



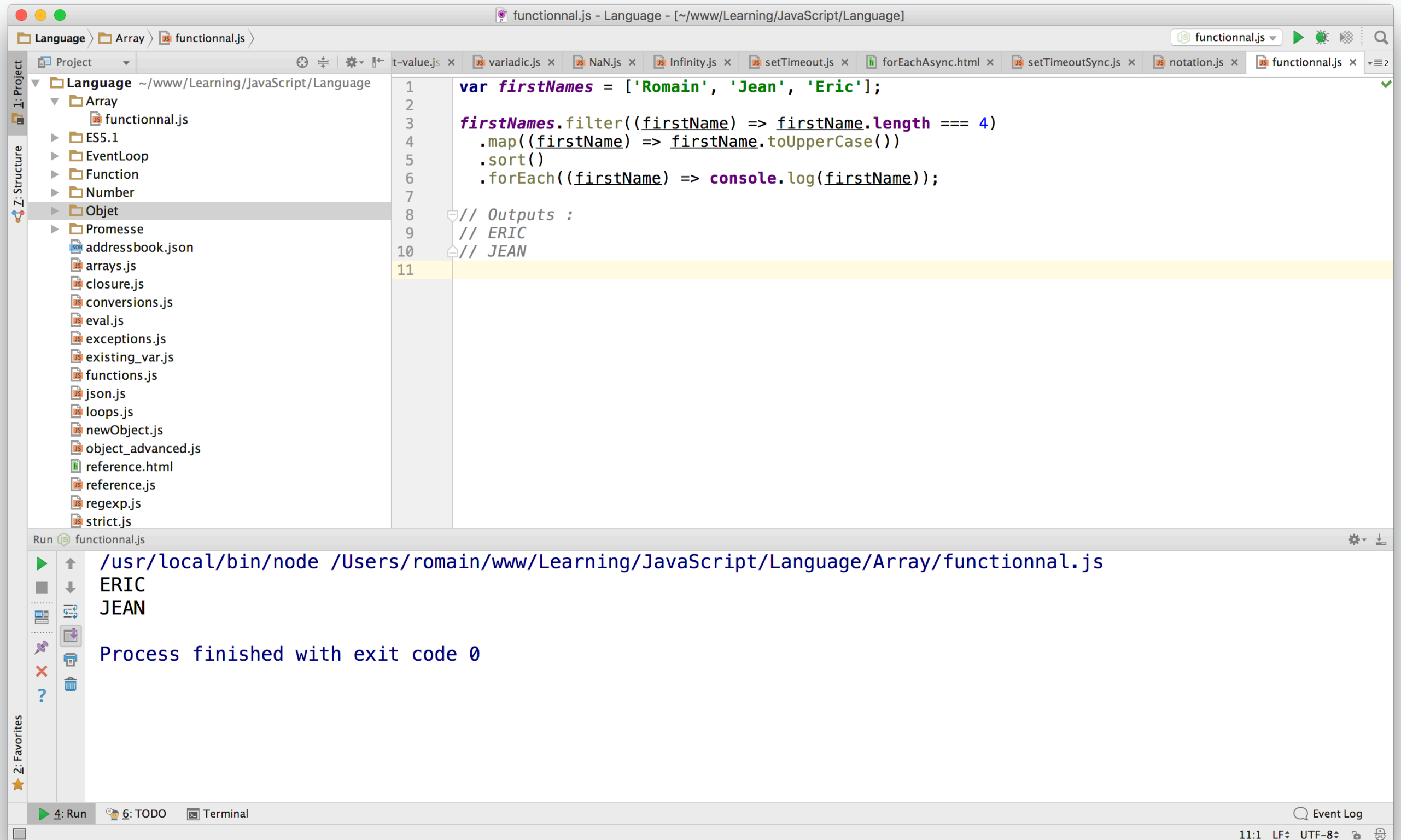
# JavaScript IDEs



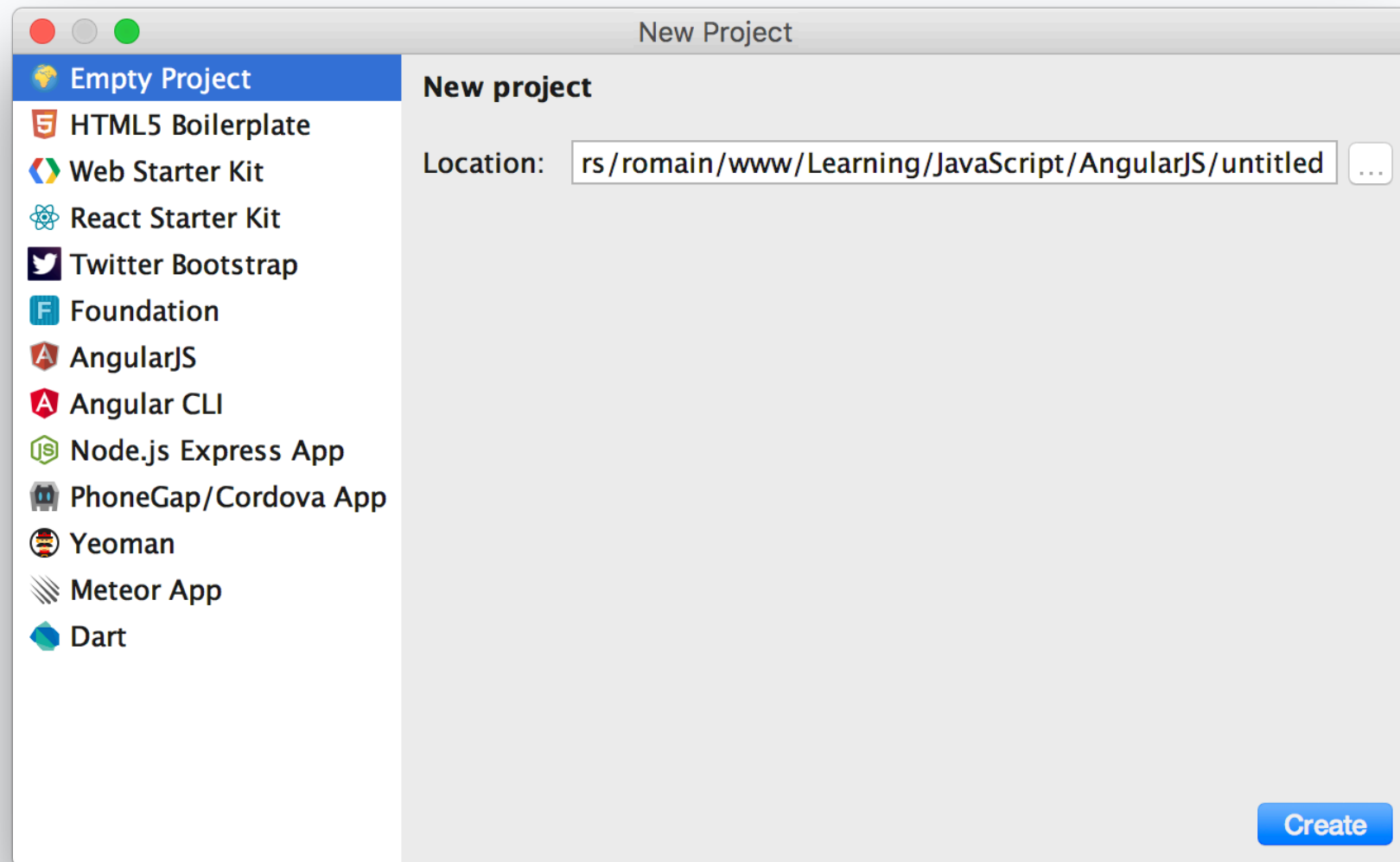
- Version orientée Web de IntelliJ IDEA de l'éditeur JetBrains  
<https://www.jetbrains.com/webstorm/>
- Licence : Commercial  
Licence entre 35 à 129 euros par an selon le profil et l'ancienneté.  
Version d'essai 30 jours.
- Plugins :  
Annuaire (642 en novembre 2016) : <https://plugins.jetbrains.com/webStorm>  
Langage de création : Java



