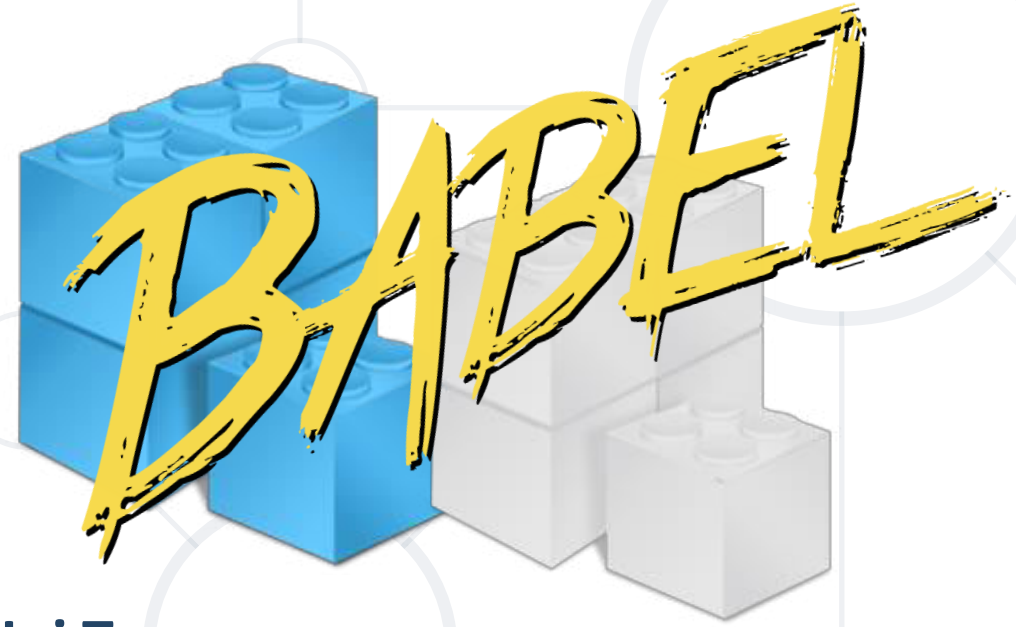
# Modules in JS

## Modules, Babel, RequireJS, Other JavaScript Module Systems




**SoftUni Team**

**Technical Trainers**

**Software University**

http://softuni.bg

# Table of Contents



# Have a Question?

SoftUni Foundation

sli.do

#JSCORE

# Modular Code

## Concepts

# What is a Module?

SoftUni Foundation

persons.js

```
class Person {
  constructor(name) {
    this.name = name;
  }
  toString() {
    return `I'm ${this.name}`;
  }
}
module.exports = Person;
```

app.js

```
let Person = require('./person');

let p = new Person('Pesho');
console.log(p.toString());
```

# Submitting Modules in the Judge

SoftUni Foundation

- SoftUni Judge (https://judge.softuni.bg) submissions should consis t of a zip file, containing all required files
- Attach members as requested by the problem description as prope rties of the **result** object:

```
let Person = require('./person');
result.Person = Person;
```

- The entry point of your solution is always **app.js**

Check your solution here: https://judge.softuni.bg/Contests/342

# Modular Code

- Reduced global scope pollution
- Improved performance - only load the code that is needed
- Easier to maintain
- Interchangeable code - use parts of one application in another

# IIFE

- An Immediately-Invoked Function Expression
  - Maintains hidden state
  - Limits the exposure of variables

```
let count = (function() {
  let counter = 0;
  return function() { counter++; }
})();
```

# Revealing Module Pattern

SoftUni Foundation

- Expands the IIFE concept
- Return objects with internal state and complex functionality

```
let revModule = (function() {
  let counter = 0; // private
  function increase(num) { counter += num; }
  function decrease(num) { counter -= num; }
  function value() { return count; }
  return { increase, decrease, value }; // public
})();
```

