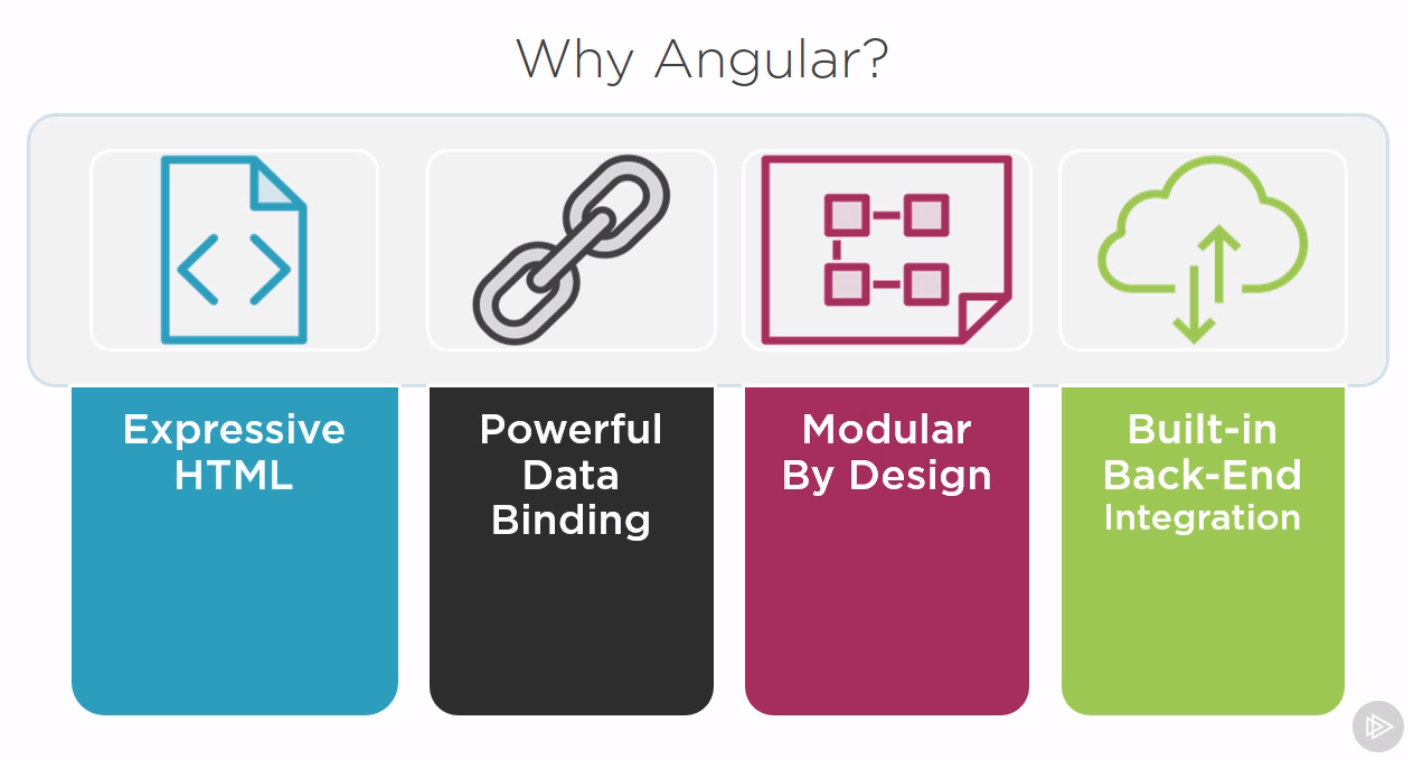
Intro:

* What is Angular?

