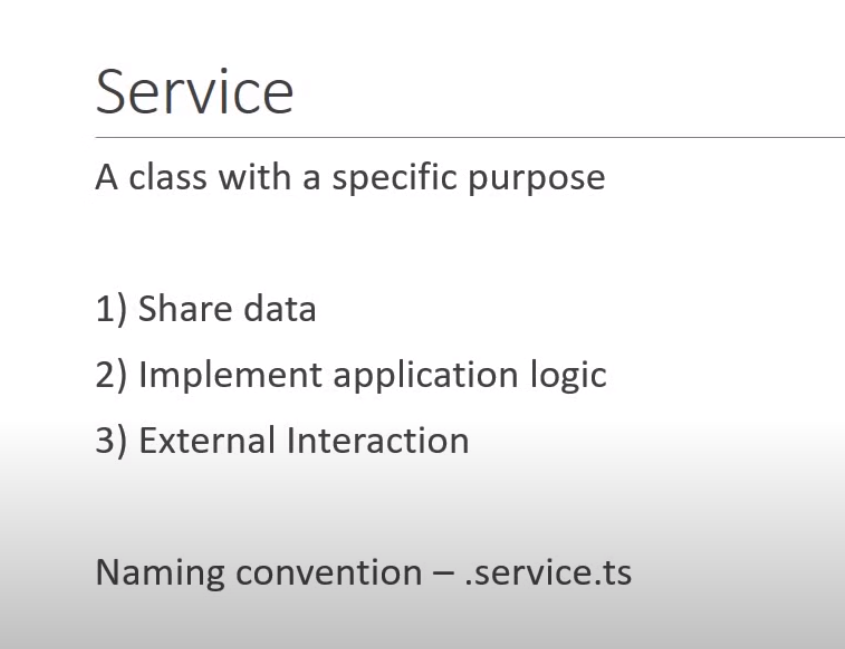
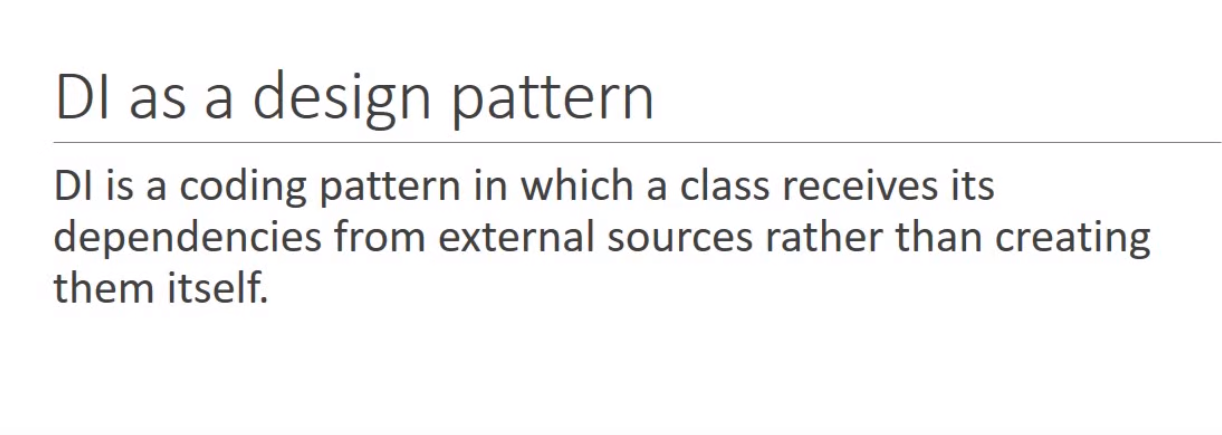