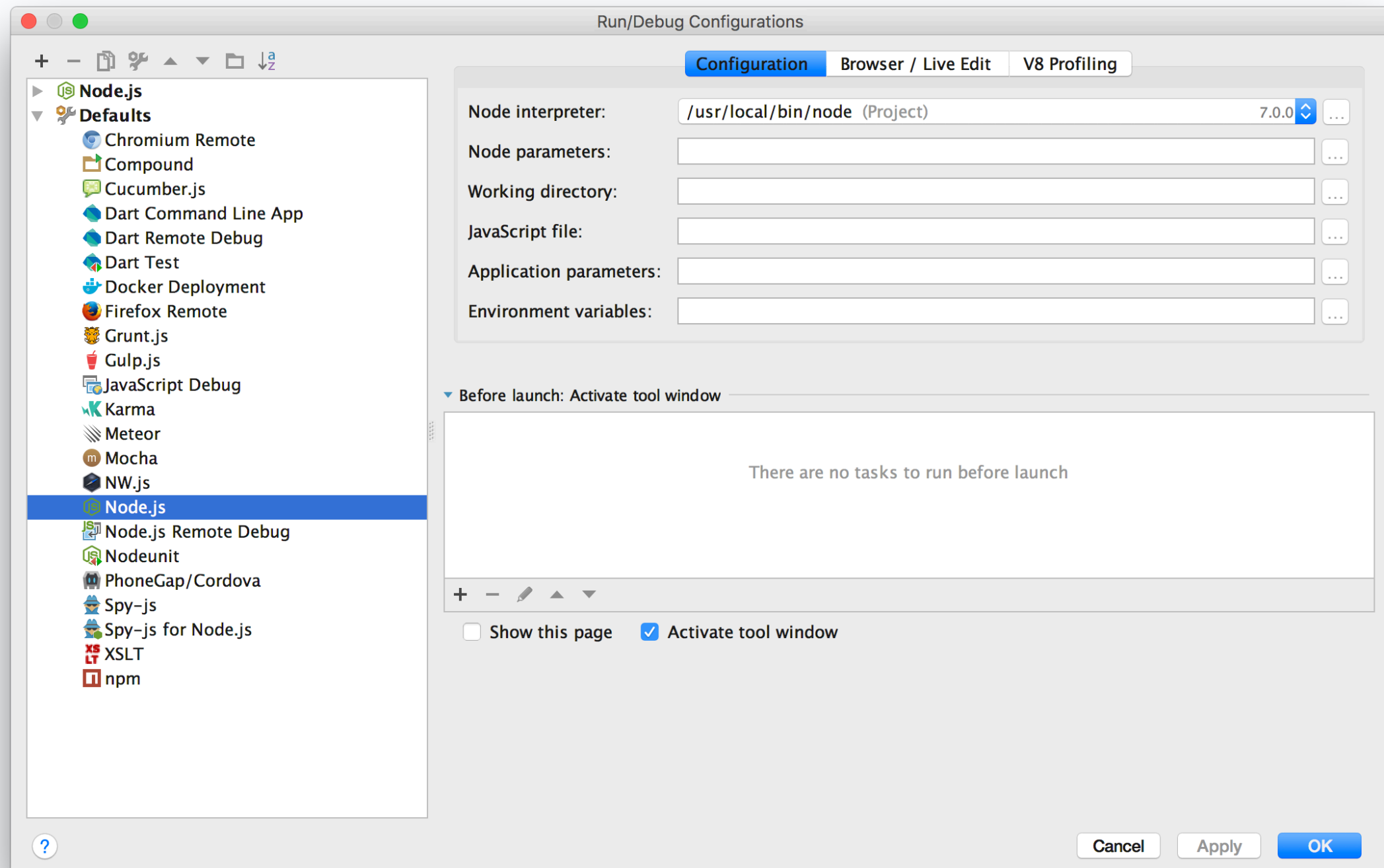
# JavaScript IDEs - Webstorm



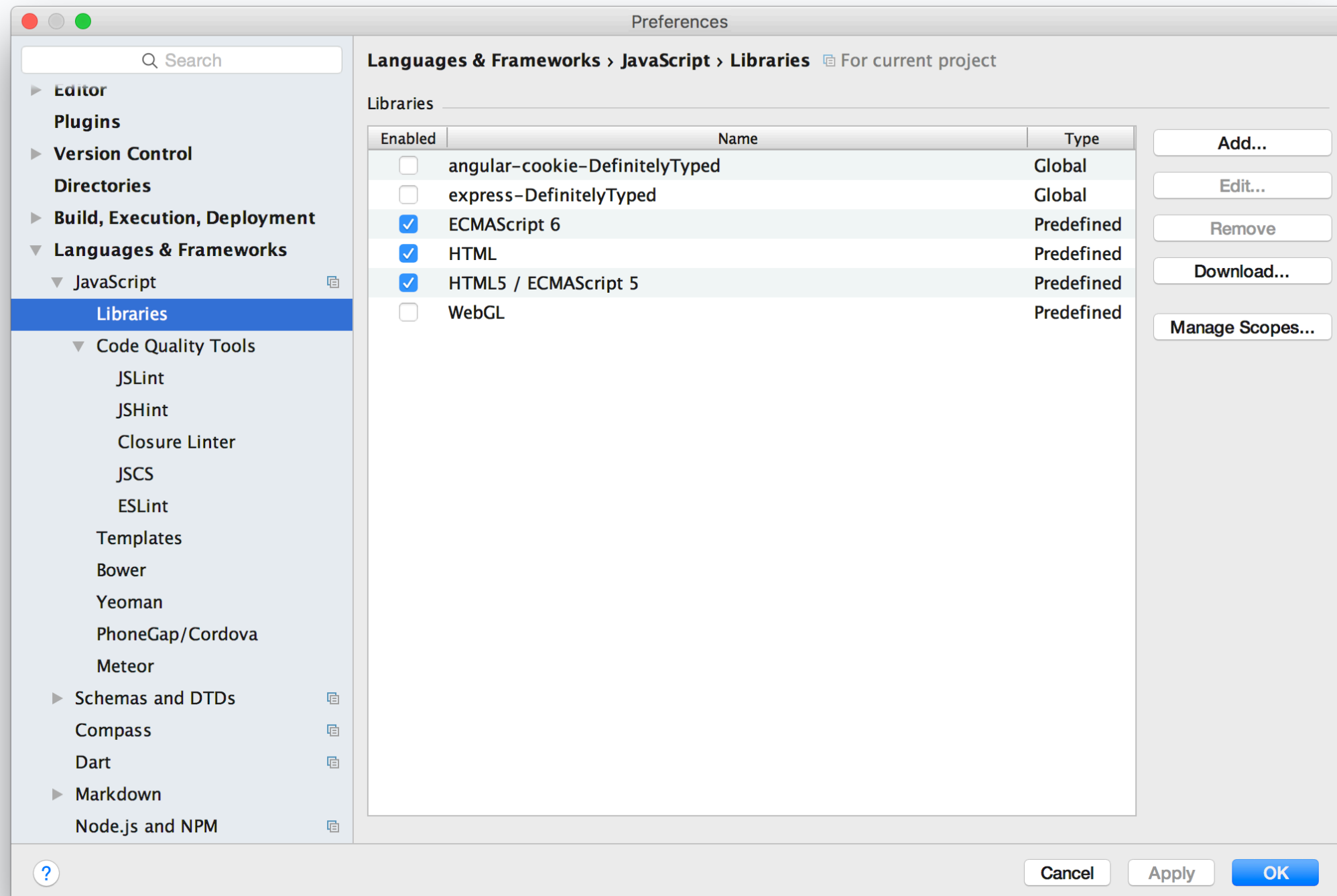
# JavaScript IDEs - Webstorm



# JavaScript IDEs - Webstorm



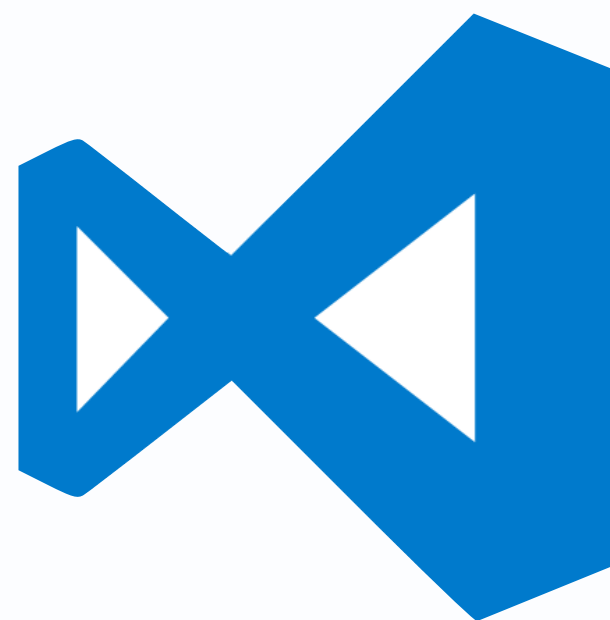
# JavaScript IDEs - Webstorm



# JavaScript IDEs - Visual Studio Code



- IDE créé par Microsoft, tourne sous Electron (Chromium + Node.js)  
<http://code.visualstudio.com/>
- Licence : MIT  
La licence open-source la plus permissive
- Plugins :  
Annuaire (1867 en novembre 2016) : <https://marketplace.visualstudio.com/VSCode>  
Langage de création : JavaScript sous Node.js
- Documentation  
<https://code.visualstudio.com/docs>