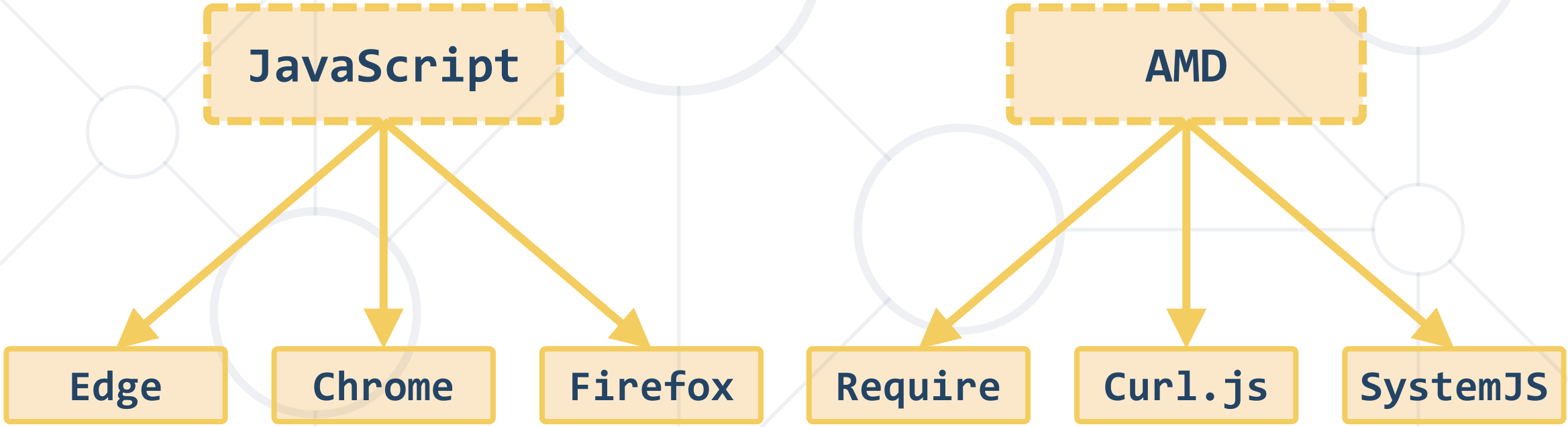

# Module Systems for JS

## Formats and Loaders

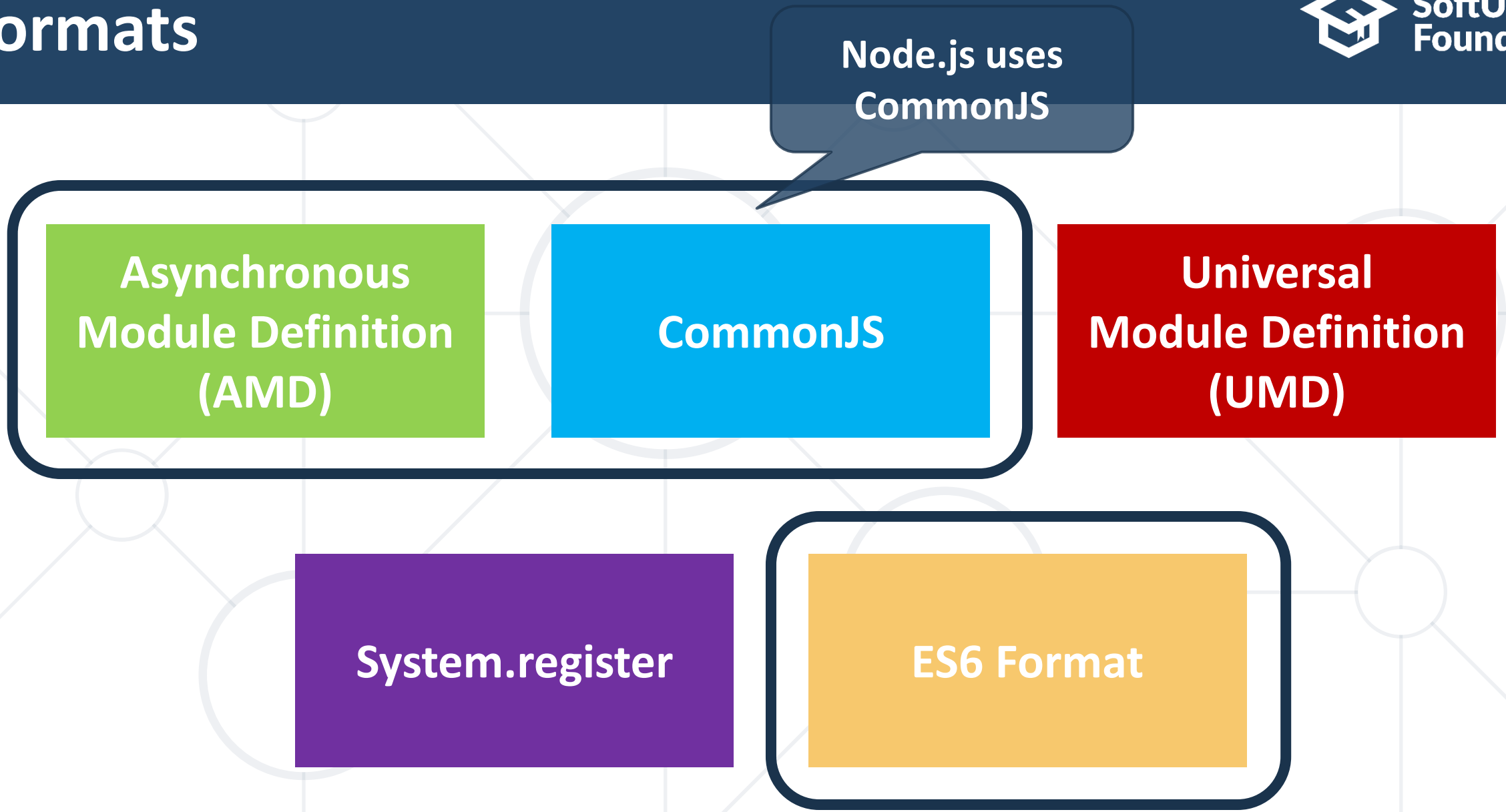
# Formats and Loaders

SoftUni Foundation

- Module Format → Syntax
- Module Loader → Execution