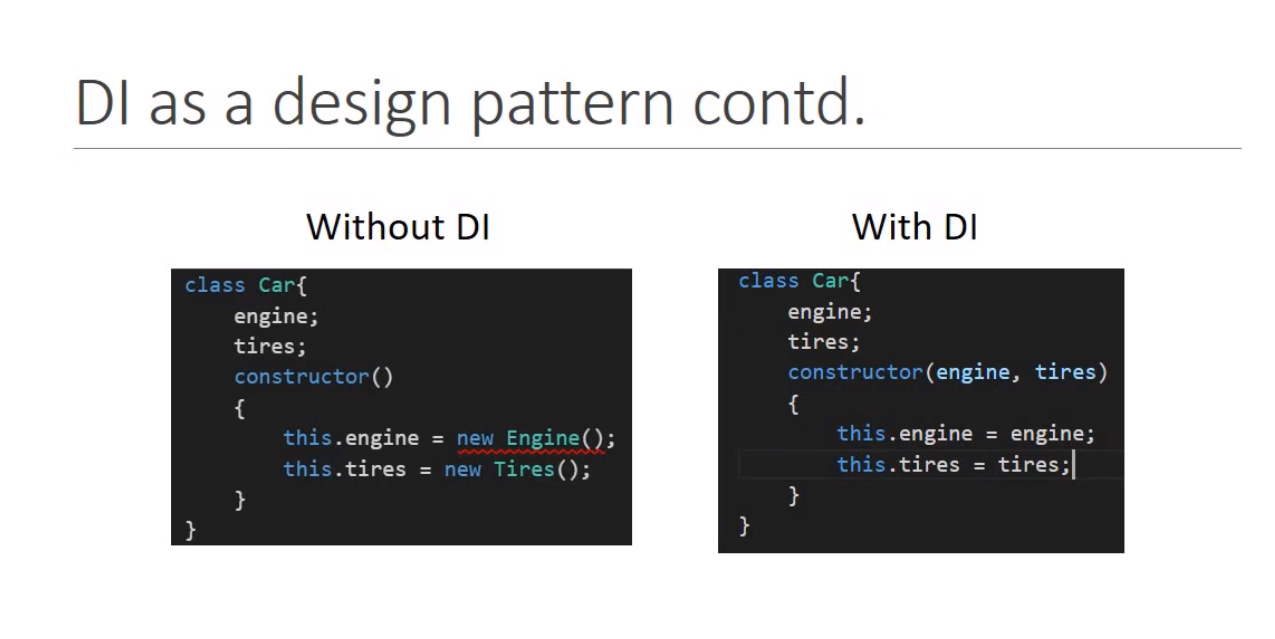
# Service