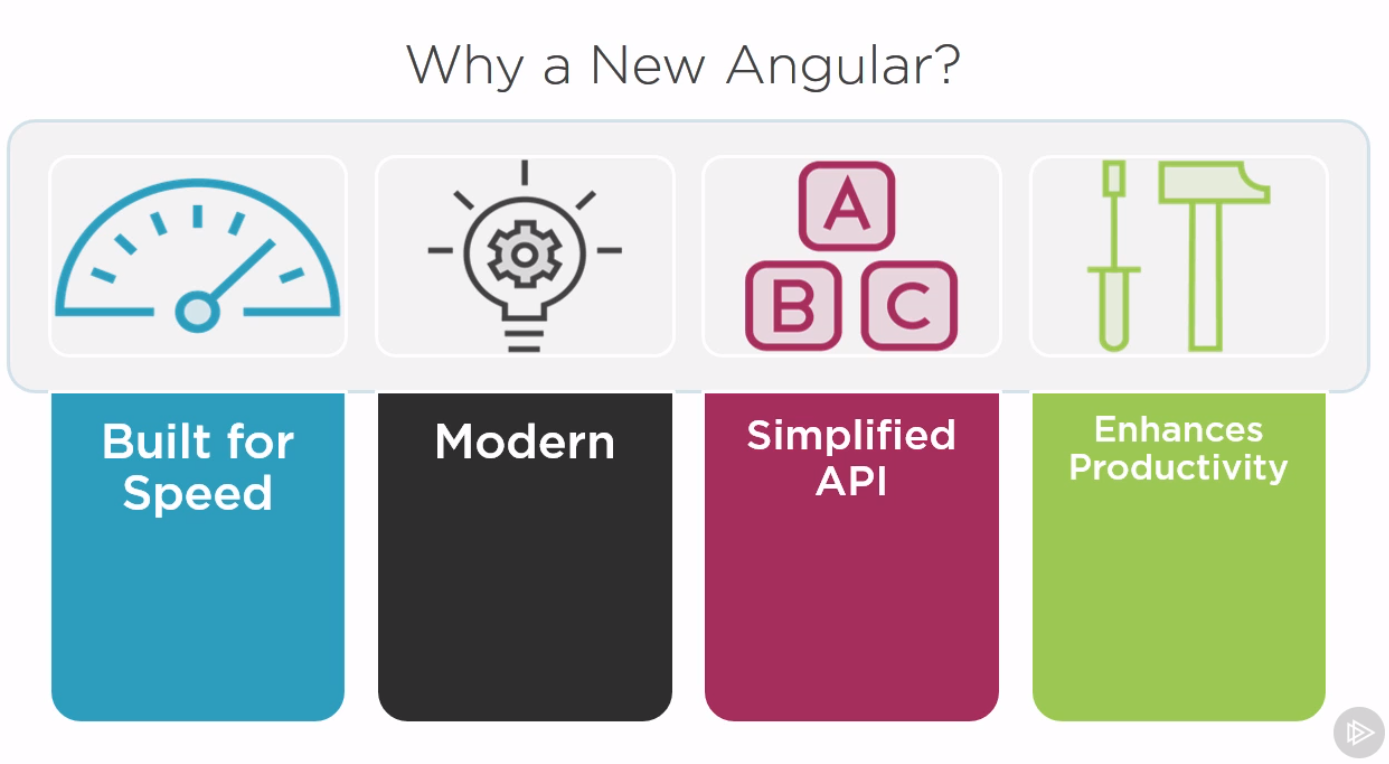
Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.

* Why Angular?

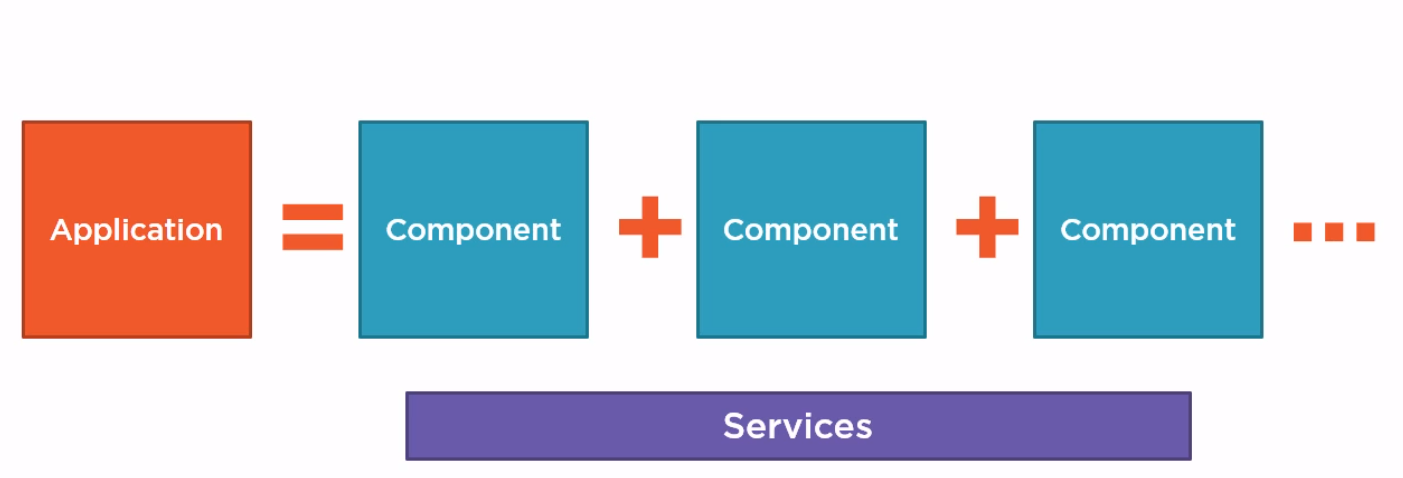
There are many front-end JavaScript frameworks to choose from today, each with its own set of trade-offs. Many people were happy with the functionality that Angular 1.x afforded them. Angular 2 improved on that functionality and made it faster, more scalable and more modern. Organizations that found value in Angular 1.x will find more value in Angular 2.