# Formats


Node.js uses CommonJS
Asynchronous Module Definition (AMD)
CommonJS
Universal Module Definition (UMD)
System.register
ES6 Format

# Loaders

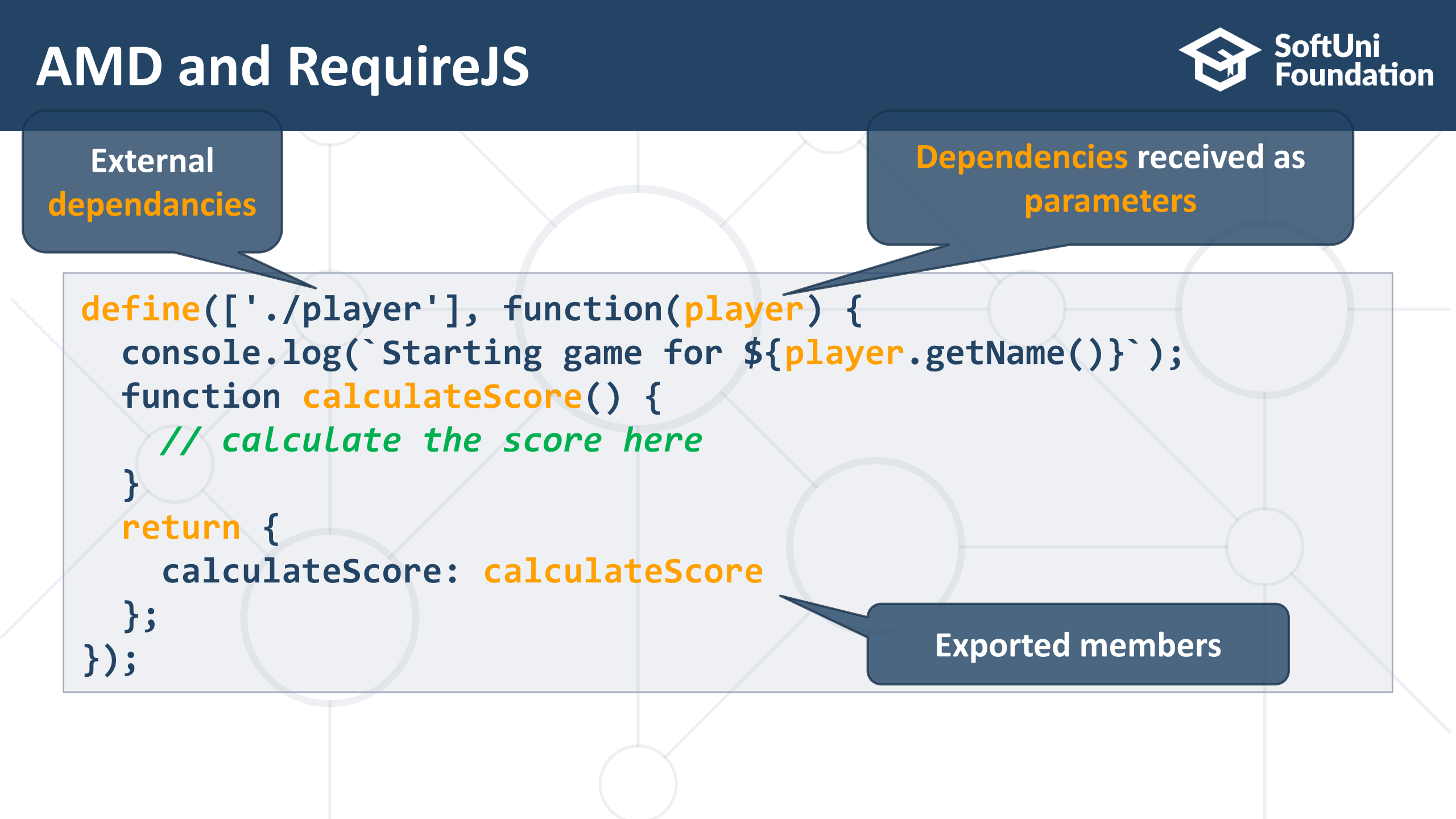
- RequireJS
  - Most popular for browsers
  - Works with AMD format