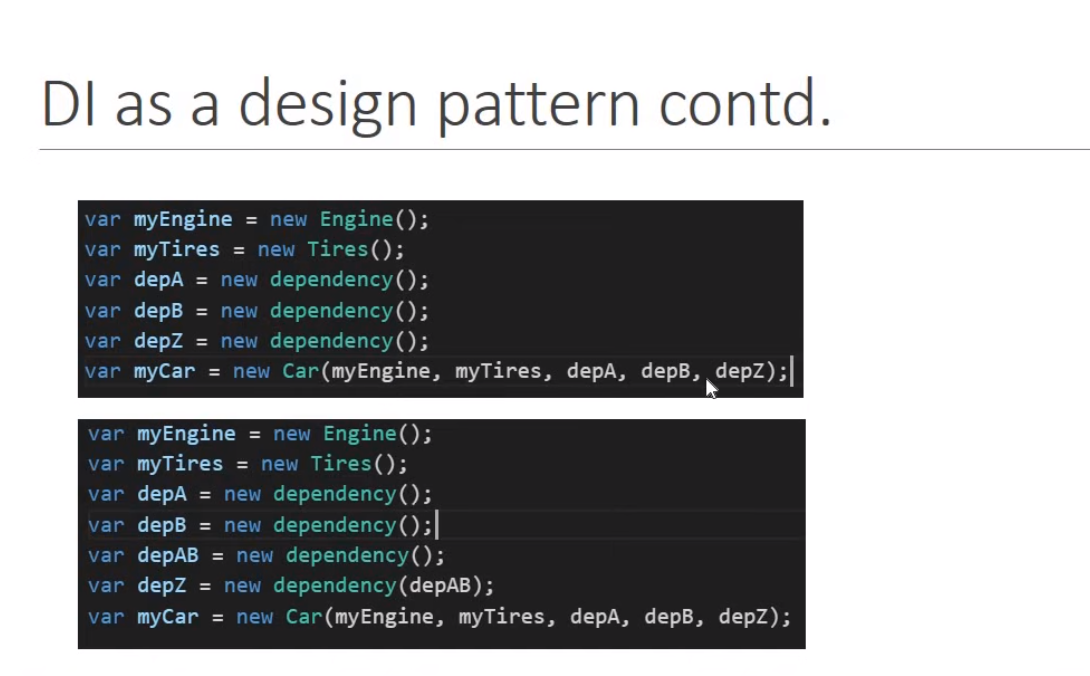
# Dependency Injection





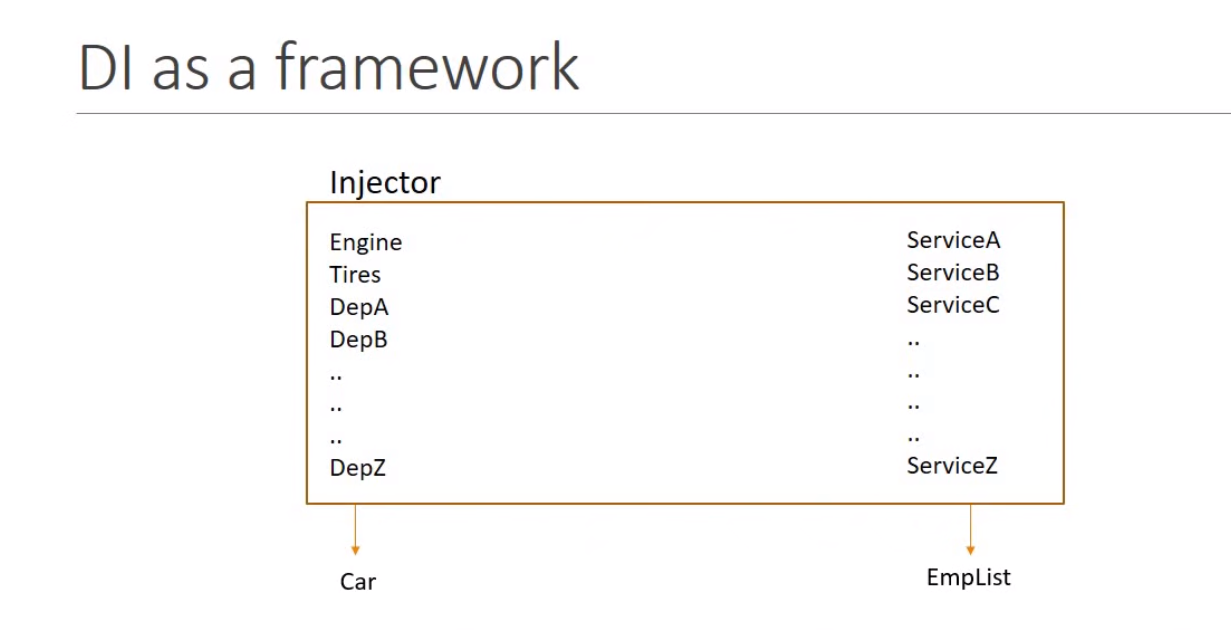
We as a developer have to create those basic dependencies first of all required and then proceed with constructing the other dependencies using them.

