Modules in ES2015

Brice Wilson

@brice_wilson www.BriceWilson.net

Overview

What are ES2015 modules?

Similarities and differences in relation to other module formats

Syntax for defining and using modules

Transpiling to earlier versions of JavaScript

Convert modules in the MultiMath app to use ES2015 modules

What are ES2015 modules?

“Native” modules

Support for modules built-in to JavaScript

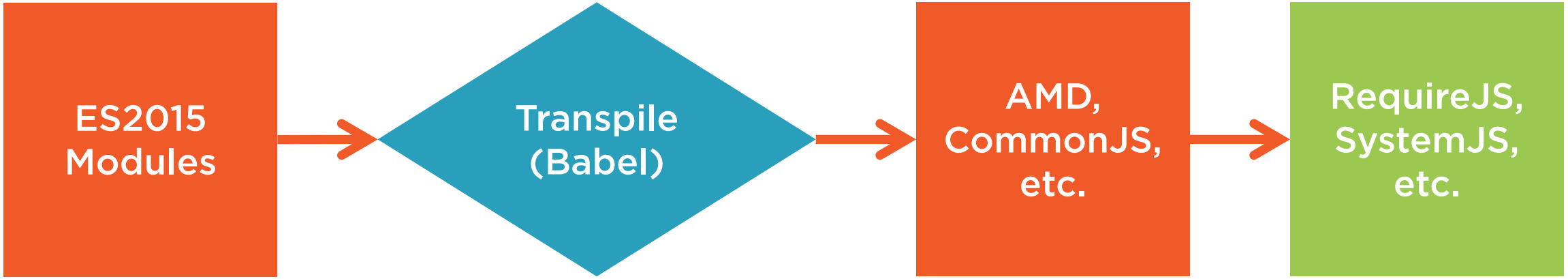
Similar to other module formats

- Support for dependency management
- Encapsulate implementation details
- Explicitly expose public API