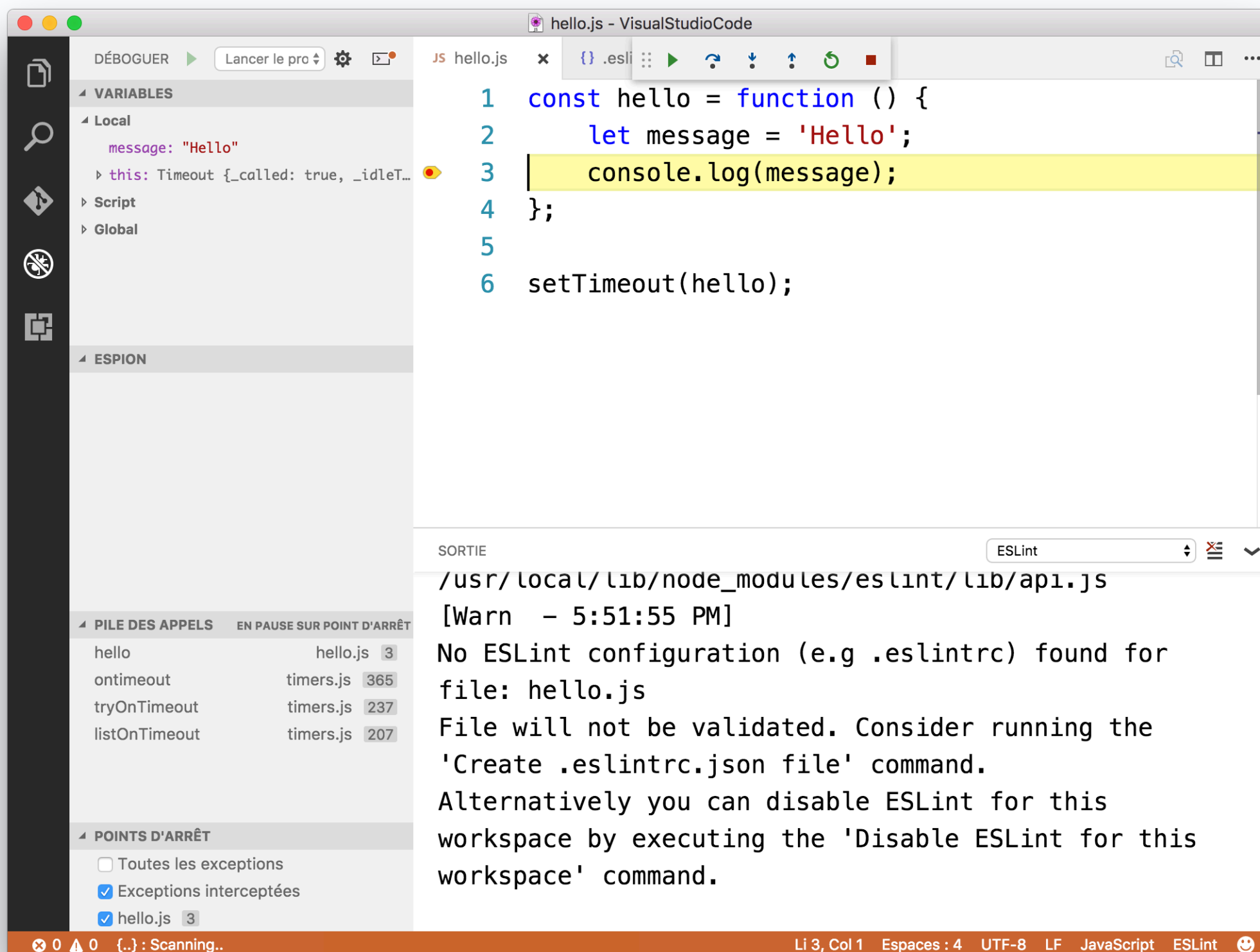
# JavaScript IDEs - Visual Studio Code

The image shows a screenshot of the Visual Studio Code editor interface. The left sidebar contains the 'EXPLORATEUR' (Explorer) view, which is divided into 'ÉDITEURS OUVERTS' (Open Editors) and 'UNIVERSAL-STARTER'. The 'UNIVERSAL-STARTER' section shows a file tree with folders like '.vscode', 'dist', 'client', 'server', 'src', 'assets', and 'backend'. The 'src/app/about' folder is expanded, showing files like 'about.component.ts', 'about.module.ts', and 'about-routing.module.ts'. The 'about.module.ts' file is selected and its content is displayed in the main editor area. The editor shows a