

[Optional] JavaScript Refresher: Classes, Properties & More

Angular makes heavy use of classes - a feature that's supported by vanilla JavaScript and TypeScript (though TypeScript "extends" it and adds some extra features as you'll see).

A class is essentially a blueprint for objects. Any properties and methods defined in the class will exist on all objects that are created based on the class.

For example, if you had this class (in vanilla JavaScript):

```
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
  
  greet() {  
    console.log('Hi, I am ' + this.name);  
  }  
}
```

You could instantiate it (and create objects) like this:

```
const person1 = new Person('Max', 35);  
const person2 = new Person('Anna', 32);
```

And you could then access the properties and methods defined by the class:

```
console.log(person1.age);  
person2.greet();
```

When using Angular, you'll often define classes which are NEVER instantiated by you!

For example, components are created as classes - i.e., you create blueprints for custom HTML elements. But it's Angular that actually instantiates the classes in the end. You never call `new SomeComponent()` anywhere in your code.

In addition, Angular uses TypeScript - therefore, you often use TS-supported "enhancements" to classes.

For example decorators:

```
@Component({})  
class SomeComponent {}
```

Decorators like `@Component` are used by Angular to add metadata & configuration to classes (and other things, as you'll see throughout the course).

In addition, TypeScript gives you more control over how properties are defined in classes.

You can, for example, mark properties (and methods) as `private`, `public` (the default) and `protected` to control which parts of your code can access which property (or method). You can learn more about these keywords [here](#).

And, in general, you can learn more about TypeScript's support for classes [here](#).

That being said, you don't have to study classes in-depth right now. You'll see most of those important features in action throughout this course.

For the moment, it's just important to understand that this classes feature exists, what it does (= create blueprints for objects) and how to work with classes.