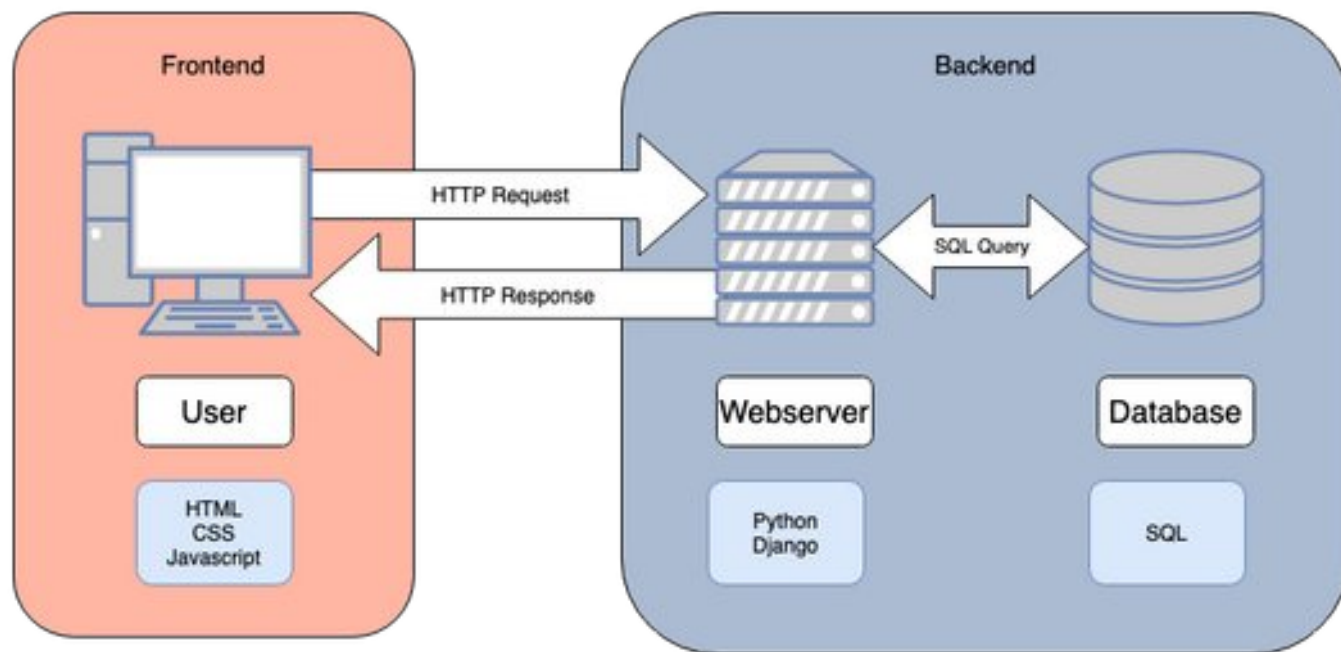




Technologie du web

Frontend vs Backend



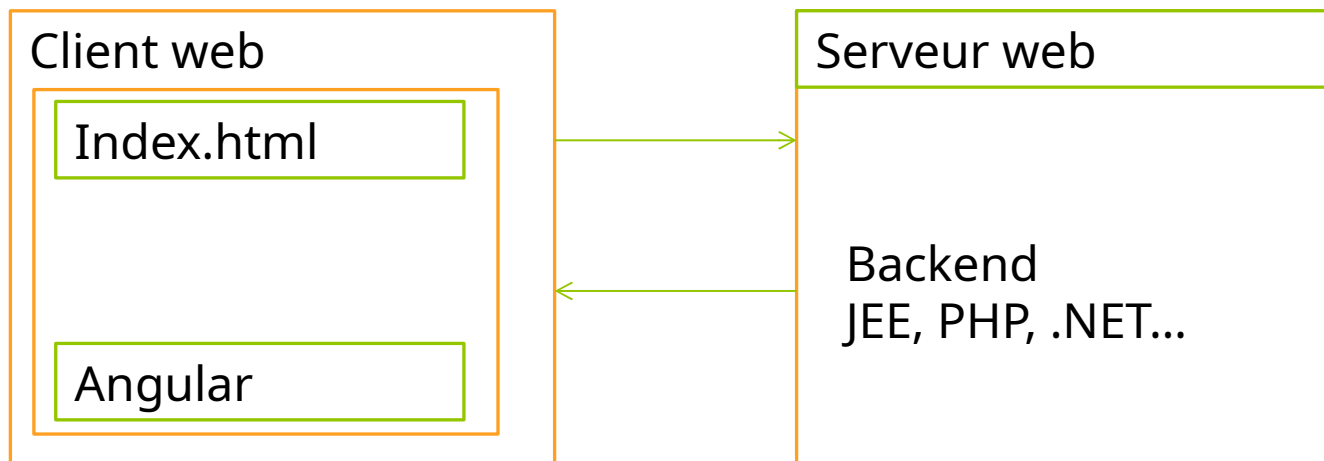
Framework

- Réduire la complexité de développement
- Gain de temps et d'efficacité
- La richesse en animations
- Meilleure structuration du code
- Maintenance simplifié

Angular

- un framework côté client, open source, et codirigé par l'équipe du projet « Angular » à Google
- Créer la partie frontend des applications web de type SPA (Single Page Application)

➡ contient une seule page html



Angular

- Angular 1 (AngularJS)