TypeScript file with imports for '@angular/platform-browser', '@angular/core', and local modules. The code defines an '@NgModule' with imports, declarations, and providers, and exports a class 'AboutModule'. The bottom status bar shows 'master\*' and other project details.

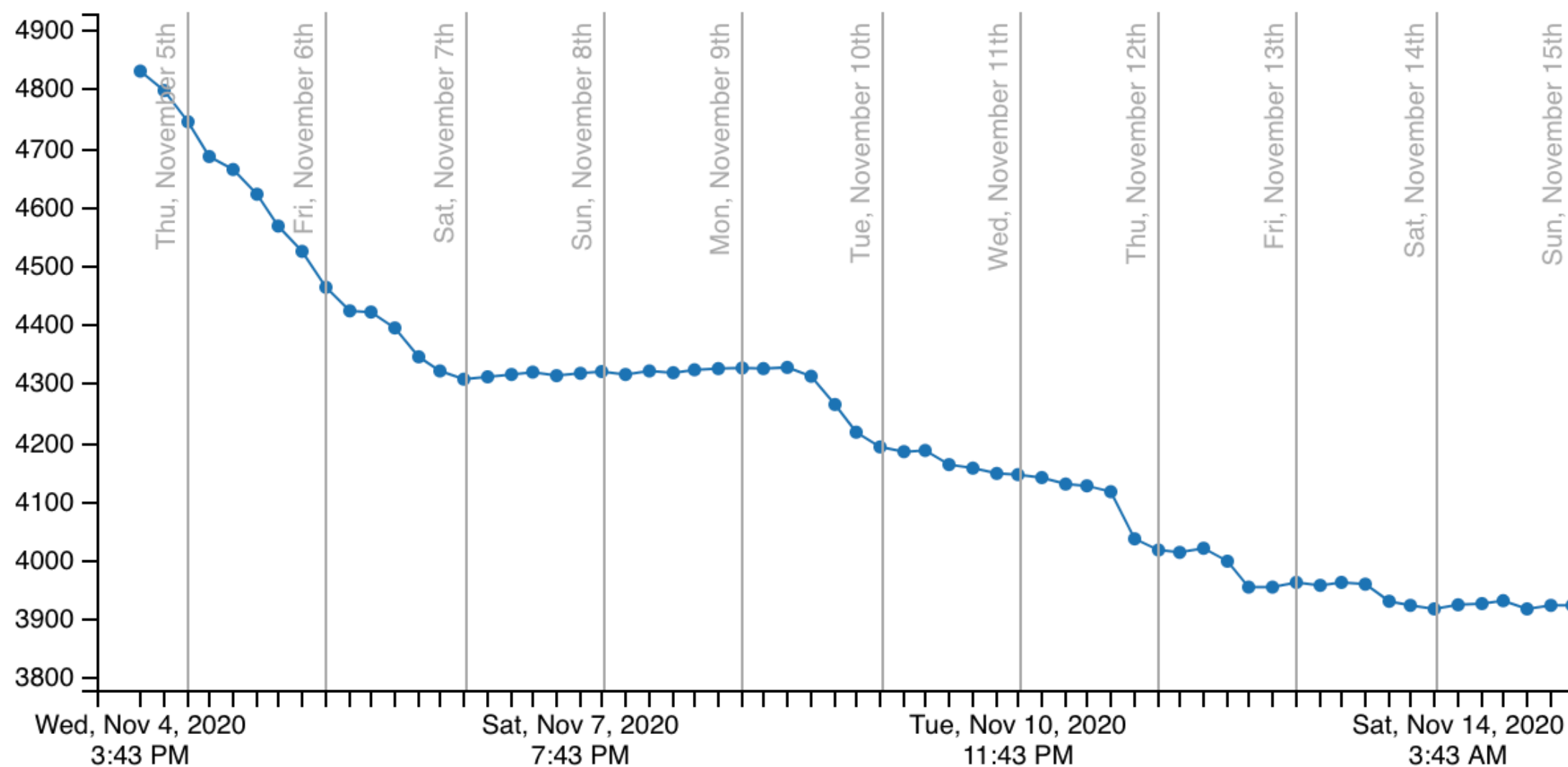
```
1 import { Title } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AboutComponent } from './about.component';
5 import { AboutRoutingModule } from './about-routing.module';
6
7 @NgModule({
8   imports: [
9     AboutRoutingModule
10  ],
11  declarations: [
12    AboutComponent
13  ],
14  providers: [
15    Title
16  ],
17 })
18 export class AboutModule { }
```



