Selector

* Selectors are Objective-C’s internal representation of a method name.
* A selector is the name used to select a method to execute for an object, or the unique identifier that replaces the name when the source code is compiled.
* A selector by itself doesn’t do anything.
* It simply identifies a method.
* The only thing that makes the selector method name different from a plain string is that the compiler makes sure that selectors are unique
* What makes a selector useful is that (in conjunction with the runtime) it acts like a dynamic function pointer that, for a given name, automatically points to the implementation of a method appropriate for whichever class it’s used with
* Suppose you had a selector for the method run, and classes Dog, Athlete, and ComputerSimulation (each of which implemented a method run). The selector could be used with an instance of each of the classes to invoke its run method—even though the implementation might be different for each.

**Note:** You use this technique in special situations, such as when you implement an object that uses the target-action design pattern. Normally, you simply invoke the method directly.

**Getting a Selector**

* There are two ways to get the selector for a method name.
* The **@selector()**directive lets you convert a source-code method name to a selector, and the **NSSelectorFromString()** function lets you convert a string to a selector (the latter is not as efficient).
* Both of these return a special data type for selectors called SEL.
* You can use SEL the exact same way as BOOL, int, or any other data type.

Compiled selectors are of type SEL. There are two common ways to get a selector:

At compile time, you use the compiler directive @selector.

**SEL aSelector = @selector(methodName);**

At runtime, you use the NSSelectorFromString function, where the string is the name of the method:

**SEL aSelector = NSSelectorFromString(@"methodName");**

You use a selector created from a string when you want your code to send a message whose name you may not know until runtime.

**Examples:**

//Using Function NSSelectorFromString

SEL stepOne = NSSelectorFromString(@"startEngine");

//Using directive @selector

SEL stepTwo = @selector(driveForDistance:);

//Using directive @selector

SEL stepThree = @selector(turnByAngle:quickly:);a

References:

* <http://rypress.com/tutorials/objective-c/methods>
* <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Selector.html>