 - Première version d'angular
 - Les apps angularjs sont écrites en javascript
- Angular 2 (Angular)
 - Une réécriture complète d' angularjs plus performante plus structuré
 - Les apps d'angular sont écrites en typescript qui est compilé et traduit en javascript et puis il est exécuté par les browsers web
 - Il est basé sur les composants web
 - Puis il y a angular 4, 5 ...

Installation des outils

- Node js est une plateforme logicielle libre en JavaScript
- Contient l'outil npm (Node Package Manager) pour installer les différents package et les dependances: télécharge et installe les les bibliothèques javascript
- Angular CLI (Command Line Interface) offre un ensemble d'outils pour gènerer, compiler, tester et déployer les projets Angular

Création d'un nouveau projet

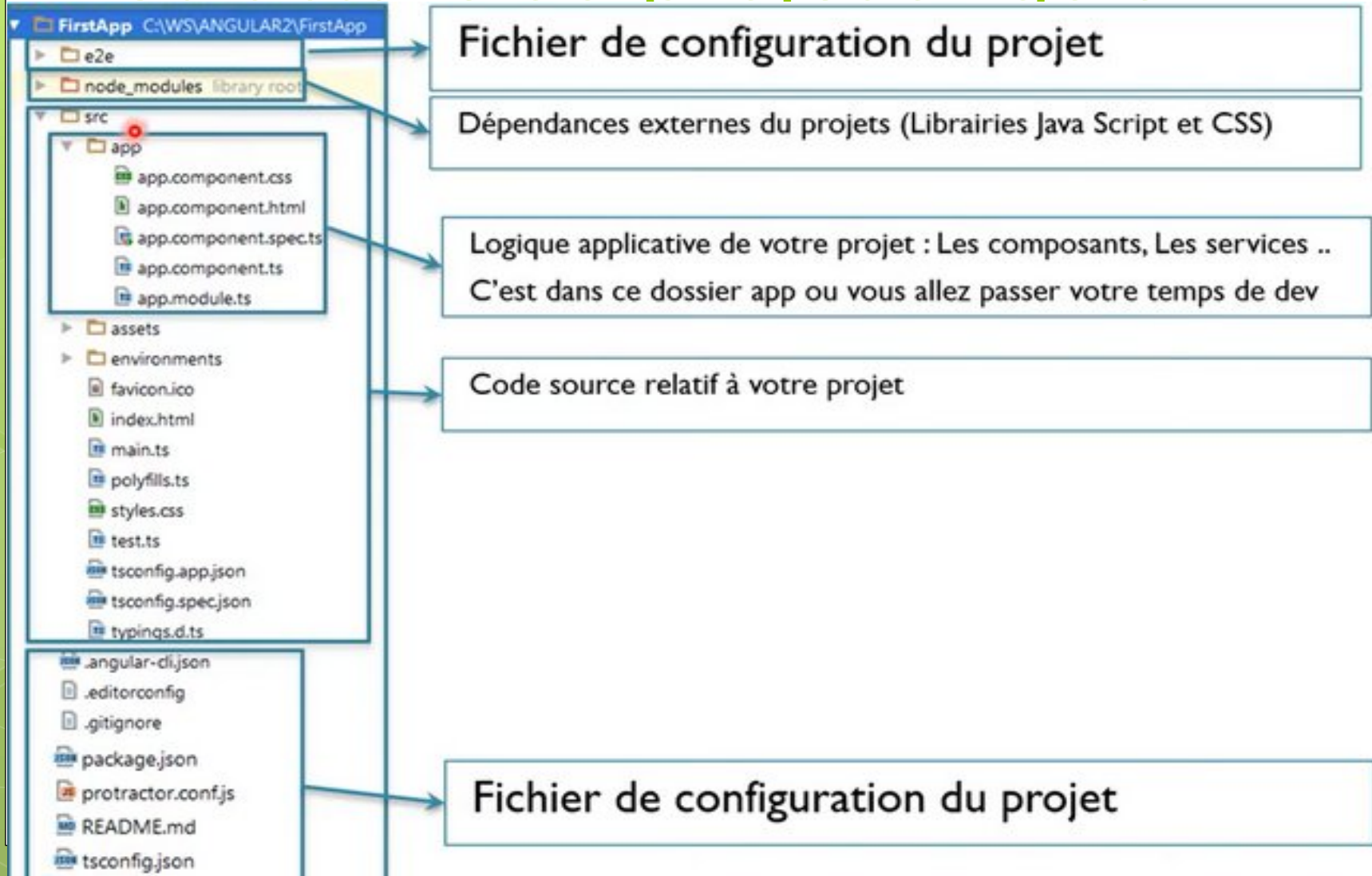
- On génère un projet en se basant sur angular cli via la commande `ng` suivie de `new` et le nom du projet
- `ng new monprojet`
- Cette commande génère les différents fichiers requis par une application basique Angular et installe toutes les dépendances requises par ce projet