# JavaScript IDEs - Visual Studio Code



# JavaScript IDEs - Visual Studio Code



# JavaScript IDEs - Atom



- IDE créé par Github, tourne sous Electron (Chromium + Node.js)  
<https://atom.io>
- Licence : MIT  
La licence open-source la plus permissive
- Plugins :  
Annuaire (5232 en novembre 2016) : <https://atom.io/packages>  
Langage de création : JavaScript sous Node.js  
Exemples : atom-ternjs, linter, JavaScript Snippets, autocomplete+, autoprefixer...)



# JavaScript IDEs - Atom



The screenshot shows the Atom IDE interface. On the left is a sidebar with a file explorer showing a project structure. The main area is a code editor displaying an HTML file named `index.html`. The code is a mix of HTML and JavaScript, using a syntax-highlighted theme. The code includes comments for application metas, standalone emulation files, and UIKIT files. It also includes a JavaScript section that uses the `require` function to load a module and defines a widget with an `onLoad` event.

```
13 <!-- Application Metas End -->
14 <!-- Application Standalone emulation files -->
15 <link rel="stylesheet" href="../c/UWA/assets/css/standalone.css" />
16 <script src="../AmdLoader/AmdLoader.js"></script>
17 <script src="../c/UWA/js/UWA_Standalone_Alone.js"></script>
18
19 <!-- UIKIT files -->
20 <link rel="stylesheet" href="../UIKIT/UIKIT.css">
21 <script src="../UIKIT/UIKIT.js"></script>
22
23 <!-- Application JS Start -->
24 <script>
25   /* global widget, require */
26   require(['DS/ToDoDS/ToDoDS'], function(main) {
27     'use strict';
28
29     var myWidget = {
30
31       //The onLoad() function is the first one,
32       //it will be triggered by widget "onLoad" event.
33       onLoad: function() {
34
35         // Replaces body contents
36         //
37         //widget.body.innerHTML= "Hello World";
38         main(widget.body);
39       }
40     };
41
42     //The "onLoad" event is the very first event triggered when
43     // the widget is fully loaded.
44     widget.addEvent('onLoad', myWidget.onLoad);
45   });
46 </script>
47 <!-- Application JS End -->
48 </head>
49 <body>
```

The status bar at the bottom shows 'File 0', 'Project 0', 'No Issues', 'ToDoDS/index.html', '1:1', '1 deprecation', 'UTF-8', 'HTML', and '1 update'.



- Permet de standardiser les configs des IDEs sur l'indentation et les retours à la ligne <http://editorconfig.org>
- Supporté par la plupart des IDE
- Il suffit de créer un fichier .editorconfig à la racine d'un projet

```
# EditorConfig is awesome: http://EditorConfig.org

# top-most EditorConfig file
root = true

# Unix-style newlines with a newline ending every file
[*]
end_of_line = lf
insert_final_newline = true
charset = utf-8
indent_style = space
indent_size = 4

# HTML + JS files
[*.{html,js}]
indent_size = 2
```