JavaScript modules vs. NgModules

JavaScript and Angular use modules to organize code, and though they organize it differently, Angular apps rely on both.

JavaScript modules

In JavaScript, modules are individual files with JavaScript code in them. To make what's in them available, you write an export statement, usually after the relevant code, like this:

```
export class AppComponent { ... }
```

Then, when you need that file's code in another file, you import it like this:

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

JavaScript modules help you namespace, preventing accidental global variables.

For more information on JavaScript modules, see JavaScript/ECMAScript modules.

NgModules

NgModules are classes decorated with `@NgModule`. The `@NgModule` decorator's `imports` array tells Angular what other NgModules the current module needs. The modules in the `imports` array are different than JavaScript modules because they are NgModules rather than regular JavaScript modules. Classes with an `@NgModule` decorator are by convention kept in their own files, but what makes them an `NgModule` isn't being in their own file, like JavaScript modules; it's the presence of `@NgModule` and its metadata.

The AppModule generated from the Angular CLI demonstrates both kinds of modules in action:

```
/* These are JavaScript import
statements. Angular doesn't know
anything about these. */
import { BrowserModule } from
'@angular/platform-browser';
import { NgModule } from
'@angular/core';

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

/* The @NgModule decorator lets Angular
know that this is an NgModule. */
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [     /* These are NgModule
imports. */
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
```

```
  })
  export class AppModule { }
```

The NgModule classes differ from JavaScript module in the following key ways:

- An NgModule bounds declarable classes only. Declarables are the only classes that matter to the Angular compiler.

- Instead of defining all member classes in one giant file as in a JavaScript module, you list the module's classes in the `@NgModule.declarations` list.

- An NgModule can only export the declarable classes it owns or imports from other modules. It doesn't declare or export any other kind of class.

- Unlike JavaScript modules, an NgModule can extend the *entire* application with services by adding providers to the `@NgModule.providers` list.

More on NgModules

For more information on NgModules, see:

- [Bootstrapping](#).

- [Frequently used modules](#).

- [Providers](#).