Exécuter un projet angular

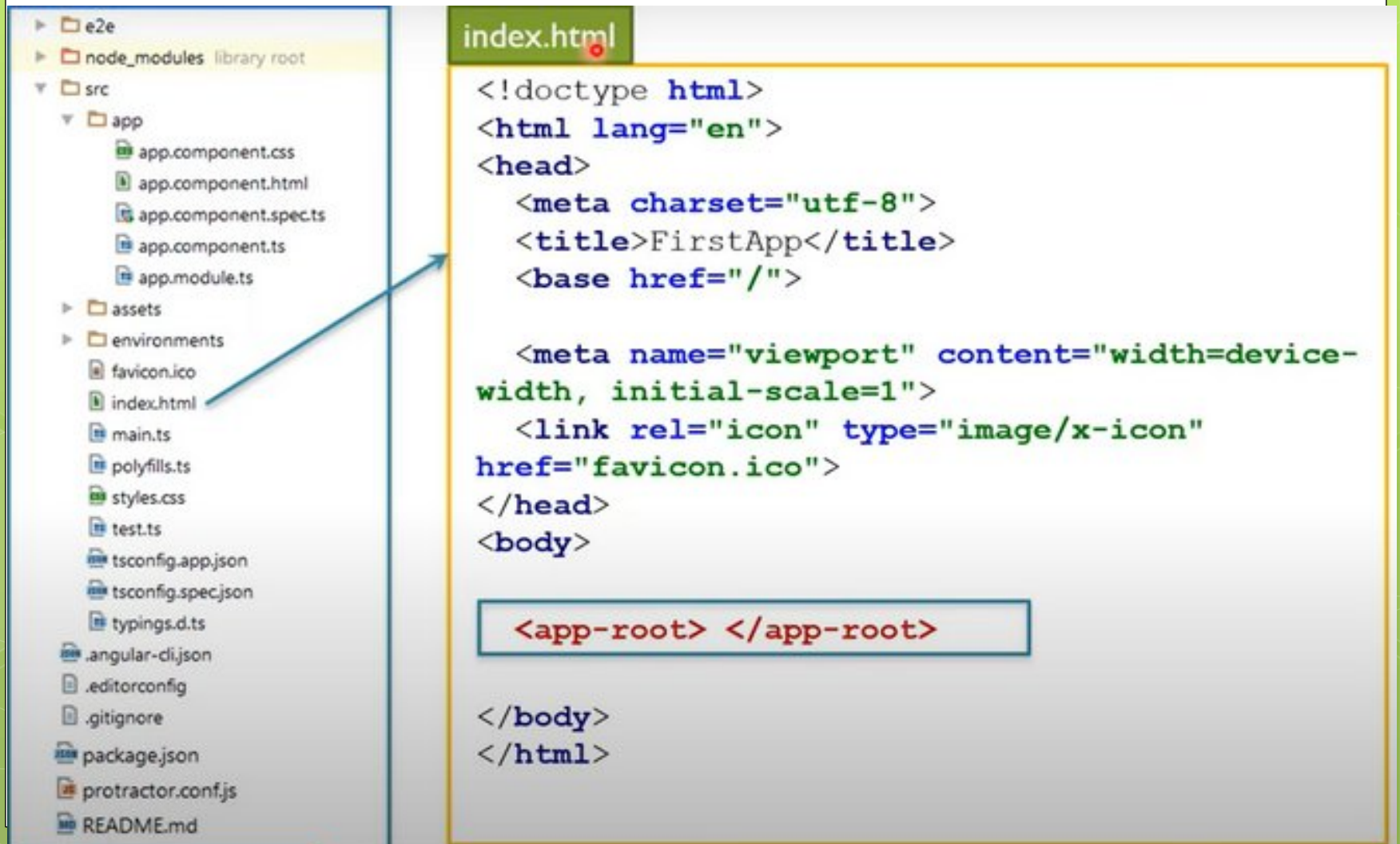
- Pour tester et exécuter le projet angular, on utilise la commande :
- `ng serve`
- Compiler le code source du projet et transpiler le code typescript en code javascript et démarre un serveur web local