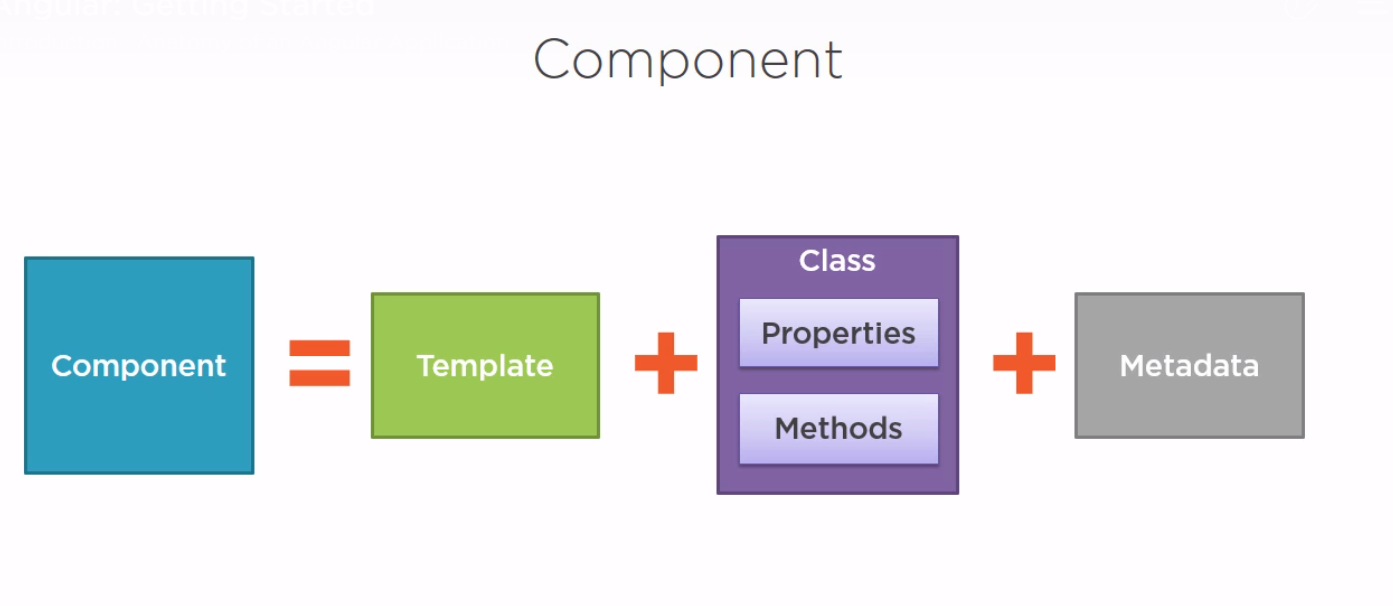


* Why a new Angular?



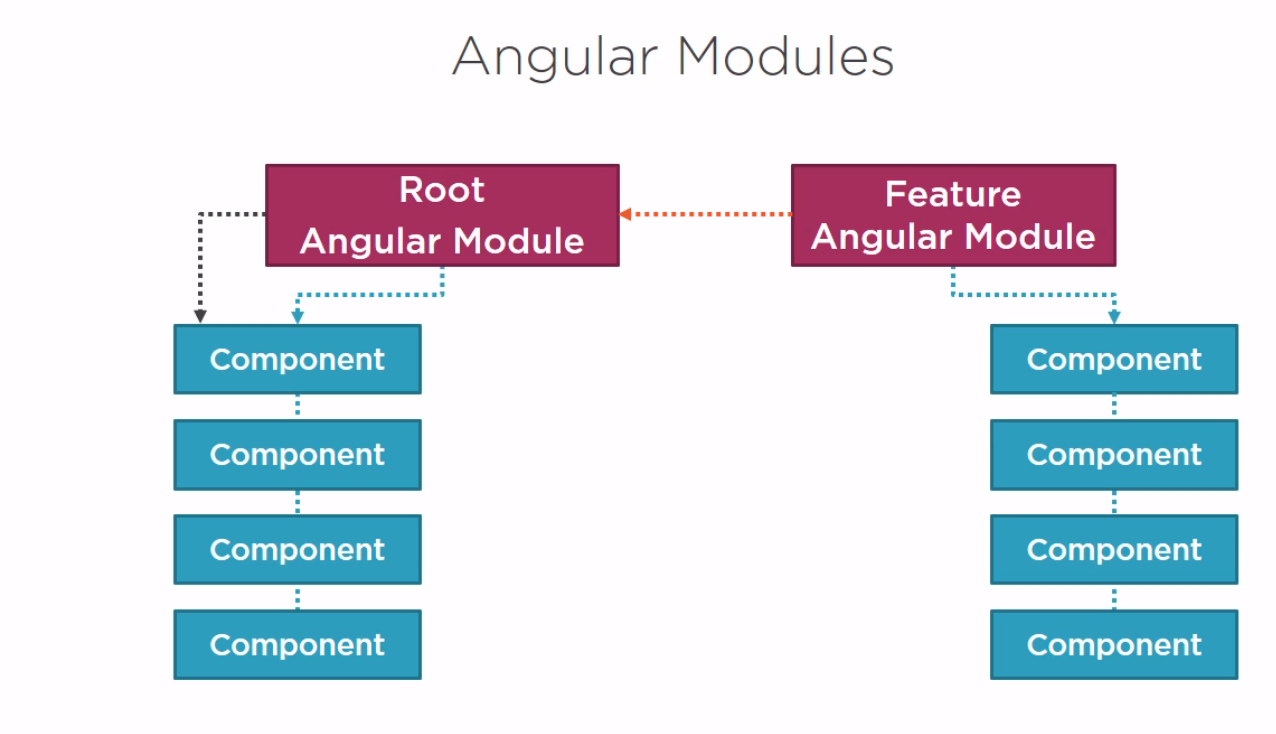
* Application anatomy:





