# Reusable Modular components

## What are Angular Modules?

While discussing modules in this Article, we will be referring to Angular Modules. I mean the TypeScript file with the @ngModule() decorator. There are other kinds of module, like the JavaScript ones you refer when you import a file, but we will not be concerned by them today.

Modules are ways to bundling Angular building blocks together. You can include in them all sorts of files, including components, directives, services, pipes, or even other modules. We have them grouped all together so that Angular is aware of these features. Because Angular doesn’t automatically scan all of the files in your project. It doesn’t search all the code you write and automatically detect all components and services. Instead, you need to tell angular what is a component and which component to load. And so one for every type of file you want your app to use. Every Angular App needs to have a least one module, but It can also be splited in smaller modules. This would allow you to delay the loading of some of your file to only when you want them loaded, increasing the performance of your application.

You can’t use feature of building blocks if they are not included in a module. How you include it, if you add it to provider or declaration, depend on the type of feature you are importing.

Lets look at the template of an application module to understand better where to import file:

import { NgModule } from '@angular/core';

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

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

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Components, pipes and custom directives will be imported under the **declarations array**, the other modules, like the routing modules, angular feature modules or third-party library modules will be added in the **imports array** while services will be added in the **providers array**.

The boostrap array is used to declare which component is used at the start of the application.

### The (not-so) special routing Module

The routing module is working exactly like a regular module. It is simply there to hold our routing configuration. We could add this into our app module. We usually outsource it because that is quite a lot of code and it keeps our app module leaner, more focused and easy to maintain.