Structure du projet angular



Structure du projet



The image shows a project structure on the left and the content of the `index.html` file on the right. A blue arrow points from the `index.html` file in the file explorer to the corresponding code block on the right.

Project Structure (Left):

- ▶ e2e
- ▶ node_modules library root
- ▼ src
 - ▼ app
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - ▶ assets
 - ▶ environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - tsconfig.app.json
 - tsconfig.spec.json
 - typings.d.ts
 - .angular-cli.json
 - .editorconfig
 - .gitignore
 - package.json
 - protractor.conf.js
 - README.md

index.html (Right):

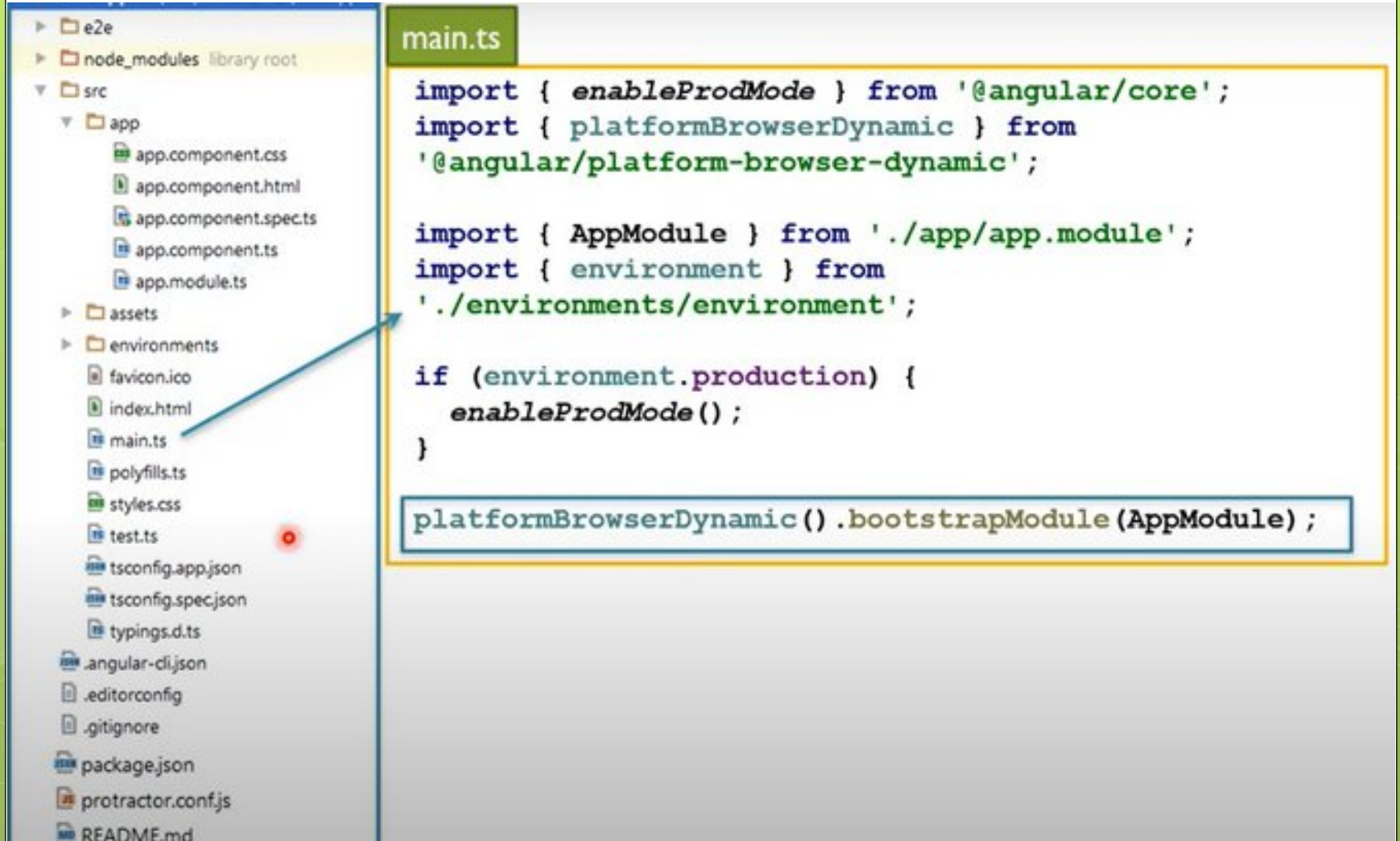
```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>FirstApp</title>
  <base href="/">

  <meta name="viewport" content="width=device-
width, initial-scale=1">
  <link rel="icon" type="image/x-icon"
href="favicon.ico">
</head>
<body>

  <app-root> </app-root>

</body>
</html>
```

Structure du projet



The image displays the structure of an Angular project. On the left, a file explorer shows the project hierarchy. The 'src' directory contains the 'app' directory, which includes 'app.component.css', 'app.component.html', 'app.component.spec.ts', 'app.component.ts', and 'app.module.ts'. Other files in 'src' include 'assets', 'environments', 'favicon.ico', 'index.html', 'main.ts', 'polyfills.ts', 'styles.css', 'test.ts', 'tsconfig.app.json', 'tsconfig.spec.json', 'typings.d.ts', '.angular-cli.json', '.editorconfig', '.gitignore', 'package.json', 'protractor.conf.js', and 'README.md'. An arrow points from 'main.ts' in the file explorer to the code editor on the right.