* How do we pull components together into an application?

We define Angular modules. Angular modules help us organize our application into cohesive blocks of functionality. Every Angular application has at least one Angular module called the **application's root Angular module**. An application can have any number of additional Angular modules including feature modules that consolidate the components for a specific application feature.



* What language to use?

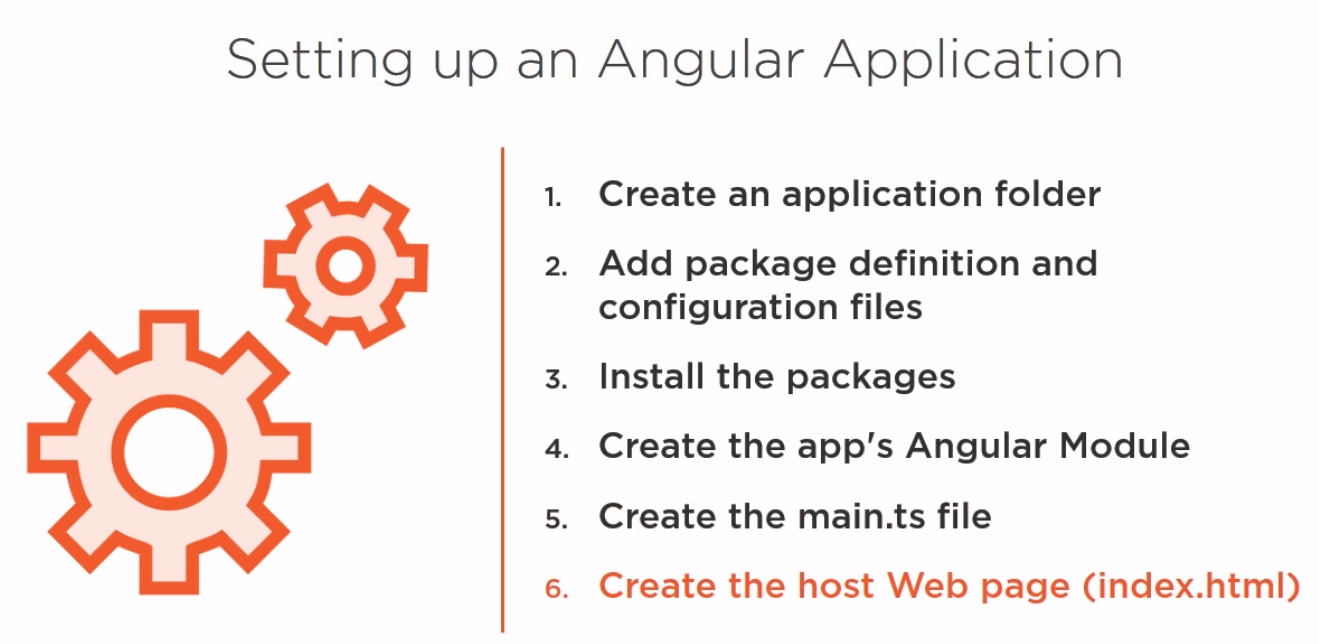
We can use:

1. ES 5 (need no transpiler  as it’s the main language for browsers to understand)
2. ES 6 (Super set of ES5 and has a lot of new features like classes, let, arrow and etc. but need to be transpiler to ES 5 to let browsers fully understand it)
3. **Typescript** (Superset of ES6, has strong typing and great IDE tooling, it’s strongly typed and it also can detect JS modules as strongly typed through \*.d.ts files and it’s OOP, Also as we said it’s superset of JS so we need to transpile it to ES5 in order to browser to understand).
4. Dart.