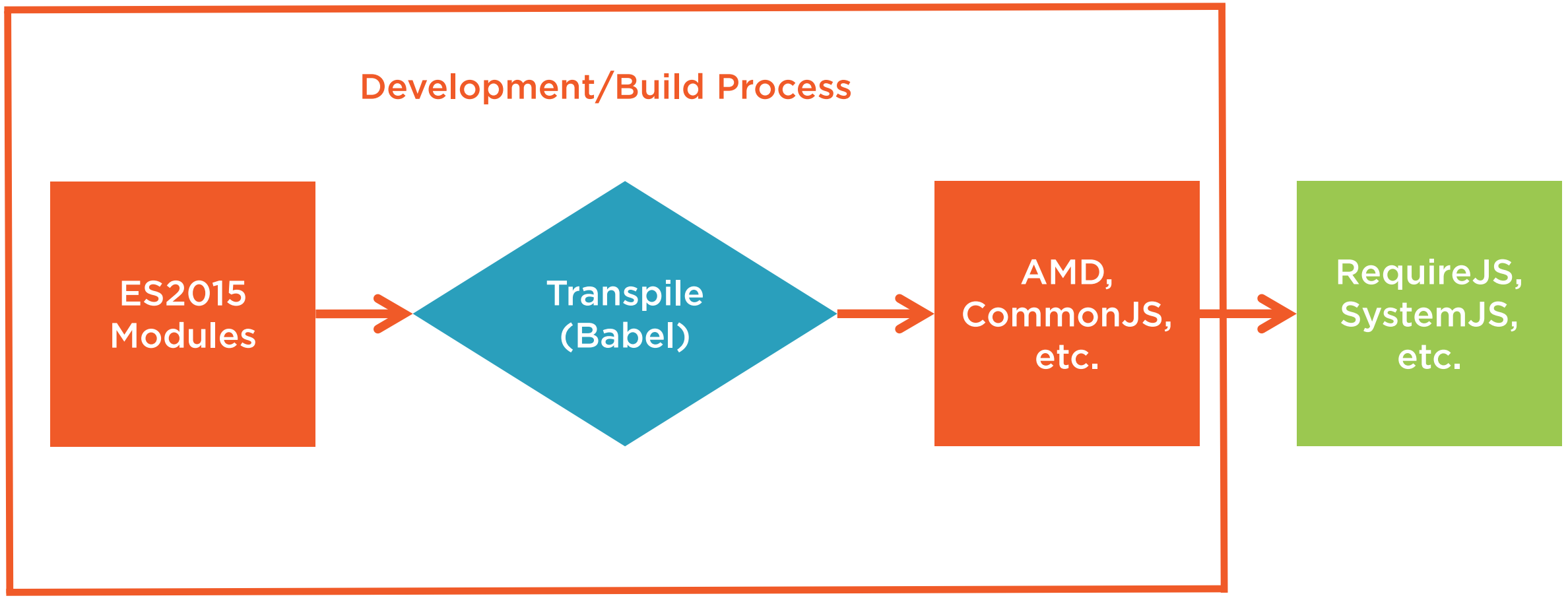
Different from other module formats

- No libraries required to create modules
- Currently must transpile to ES5
- Syntax

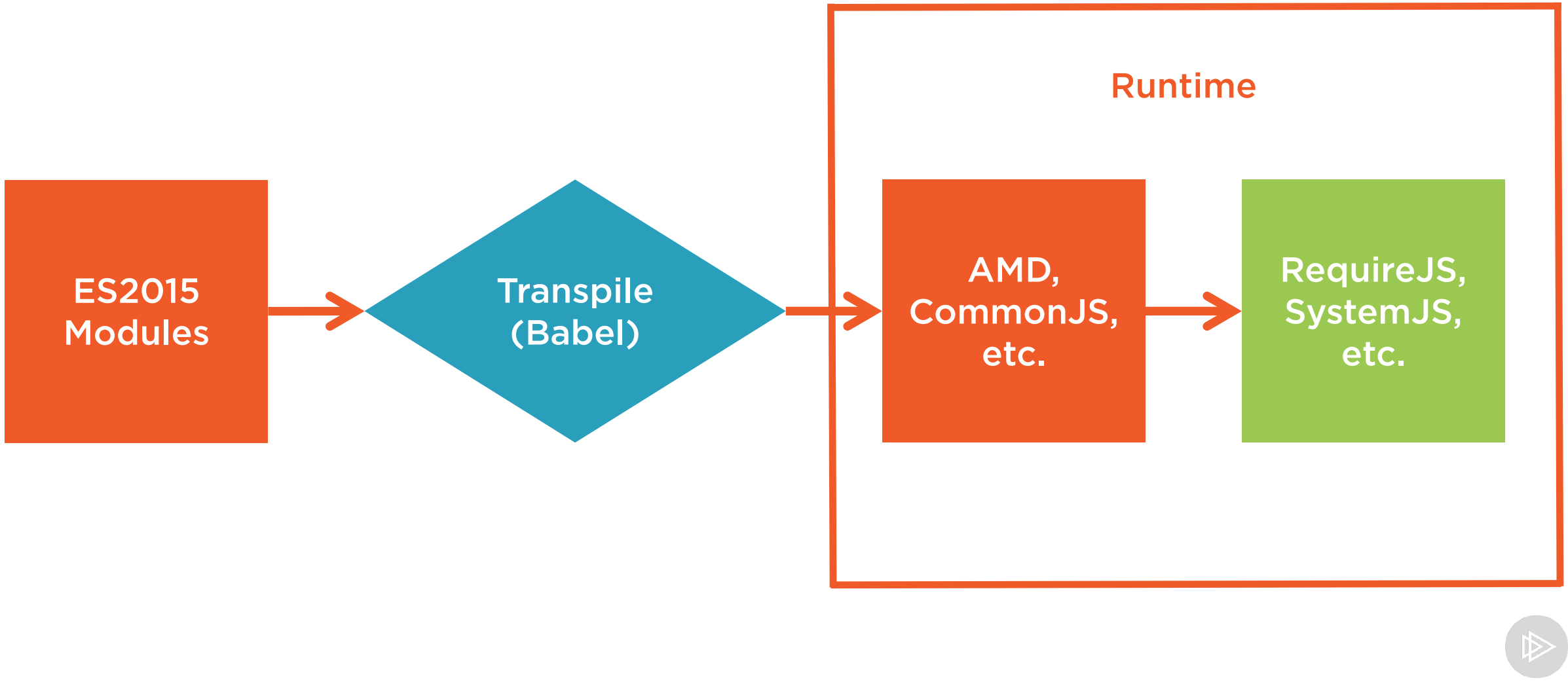
ES2015 Module Workflow



ES2015 Module Workflow



ES2015 Module Workflow



Importing and Exporting

Importing from a module

- Imported items are dependencies
- May import an entire module or just part of it
- May create an alias for imported items