- SystemJS
  - Prefered on server-side implementations
  - Supports AMD, CommonJS, UMD, System.register
  - CommonJS is used by Node.js


# AMD and RequireJS

## Syntax and Configuration

# AMD and RequireJS




External dependancies

Dependencies received as parameters

```
define(['./player'], function(player) {
  console.log(`Starting game for ${player.getName()}`);
  function calculateScore() {
    // calculate the score here
  }
  return {
    calculateScore: calculateScore
  };
});
```

Exported members

# Installing RequireJS

SoftUni Foundation

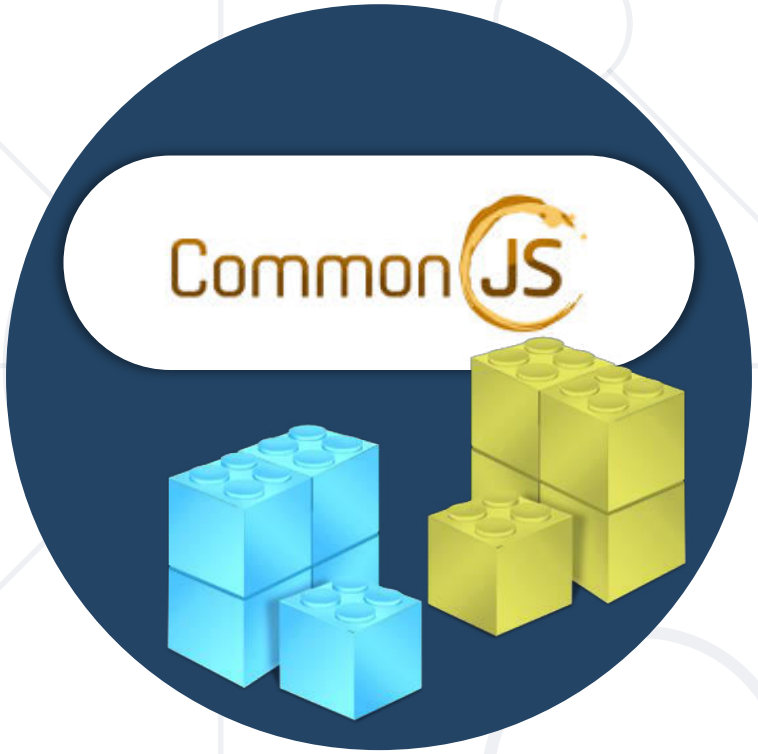
- Download RequireJS using WebStorm's terminal

```
npm install --save requirejs
```

- Or download from requirejs.org
- Use a script tag with **data-main** set to your app's path

```
<script data-main="./app.js"
src="node_modules/requirejs/require.js">
</script>
```

Note: It's best if your project has a **package.json** file.




