

Angular CLI will by default generate a separate file containing this bootstrap logic: `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)
  .catch(err => console.error(err));
```

Yay! But wait a second. We need an HTML file to serve to our users, right?

The CLI created an `index.html` file for us, which is the single page of our application. You might wonder how it could possibly work, since it doesn't contain any `script` element.

When you run `ng serve`, the CLI calls the TypeScript compiler. The compiler outputs JavaScript files. The CLI then bundles them and adds the necessary `script` elements to the `index.html` file (using [Webpack](#) behind the scenes).

Hopefully, you now have a better understanding of the various parts of this first Angular application. It's not really a dynamic app yet, and we could have done the same in one second in a static HTML page, I'll give you that. So let's jump to the next sections, and learn all about dependency injection and templating.