

Laboratorios

Ejercicio 01

- Crear un componente llamado Auto
- Crear una clase Motor con un atributo number cilindros cuyo valor sea de 4.
- Crear una clase Ruedas, con dos atributos string : make cuyo valor va a ser Flinstone, y modelo, con el valor de Square.
- Crear una clase Auto con el decorador Injectable()
- En el constructor de la clase Auto inyectar una instancia de las clases Motor y Ruedas.
- Dentro de Auto, definir un método llamado Drive que retorne un string con la descripción del auto.
- En el componente Auto hacemos lo siguiente:
 - Inyectamos una instancia de la dependencia Auto a través del constructor.
 - Inyectamos una instancia de la dependencia Auto sin DI (a través del new).
 - Creamos un Factory para que al llamarlo genere una nueva instancia de la dependencia Auto.
 - Inyectamos el servicio auto a través del injector.get.

Ejercicio 02

- Crear un servicio llamado LoggerService.
- Dentro de LoggerService hacemos lo siguiente:
 - definimos la propiedad "provideIn" del decorador @Injectable() con el valor "root".
 - creamos la propiedad logs, del tipo array de string, con un valor inicial vacío.
 - Creamos un método llamado log, que reciba como parámetro un string. En el agregamos a la propiedad logs el valor ingresado como parámetro.
- Crear un componente ProviderUno:
 - inyectar el servicio LoggerService a través del constructor.
- Crear un componente ProviderDos:
 - inyectar el LoggerService a través de la propiedad providers del decorador @Component().
 - inyectar LoggerService a través de la propiedad providers del decorador @Component() usando la propiedad useValue con un valor null. En el constructor agregar el decorador @Optional() para que no haya problemas con la dependencia.