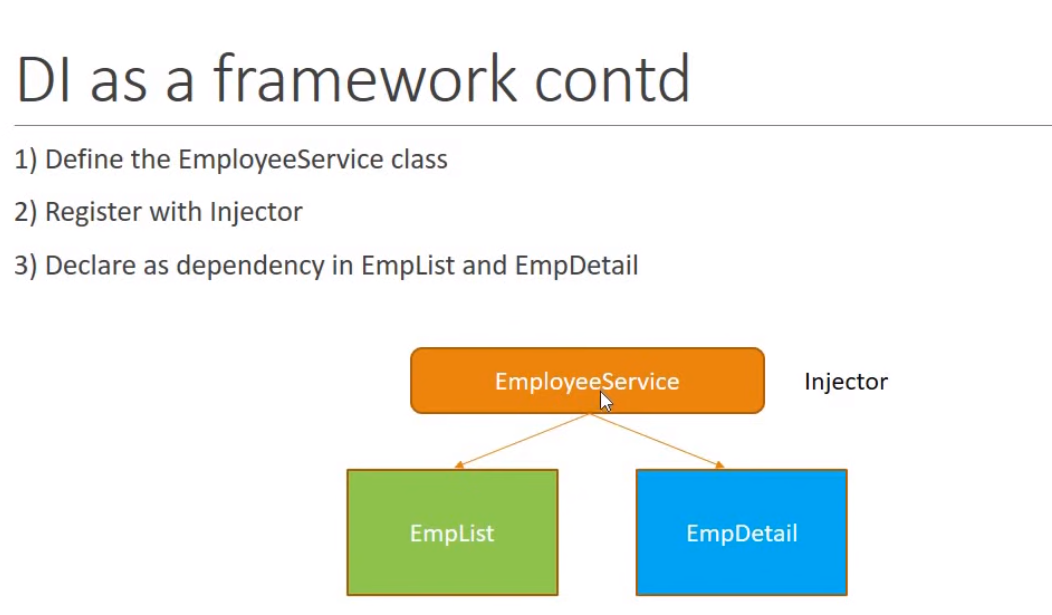
This is very difficult to manage the code like that.



DI framework in angular has something called Injector where you register all your dependencies.

Injector is basically like a container of all dependencies , this framework will manage all the dependencies so that you don’t have to keep track of it.





# Registering and using the service

To register a service in a particular component and its children only, you can use the providers array in the component decorator. Here are the steps to follow:

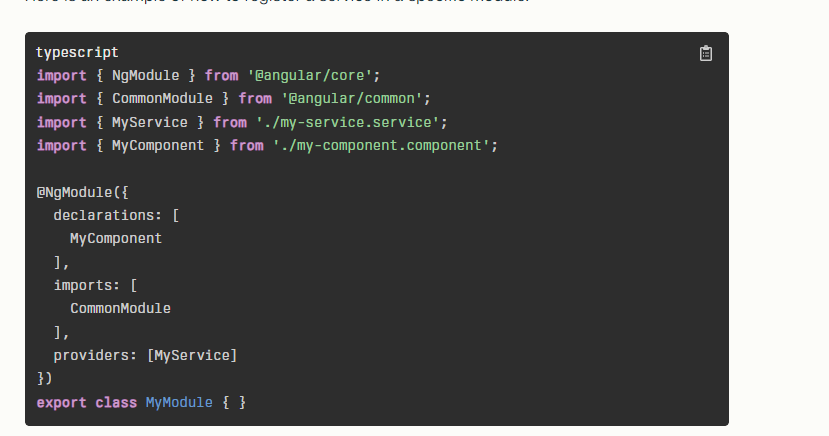
* Import the service that you want to register in the component.
* In the component decorator, add the service to the providers array.
* Use the service in the component and its children.



Registering a service in a component and its children only is not a common use case. Services are usually registered in the root module or a shared module to make them available to the entire application.

To register a service only in a specific module in Angular, you can use the providers array in the module decorator. Here are the steps to follow:

* Import the service that you want to register in the module.
* In the module decorator, add the service to the providers array.
* Use the service in the components that are declared in the module.



The providedIn: 'root' property is used in Angular to specify how a service should be provided. It is a metadata property that is added to the @Injectable decorator of a service. When providedIn: 'root' is used, Angular creates a single instance of the service that is shared across the entire application. This means that the service is available to all components and modules in the application.

Here are some use cases for providedIn: 'root':

* When you want to create a singleton service that is shared across the entire application.
* When you want to make the service tree-shakable, meaning that it can be removed from the final bundle if it is not used.
* When you want to simplify the code by not having to add the service to the providers array of every module that needs it.

Overall, providedIn: 'root' is a convenient way to provide a service in an Angular application, especially when the service needs to be shared across multiple components and modules.