The code editor shows the content of 'main.ts':

```
main.ts

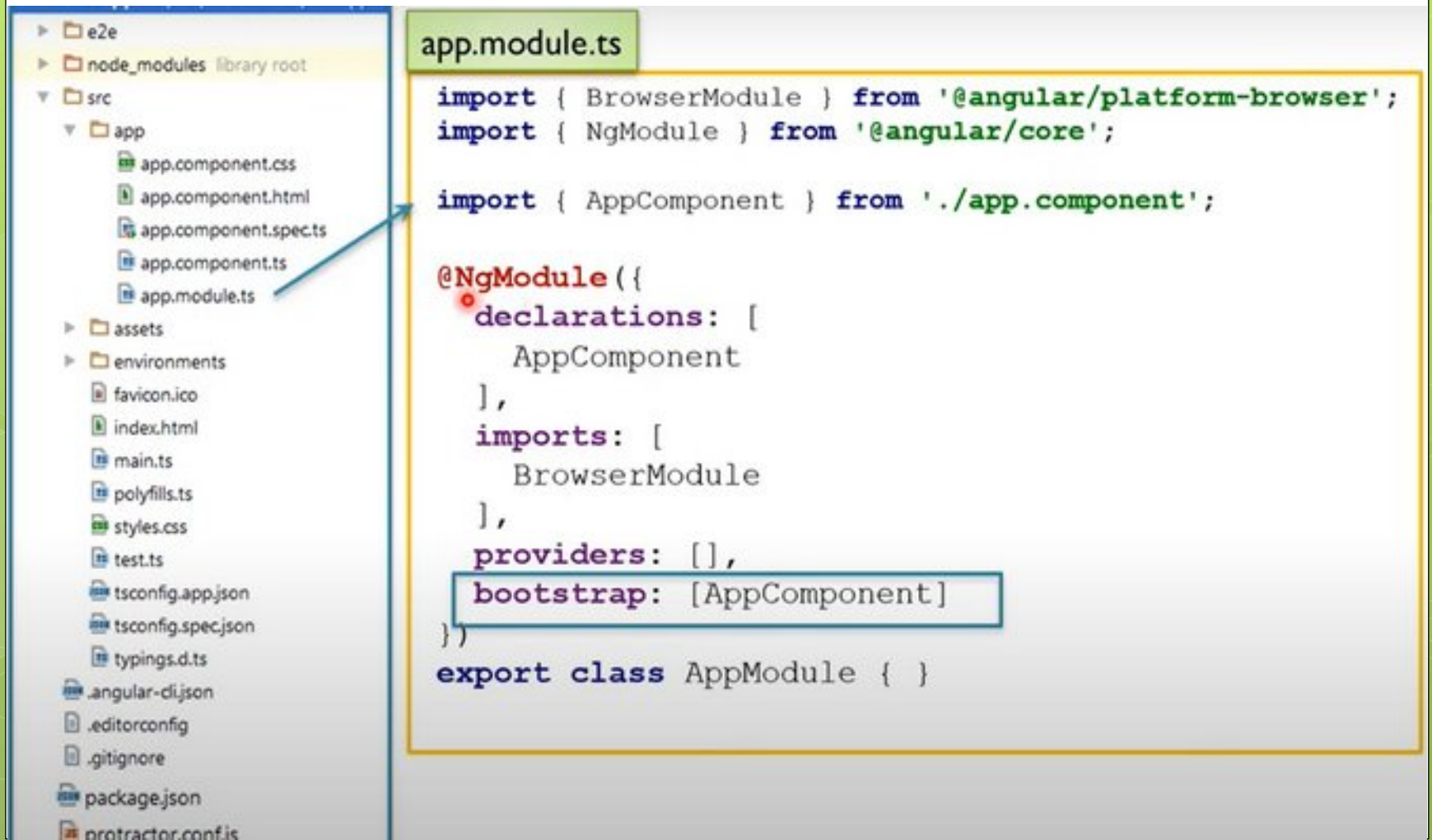
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from
  '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from
  './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

Structure



The image displays the structure of an Angular application, showing a file explorer on the left and the content of `app.module.ts` on the right.

File Explorer (Left):

- e2e
- node_modules library root
- src
 - app
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - tsconfig.app.json
 - tsconfig.spec.json
 - typings.d.ts
 - .angular-cli.json
 - .editorconfig
 - .gitignore
 - package.json
 - protractor.conf.js

Code Editor (Right):

The code editor shows the content of `app.module.ts`:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

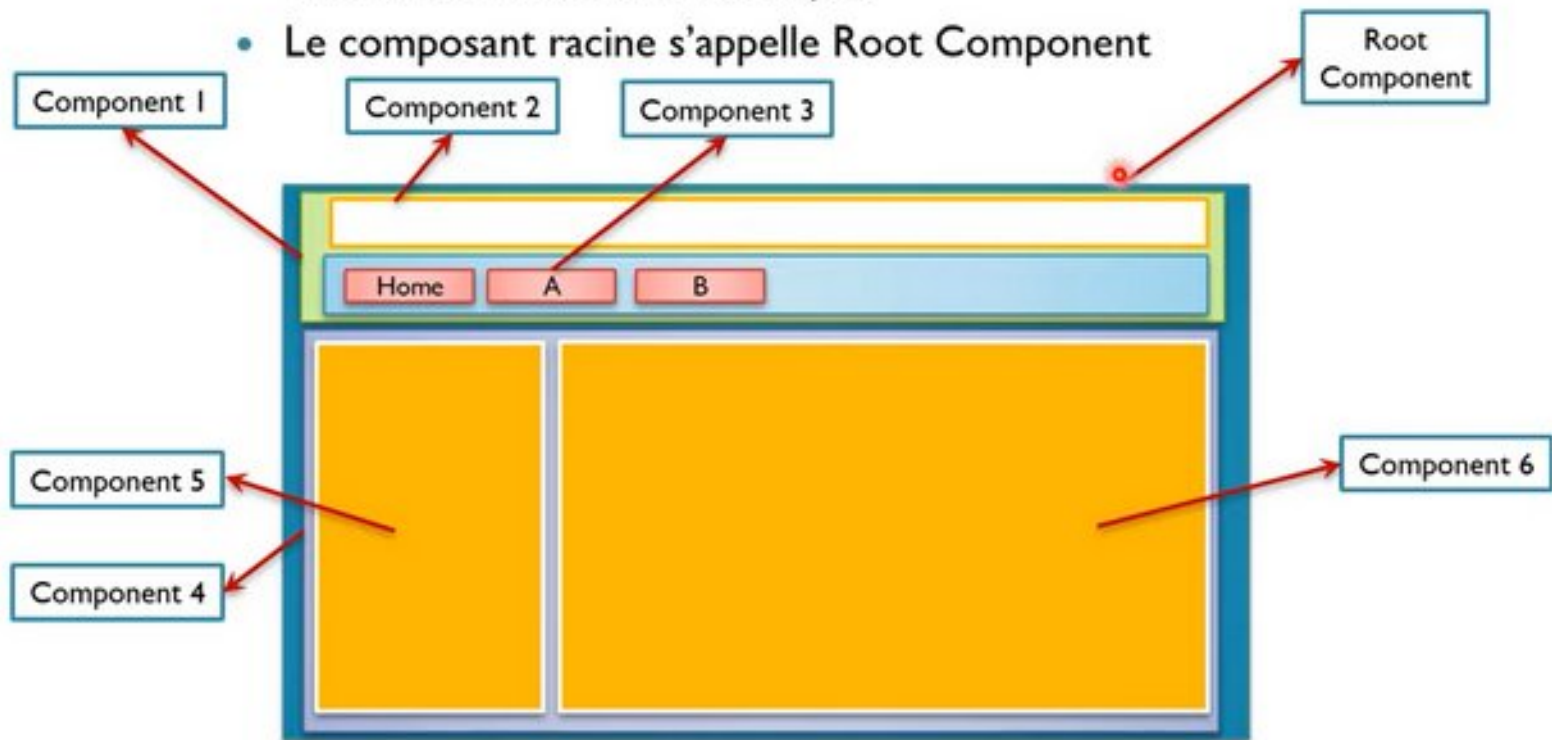
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

The `bootstrap` property in the `@NgModule` decorator is highlighted with a red box.

Les webcomponents

- Les composants sont des éléments importants dans Angular.
- L'application est formée par un ensemble de composants.
- Chaque composant peut imbriquer d'autres composants définissant ainsi une structure hiérarchique.
- Le composant racine s'appelle Root Component



Les composants

Ng generate component Nomcomposant

The diagram illustrates the relationship between the source files of an Angular component and its runtime output. On the left, a file explorer shows the project structure with the following files in the `src/app` directory:

- `app.component.css`
- `app.component.html`
- `app.component.spec.ts`
- `app.component.ts`
- `app.module.ts`

Arrows point from these files to their respective code blocks on the right:

- app.component.ts**: Contains the component metadata and the `AppComponent` class.

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
}
```
- app.component.html**: Contains the component's template.

```
<div style="text-align:center">
  <h1>
    Welcome to {{title}}!!
  </h1>
</div>
```
- app.component.css**: Contains the component's styles.

A screenshot of a web browser shows the rendered output: "Welcome to app!!".

Les fichiers spec sont des tests unitaires pour vos fichiers source. La convention pour les applications Angular2 est d'avoir un fichier `.spec.ts` pour chaque fichier `.ts`. Ils sont exécutés à l'aide du framework de test javascript Jasmine via le programme de tâches Karma lorsque vous utilisez la commande `'ng test'`.

commandes

- `npm i -g @angular/cli`
- `Ng v`
- `Ng new [nom-projet]`
- `Ng serve`
- `Set-ExecutionPolicy RemoteSigned`