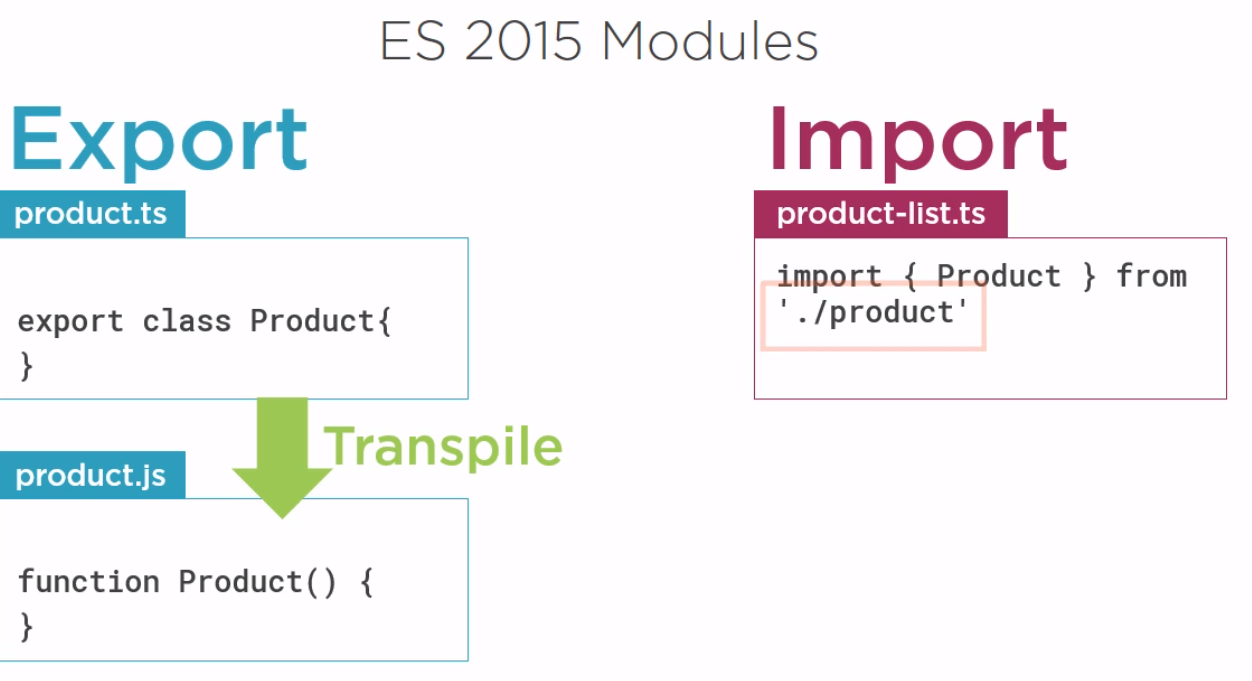
* We will use VS code as our editor.
* Setting up env:

We need only npm and it’s required because using npm, we can install libraries, packages, and applications along with their dependencies. We'll need npm to install all of the libraries for Angular and to execute scripts to transpile our code and launch our Angular application.

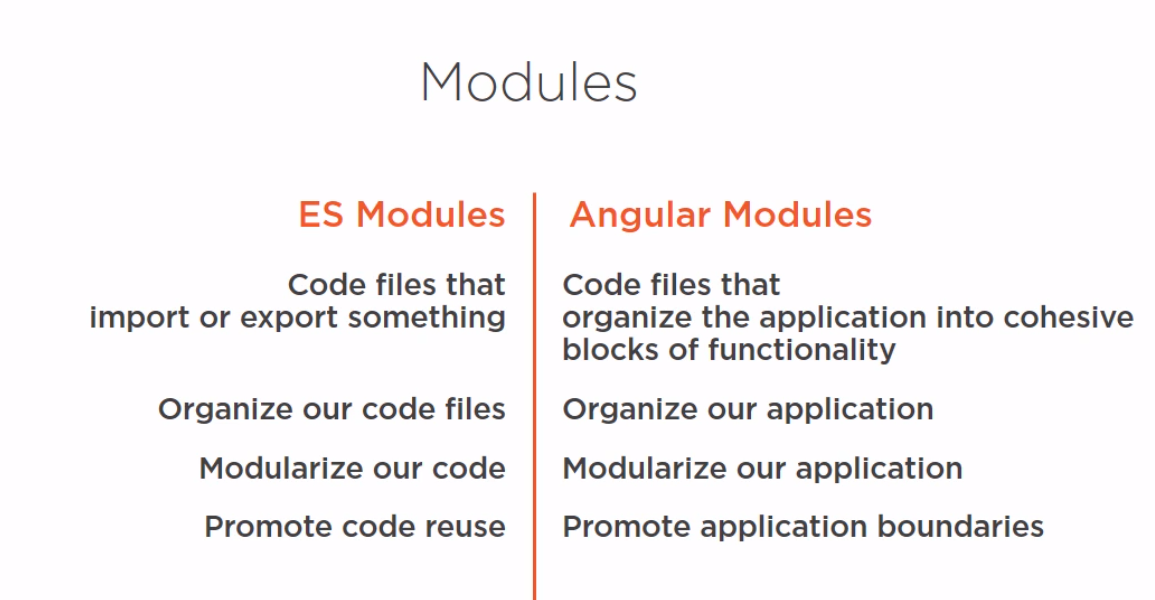
* Setting up our angular app:
  + To manually set up an Angular application we'd need to:



* + - Create an application folder.
    - Add package definition and configuration files.
    - Install the packages.
    - Create the application's root angular module because every Angular application needs at least one Angular module.
    - Then create the main.ts file to load that Angular module.
    - And create the host web page, normally index.html.
  + But you don’t want to do all of this manually, but instead you can:
    - Use AngularCLI (The best to go)
    - Use Formal Angular Startup projects
    - Or clone Debra Corata startup project: <https://github.com/DeborahK/Angular-GettingStarted>
  + For Debra Corata startup project:
    - Clone it.
    - Npm install
    - Npm start (to run the start script which let webpack run the server)
* Angular Modules:
  + There are 2 main different things. ES6 modules which are a modules to organize code and allow us easily reuse our code as shown below:



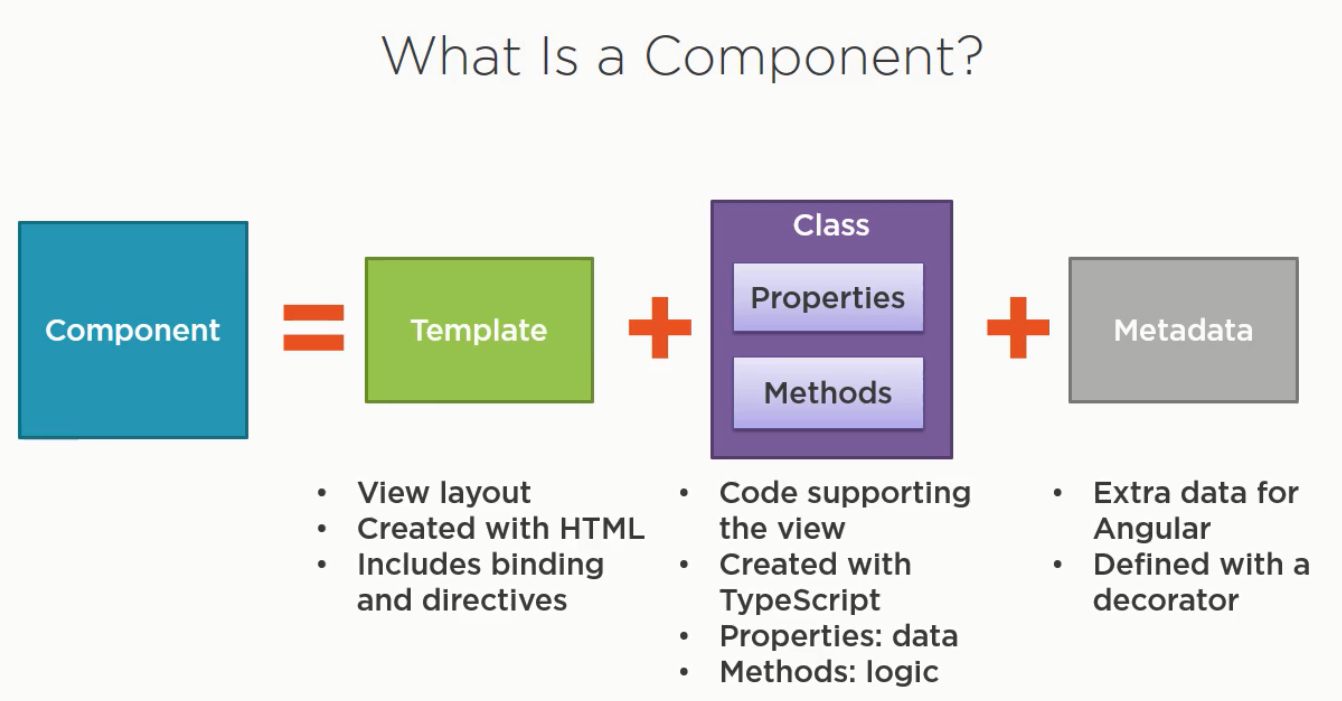
As we see in ES6 each js file is a module, and we can use this also in typescript so each file is a module and we can export it using export keyword to make it available to be imported in other modules (files) and as a note we import the js file not the ts file. But the angular module is another thing as we use it as organizer to many modules.