# CommonJS and System.js

## Syntax and Configuration

# CommonJS and SystemJS

SoftUni Foundation

External dependancies

```
let player = require('./player.js');

console.log(`Starting game for ${player.getName()}`);
function calculateScore() {
  // calculate the score here
}

exports.calculateScore = calculateScore;
```

Exported members

```
// module.exports === exports
```

# Installing SystemJS

SoftUni Foundation

- Download SystemJS using WebStorm's terminal

```
npm install --save systemjs
```

- You can find documentation at SystemJS' github

- Node.js supports CommonJS format natively – you don't need to d ownload anything if you're writing a node module

Note: It's best if your project has a **package.json** file

# Configuring SystemJS

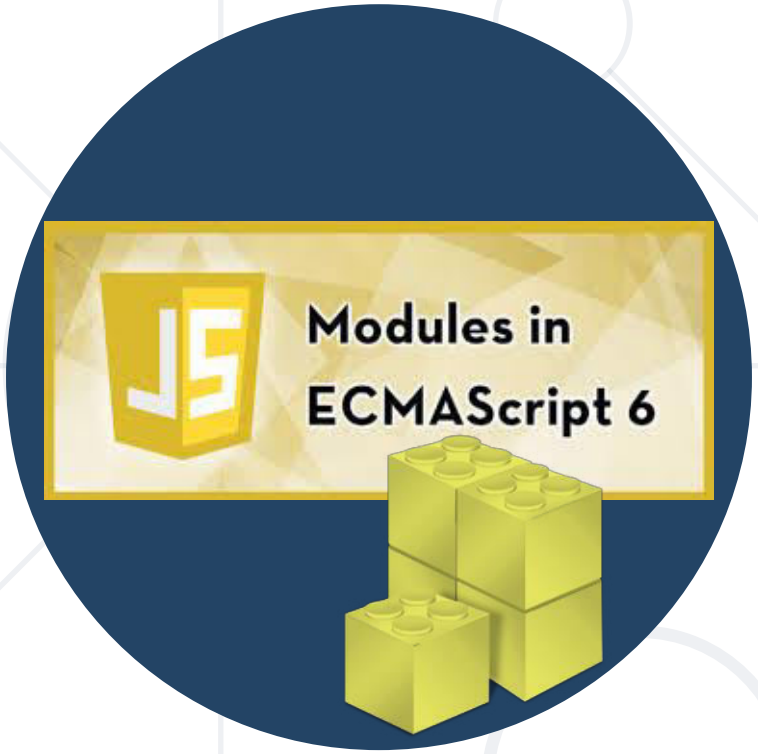
SoftUni Foundation

- Load the library in your host HTML

```
<script src="node_modules/systemjs/dist/system.js">
</script>
```

- Configure and load your app's path

```
<script>
  System.config({ meta: { format: 'cjs' } });
  System.import('./app.js');
</script>
```



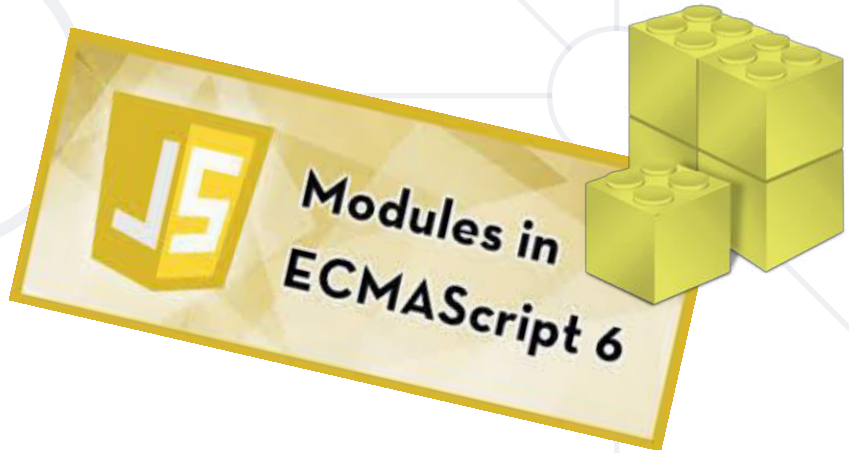

# ES6 (Harmony) Modules

## Syntax and Configuration

# ES6 Native Modules

SoftUni Foundation

- Functions similar to module formats
  - Dependency management
  - Hide from scope and expose explicitly
  - Used from Angular, React, Ember
- Some major differences
  - Cleaner syntax
  - Slightly better performance



