

# Angular

Créer des applications web front-end

# Démarrage

## Installer l'environnement sur votre machine

Pour coder avec Angular sur votre machine, on a besoin d'installer 3 outils:

- Node.js (environnement d'exécution Javascript dont se sert Angular)
- NPM (node Package Manager)
- Angular CLI (pour utiliser les commande ng d'Angular)

[www.nodejs.org](http://www.nodejs.org)

[www.angular.io](http://www.angular.io)

# 1 Créer une nouvelle application Angular

- Ouvrir le terminal dans le dossier où l'on veut créer le dossier d'application
- Syntaxe de la commande pour créer une nouvelle app Angular

`ng new <nomDeVotreApp> --options`

Exemple : `ng new angulardemo`

## 2 Lancer l'application dans le navigateur

- Ouvrir le dossier de l'application dans VS Code ou votre éditeur de code
- Depuis le terminal, exécuter la commande suivante :

**ng serve** cette commande va compiler notre application

**ng serve --open** cette également ouvrir l'application dans le navigateur

# 3 Architecture d'un dossier Angular (package.json)

```
(1) package.json x
1 {
2   "name": "angulardemo",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "test": "ng test",
9     "lint": "ng lint",
10    "e2e": "ng e2e"
11  },
12  "private": true,
13  "dependencies": {
14    "@angular/animations": "~7.2.0",
15    "@angular/common": "~7.2.0",
16    "@angular/compiler": "~7.2.0",
17    "@angular/core": "~7.2.0",
18    "@angular/forms": "~7.2.0",
19    "@angular/platform-browser": "~7.2.0",
20    "@angular/platform-browser-dynamic": "~7.2.0",
21    "@angular/router": "~7.2.0",
22    "core-js": "~2.5.4",
23    "rxjs": "~6.3.3",
24    "tslib": "^1.9.0",
25    "zone.js": "~0.8.26"
26  },
27  "devDependencies": {
28    "@angular-devkit/build-angular": "~0.13.0",
29    "@angular/cli": "~7.3.1",
30    "@angular/compiler-cli": "~7.2.0",
31    "@angular/language-service": "~7.2.0",
32    "@types/node": "~8.9.4",
33    "@types/jasmine": "~2.8.8",
34    "@types/jasminewd2": "~2.0.3",
35    "codelyzer": "~4.5.0",
36    "jasmine-core": "~2.99.1",
37    "jasmine-spec-reporter": "~4.2.1",
38    "karma": "~3.1.1",
39    "karma-chrome-launcher": "~2.2.0",
40    "karma-coverage-istanbul-reporter": "~2.0.1",
41    "karma-jasmine": "~1.1.2",
42    "karma-jasmine-html-reporter": "~0.2.2",
43    "protractor": "~5.4.0",
44    "ts-node": "~7.0.0",
45    "tslint": "~5.11.0",
46    "typescript": "~3.2.2"
47  }
48 }
49
```

## Le fichier package.json

Il est lié à NPM.

Il centralise TOUTES les dépendances dont notre application se sert pour fonctionner.

Il se décompose en 3 clés principales :

- scripts  
les scripts npm utilisables dans notre projet
- dependencies  
On y retrouve les dépendances de notre projet
- devDependencies  
On y retrouve les dépendances de développement de notre projet

# 3 Architecture d'un dossier Angular (angular.json)

```
"projects": {
  "angulardemo": {
    "root": "",
    "sourceRoot": "src",
    "projectType": "application",
    "prefix": "app",
    "schematics": {},
    "architect": {
      "build": {
        "builder": "@angular-devkit/build-angular:browser",
        "options": {
          "outputPath": "dist/angulardemo",
          "index": "src/index.html",
          "main": "src/main.ts",
          "polyfills": "src/polyfills.ts",
          "tsConfig": "src/tsconfig.app.json",
          "assets": [
            "src/favicon.ico",
            "src/assets"
          ],
          "styles": [
            "src/styles.css"
          ],
          "scripts": [],
          "es5BrowserSupport": true
        }
      }
    }
  }
}
```

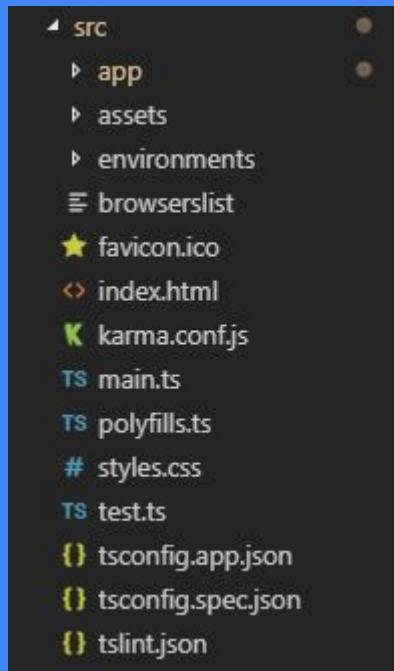
## Le fichier angular.json

C'est un fichier de configuration de notre projet  
Il permet de nommer les paths des fichiers et dossiers importants et de déterminer la configuration globale.

Exemples :

- sourceRoot : src  
le dossier où toute notre application se trouve
- outputPath: dist/application  
Le dossier de sortie du build de l'application
- index.html  
Le point d'entrée de l'application
- le tableau des styles et des scripts qui seront ajoutés de façon globale à notre application

# 3 Architecture d'un dossier Angular (src)



## Le dossier src

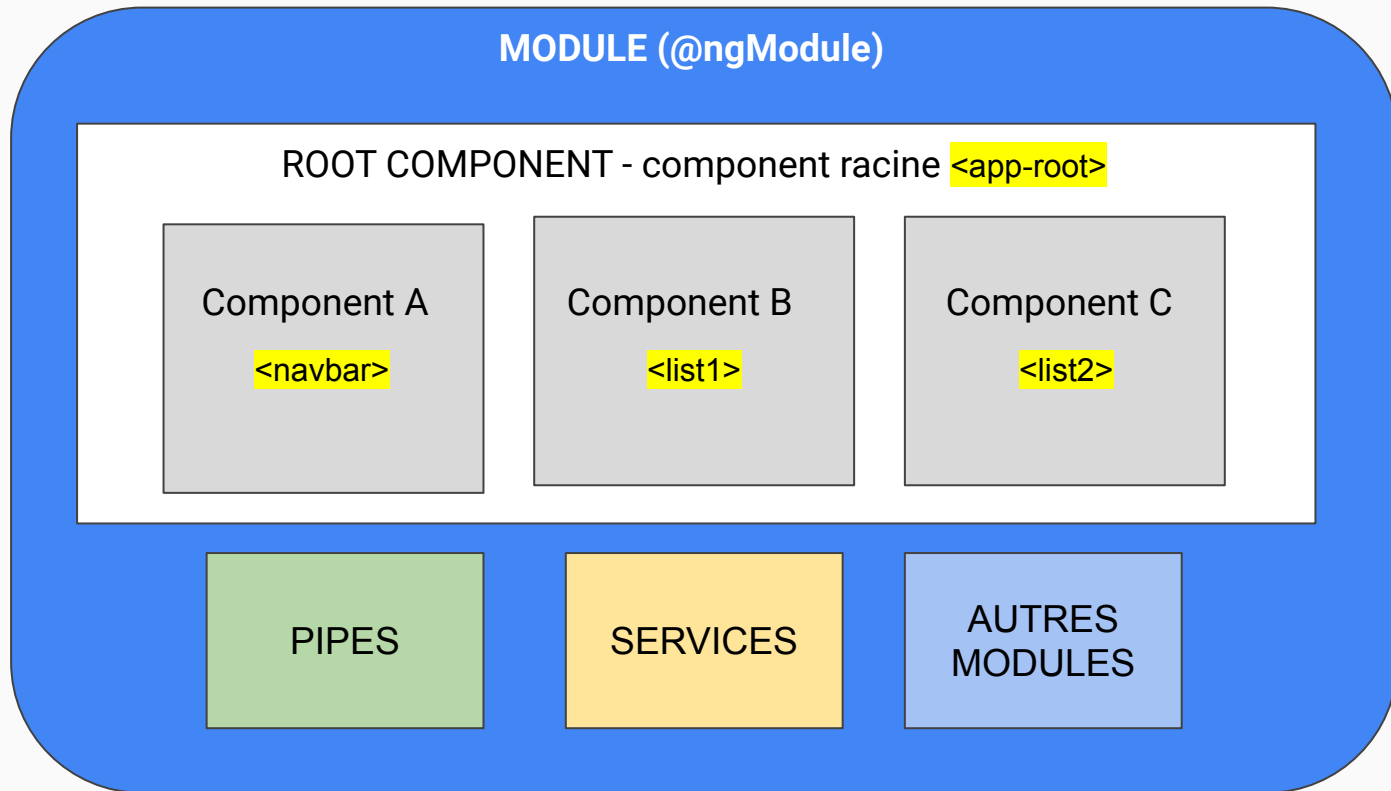
C'est dans ce dossier que se trouve toute l'application. Parmi ces fichiers et dossiers importants :

- index.html  
Le point d'entrée de l'application
- styles.css  
le fichier de styles qui s'appliquera de façon globale à tout l'application
- le dossier src/app  
C'est dans ce dossier que nous passeront 99% de notre temps à coder notre application. On y retrouve :
  - Le **component root** (le component app racine)
  - Les **components** de notre application
  - Les **services**
  - Les **pipes**

# 4 MODULES / COMPONENTS : comment ça marche ?

Un Module est un élément principal de l'application. Il déclare les dépendances :

- Les Components
- Les Pipes
- Les Services
- D'autres Modules
- etc...





## 5 CLI ANGULAR

Scaffold	Usage
Component	<code>ng g component my-new-component</code>
Directive	<code>ng g directive my-new-directive</code>
Pipe	<code>ng g pipe my-new-pipe</code>
Service	<code>ng g service my-new-service</code>
Class	<code>ng g class my-new-class</code>
Guard	<code>ng g guard my-new-guard</code>
Interface	<code>ng g interface my-new-interface</code>
Enum	<code>ng g enum my-new-enum</code>
Module	<code>ng g module my-module</code>

### Quelques commandes utiles avec le CLI Angular

- Créer, et compiler l'application
  - **ng new demo** (créer l'application demo)
  - **ng serve --open** (lancer l'app dans le navigateur)
  - **ng build** (compiler l'app pour la production)
- Générer des composants, des services, des pipes
  - **ng generate component product** ou **ng g c product**
  - **ng generate service user** ou **ng g s user**
  - **ng generate pipe pluralize** ou **ng g p user**

<https://www.npmjs.com/package/@angular/cli>