Exporting from a module

- Exposes the API of the module
- May export items at declaration or all at once as a list
- May specify a “default” export

Named Exports

```
// scoreboard.js
export function addResult(newResult) {
    // add new result to list
}
```

Named Exports

```
// scoreboard.js
export function addResult(newResult) {
    // add new result to list
}
export function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
export var homeTeam = 'Tigers';
```

Named Exports

```
// scoreboard.js
function addResult(newResult) {
    // add new result to list
}
function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
var homeTeam = 'Tigers';

export { addResult, updateScoreboard as show, homeTeam };
```

Named Exports

```
// scoreboard.js
function addResult(newResult) {
    // add new result to list
}
function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
var homeTeam = 'Tigers';

export { addResult, updateScoreboard as show, homeTeam };
```

Named Exports

```
// scoreboard.js
function addResult(newResult) {
    // add new result to list
}
function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
var homeTeam = 'Tigers';

export { addResult, updateScoreboard as show, homeTeam };
```

Default Exports

```
// scoreboard.js
export default function addResult(newResult) {
    // add new result to list
}
export function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
export var homeTeam = 'Tigers';
```

Default Exports

```
// scoreboard.js
export default function addResult(newResult) {
    // add new result to list
}
export function updateScoreboard() {
    // update the scoreboard here
}
function somePrivateFunction() {
    // not part of the API
}
export var homeTeam = 'Tigers';
```

Import Syntax

```
// game.js

import * as scoreboard from './scoreboard.js';
```

Import Syntax

```
// game.js

import * as scoreboard from './scoreboard.js';
```

Import Syntax

```
// game.js

import * as scoreboard from './scoreboard.js';

scoreboard.updateScoreboard();
```

Import Syntax

```
// game.js

import * as scoreboard from './scoreboard.js';

scoreboard.updateScoreboard();

// ***********************************

// game.js

import { addResult, updateScoreboard } from './scoreboard.js';

updateScoreboard();
```

Import Syntax

```
// game.js

import * as scoreboard from './scoreboard.js';

scoreboard.updateScoreboard();

// ***********************************

// game.js

import { addResult, updateScoreboard } from './scoreboard.js';

updateScoreboard();
```

Import Syntax

```
import * as scoreboard from './scoreboard.js';

import { addResult, updateScoreboard } from './scoreboard.js';

import { updateScoreboard as update } from './scoreboard.js';
```

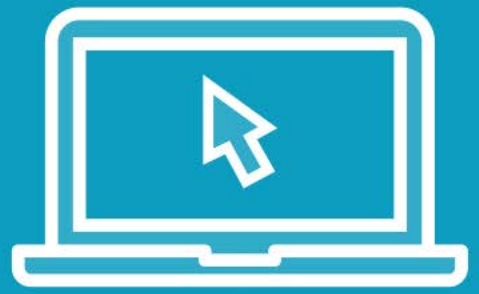
Import Syntax

```
import * as scoreboard from './scoreboard.js';

import { addResult, updateScoreboard } from './scoreboard.js';

import { updateScoreboard as update } from './scoreboard.js';

import newResult from './scoreboard.js’;
```

Import Syntax

```
import * as scoreboard from './scoreboard.js';

import { addResult, updateScoreboard } from './scoreboard.js';

import { updateScoreboard as update } from './scoreboard.js';

import newResult from './scoreboard.js';

import newResult, { updateScoreboard } from './scoreboard.js';
```

Demo

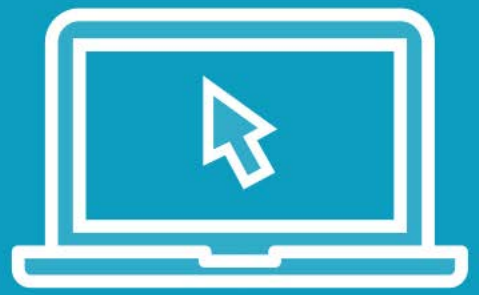


Creating ES2015 modules

What is Babel?

- Transpiler
- Supports most ES2015 features
- Executed as a build step
- Produces clean, readable JavaScript
- Highly configurable
- Supports all of the popular module formats

Demo



Transpiling ES2015 modules with Babel

Summary

ES2015 modules are the future

Importing modules

Exporting modules

Transpiling with Babel

Related courses

- *JavaScript Fundamentals for ES6*
- *Babel: Get Started*