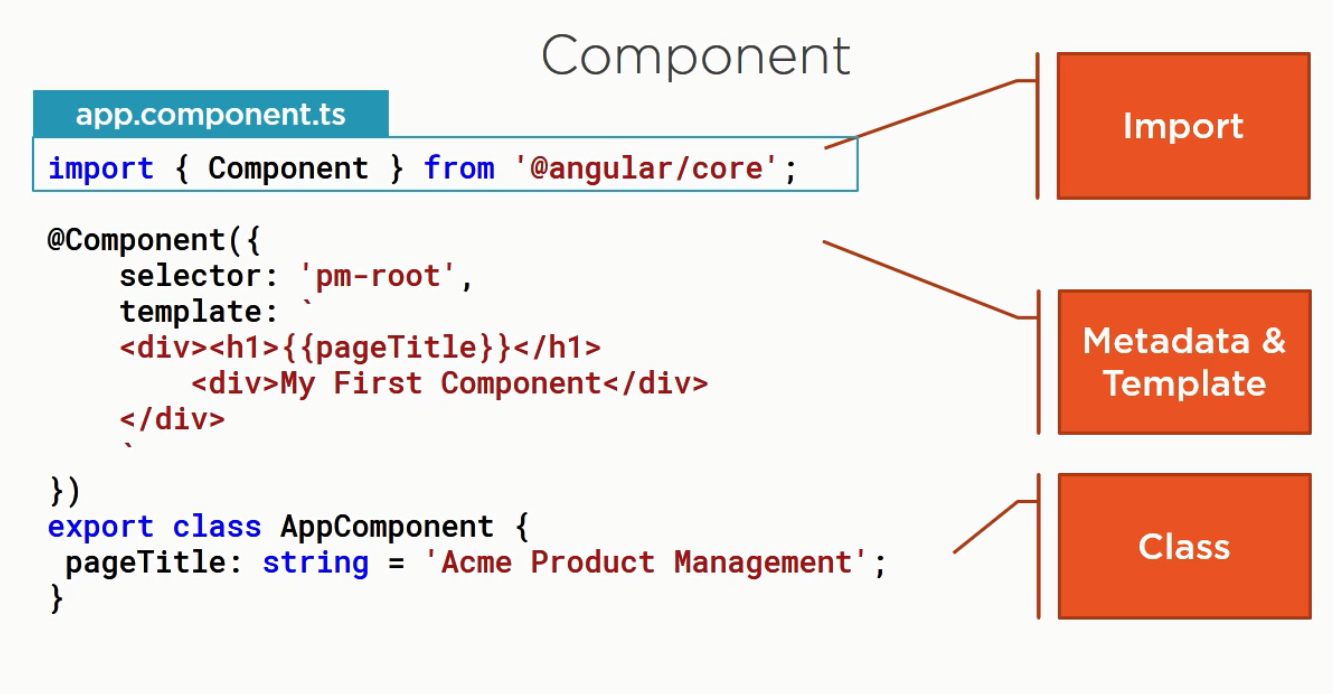
Intro to components:

* What is components:

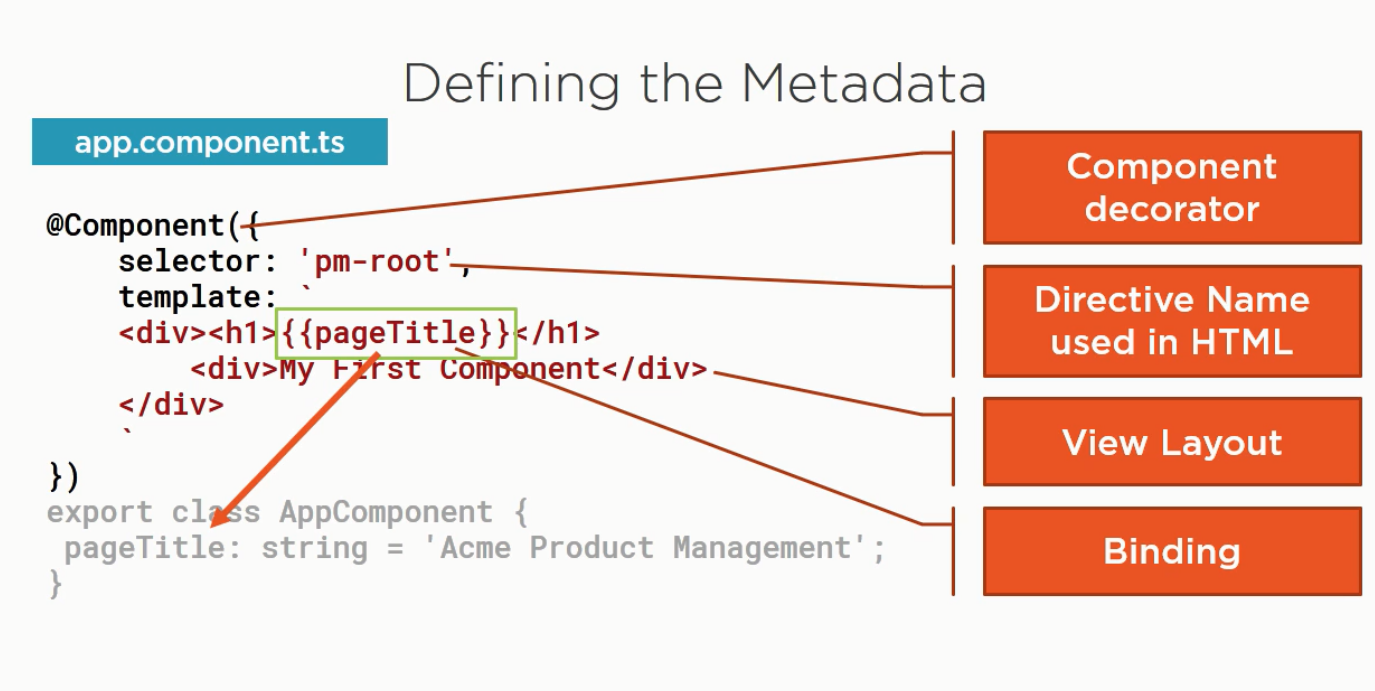
Component Anatomy:



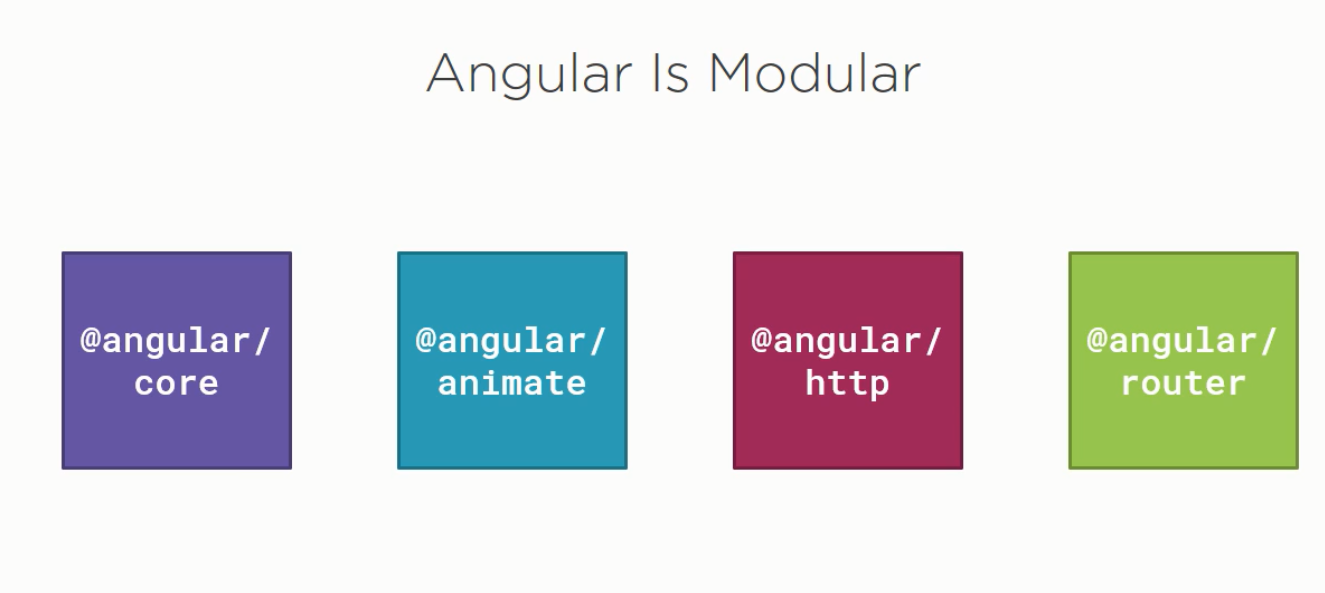
Component Example:



* Important notes about component:
  + Root component always called AppComponent.
  + A class become an angular component when we decorate it with component metadata.
  + Decorator is a function that adds metadata to a class, its members, or its method arguments.
  + Decorators start by @ and it’s a JS feature that Typescript implement.
  + Angular has a lot of built-in decorators and @Component decorator comes from @angular/core module.
  + We can build our own custom decorators.
  + @Component decorator takes one object has a lot of parameters like the selector and template or templateUrl and a lot more.



* + We can import from 3rd party libraries, our own ES modules (That transpiled from our typescript class for example: components) or from angular.
  + We can import from angular because angular is modular:



* + Angular CLI creates the AppComponent for us:

