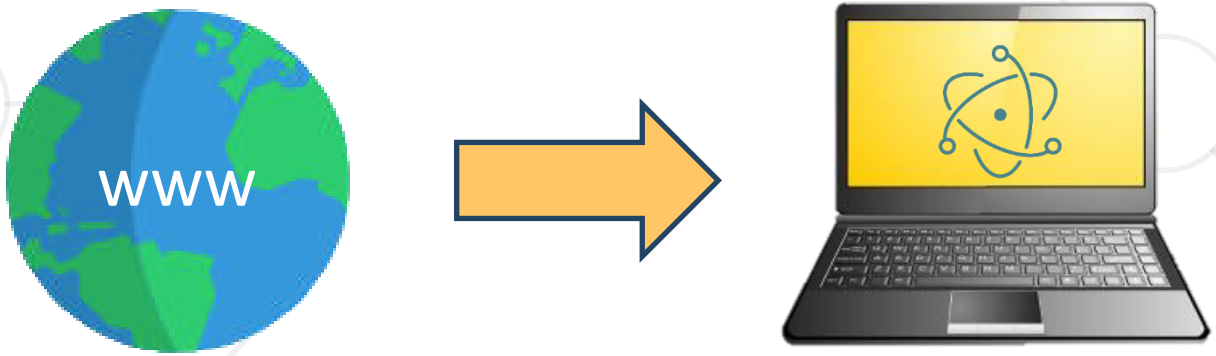
Javascript Framework for Desktop apps

Electron.js

SoftUni Foundation

- Open-source **framework** developed by GitHub
  - You can build **desktop** applications using the **Node.js** runtime and the **Chromium** web browser
- For the **view** rendering you can use the very same **HTML** and **CSS** you already use in the web browser
- Apps made with Electron.js
  - VS Code
  - Postman
  - Atom





Demo

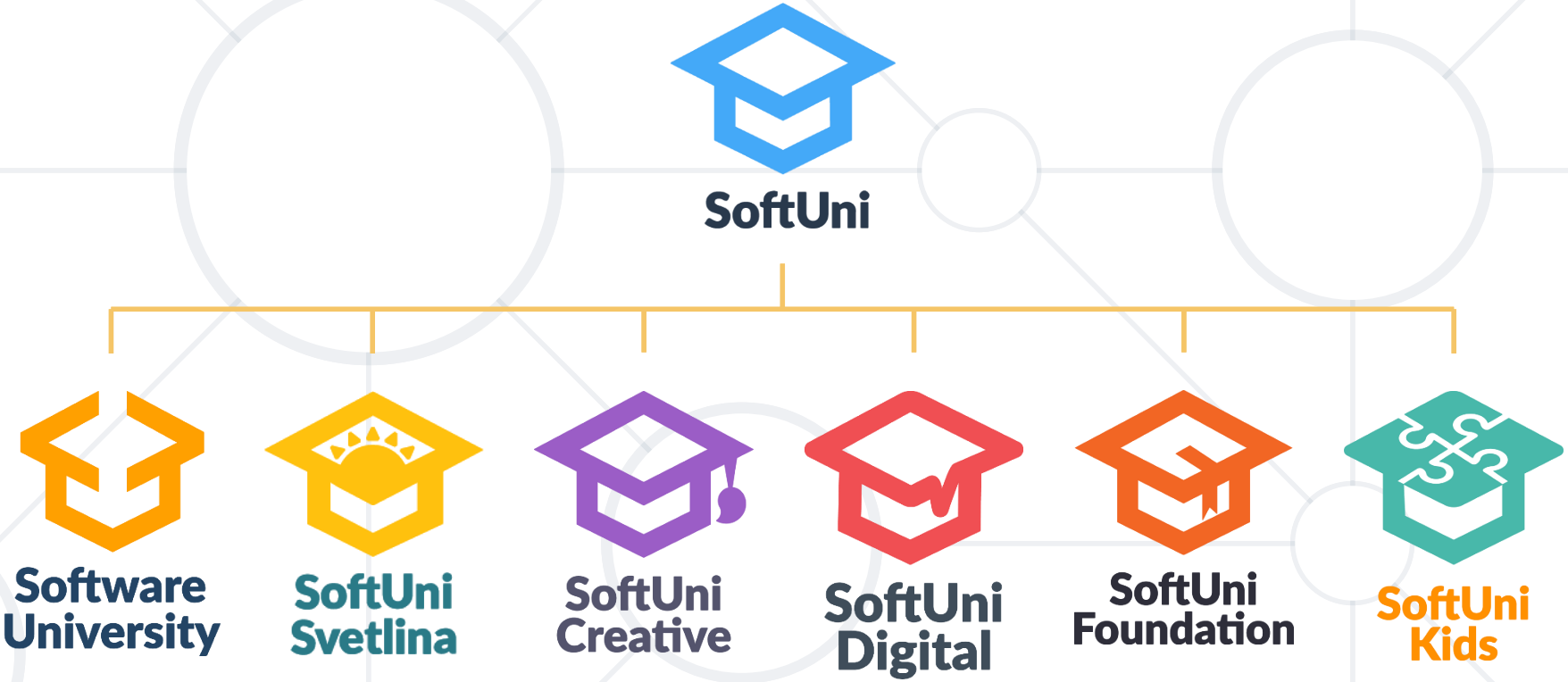
Deploy our Web App as Desktop App

Summary

SoftUni Foundation

- **Webpack**/Grunt/Gulp **tools** help with the deployment process of our application.
- **Lodash** is fast and reliable **library** which extends how we can work with objects, arrays etc.
- **ESLint** helps you follow the best practices for writing good code.
- **Electron.js** gives the functionality to create **desktop app** using **Javascript**

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SoftwareGroup
doing it right

NETPEAK

SmartIT

SUPER HOSTING .BG

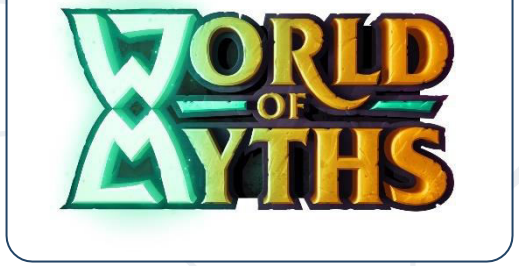
INDEAVR
Serving the high achievers

INFRAGISTICS®

LIEBHERR

SoftUni Organizational Partners

SoftUni Foundation



Trainings @ Software University (SoftUni)

SoftUni Foundation

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license