

*//protocols*

*A protocol is a group of related properties and methods that can be implemented by any class. They are more flexible than a normal class interface, since they let you reuse a single API declaration in completely unrelated classes. This makes it possible to represent horizontal relationships on top of an existing class hierarchy.*

*Protocols are a way of enforcing certain methods to be utilized, regardless of the actual class the object is part of, thus ensuring that a certain form of method template is implemented. Basically, a way of declaring an interface to an object while concealing it's class. This is why we write a protocol in a similar way to how we would write an interface declaration.*

*Protocols can include declarations for both instance methods and class methods, as well as properties. protocols, which declare the methods expected to be used for a particular situation.protocol uses the `NSUInteger` value for unsigned integer scalar values.*

*The basic syntax to define a protocol looks like this:*

```
@protocol ProtocolName
```

```
// list of methods and properties
```

```
@end
```

```
//subclasses
```

*In Objective-C a subclass can only be derived from a single direct parent class. This is a concept referred to as single inheritance.A subclass inherits methods and instance variables from its superclass. A subclass can override a method it inherits from the class it is based on.*

*Subclassing is just a subclass of another class and it inherits behavior and members of its parent class.*

*For example, you may have a "car" class which has properties that are common to all cars such color, tires, engine, etc. It also has behaviors that are common to all cars such as drive forward, drive backwards, stop, etc.*

*Now for some reason you need to differentiate between different types of cars but you don't want to repeat all those properties and behaviors in each new class so you write subclasses of the "car" class.*

*So you may have subclasses such as "manual cars", "automatic cars", "convertibles", "electric cars", etc. All of these subclasses have some common traits inherited from the "car" class but will also have some unique characteristics that can be included in the subclass.*