


✓ 100 XP 

Exercise - Declare an interface to ensure class shape

5 minutes

Recall that in Typescript, you can use an interface to establish a "code contract" that describe the required properties of an object and their types. So, you can use an interface to ensure class instance shape. Class declarations may reference one or more interfaces in their `implements` clause to validate that they provide an implementation of the interfaces.

Continue working in the Playground.

1. Declare a `Vehicle` interface that describes the properties and methods of the `car` class.

TypeScript

```
interface Vehicle {  
  make: string;  
  color: string;  
  doors: number;  
  accelerate(speed: number): string;  
  brake(): string;  
  turn(direction: 'left' | 'right'): string;  
}
```

2. Notice that the interface includes the parameters of the constructor, not the properties. Try including one of the private properties (for example, `_make: string`). TypeScript will raise an error, because the interface can only describe the *public-facing* side of the class and may not include private members. This prohibits you from using them to check that a class also has the correct types for the private side of the class instance.
3. You can now implement the `Vehicle` interface in the `car` class. As you build out the details for the class, TypeScript will ensure that the class adheres to the code contract described in the interface.

TypeScript

```
class Car implements Vehicle {
```

```
} // ...
```

Next unit: Design considerations

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