# ES6 Export Syntax

SoftUni Foundation

- **export** → expose public API

```
export function updateScoreboard(newResult) { … }
export let homeTeam = 'Tigers';
```

- You can export multiple members

```
export { addResult, homeTeam as host };
```

- Default exports can be imported without a name

```
export default function addResult(newResult) { … }
```

# ES6 Import Syntax

SoftUni Foundation

- **import** → load dependency

```
import * as scoreboard from './scoreboard.js';
scoreboard.updateScore(); // call from module
```

- Import specific members

```
import {addResult, homeTeam} from './scoreboard.js';
addResult(); // call directly by name
```

- Import default member by specifying alias

```
import addResult from './scoreboard.js';
addResult(); // call directly by name
```

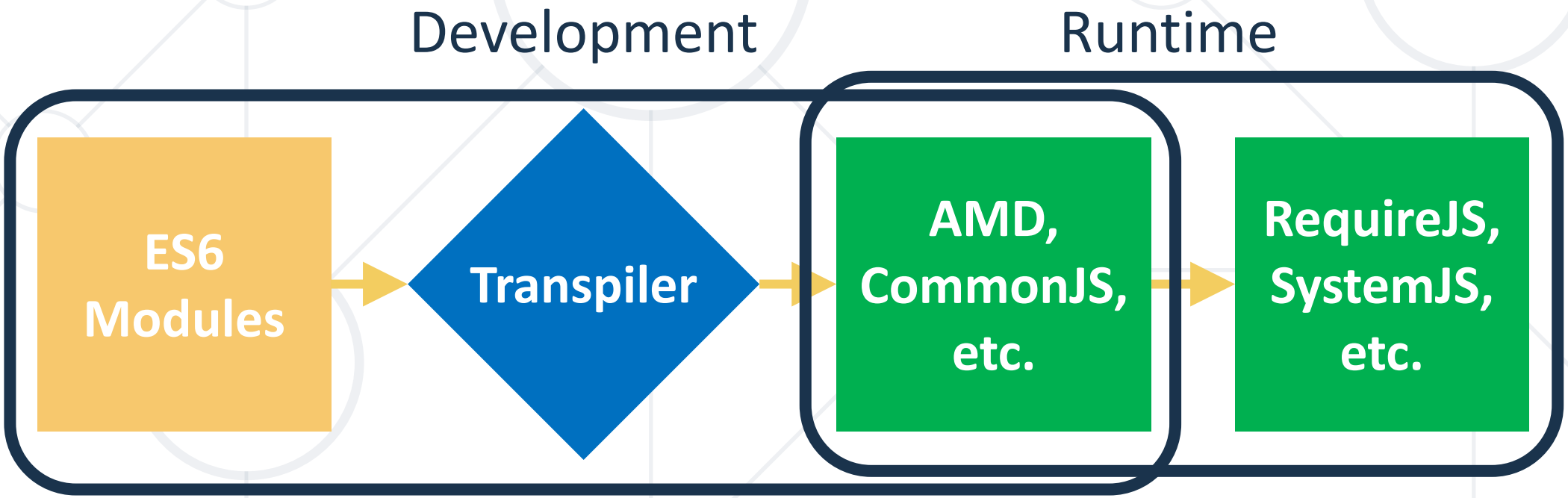

# Transpilers

## Using Babel to Transpile ES6 Modules

# Workflow with Transpilers

SoftUni Foundation

- Transpilers convert source code from one language to another
  - Or from one version to another of the same language

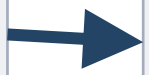


# Transpiling with Babel

SoftUni Foundation

- Babel allows us to use the latest JS features in today's browsers
  - Creates clean and readable code
  - Integrates seamlessly into WebStorm

```
export function getScore() {
  // get score
}
```



```
Object.defineProperty(
  exports, "__esModule",
  { value: true });
exports.getScore = getScore;
function getScore() {
  // get score
}
```

# Installing Babel

- Download Babel-CLI globally using WebStorm's terminal

```
npm install --save-dev babel-cli -g
```

- Configure WebStorm with the correct path to **babel.cmd**
- You can find documentation at **babeljs.io**
- Babel requires plug-ins to work

Note: It's best if your project has a **package.json** file

# Babel Configuration Cheat Sheet (2)



- Quick settings:
  - Program: (path to) node_modules\.bin\babel.cmd
  - Arguments: $FilePathRelativeToProjectRoot$ --source-maps --out-dir build
  - Working dir: $ProjectFileDir$
  - Output paths: build

# Configuring for AMD and RequireJS



- Download the plugin from WebStorm's terminal

```
npm install --save-dev babel-plugin-transform-es2015-modules-amd
```

- Create a **.babelrc** configuration file in the project's root

```
echo { "plugins": ["transform-es2015-modules-amd"] } > .babelrc
```

- To load the resulting files, you also need RequireJS installed

Note: It's best if your project has a **package.json** file

# Configuring for CommonJS and SystemJS


- Download the plugin from WebStorm's terminal

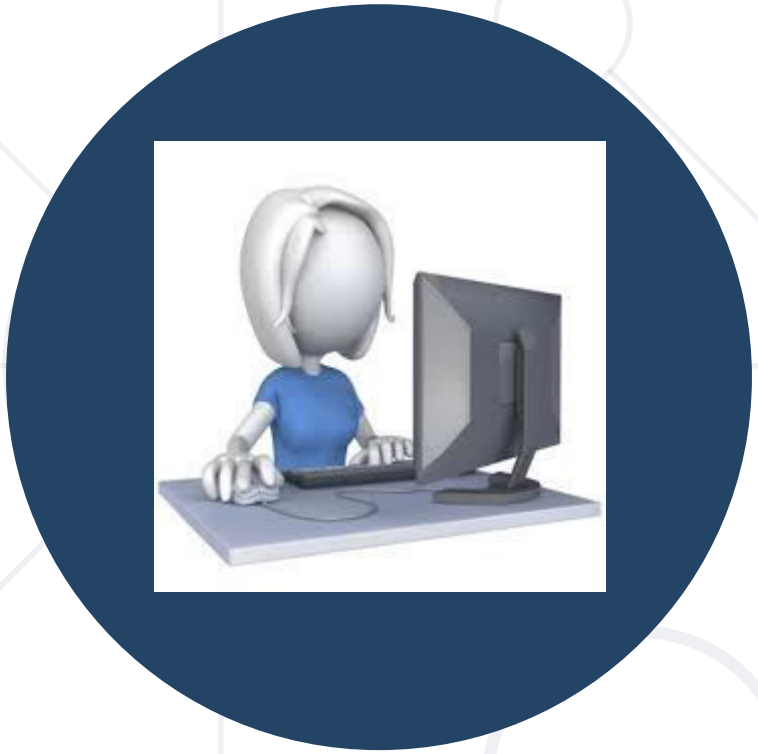
```
npm install --save-dev babel-plugin-transform-es2015-modules-commonjs
```

- Create a **.babelrc** configuration file in the project's root

```
echo { "plugins": ["transform-es2015-modules-commonjs"] } > .babelrc
```

- To load the resulting files, you also need SystemJS installed

Note: It's best if your project has a **package.json** file



# Live Exercises in Class (Lab)

## Practice: Writing Modular Code

# Summary



- Modular code improves workflow
- JS files can be loaded with external libraries
- ES6 gives us built-in module support

```
import addResult from './scoreboard';
export { homeTown as host };
```

- Use the latest features today with Transpilers

# Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

# SoftUni Diamond Partners

SoftUni Foundation



XSsoftware

SBTech

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

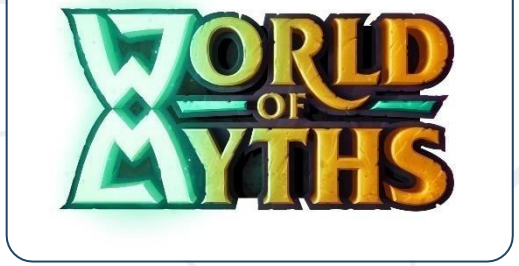
æternity

codeXio

# 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

SoftUni Foundation

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