# Objective-C Runtime

* Message Passing
  + Methods are not called in objective C. You dont hump to the method when it is called.
  + When you call a method, the complier will rewrite that code for you in a method call that looks like this:
    - ClassName \*myObject = [[ClassName alloc] init];
    - [myObject doItWithA:a];
    - |||||||
    - VVVVVVV
    - a = objc\_msgSend(myObject, @selector(doItWithA:), allArguments);
  + This message then goes to a lookup table called dispatch table. Dispatch table has all the selectors of methods and their address of the method.
  + This does not slow down the application but lets you handle things like categories because the methods are saved as pointers in the loop up table.

|  |  |
| --- | --- |
| @selector(doIt) | 0x000000 |
| @selector(doItWithA:) | 0x000001 |

* Introspection - It allows you to get, in runtime, the information about classes and methods inside your code.
  + Class - Used for knowing information about a class such as if you want to know what is the superclass.
    - [object class]
    - NSClassFromString - can be used for getting the name of a class in real time.
    - Example:
      * Class myClass = [objectName class];
      * class\_getName(myClass);
      * Class superClass = class\_getSuperClass(myClass);
  + Selector - It is used to get the internal id for a method.
    - @selector
    - Example:
      * SEL selector = @selector(methodName);
  + Method - To get information about a method.
    - Example:
      * Method method = class\_getInstanceMethod([objectName class], @selector(methodName));
      * NSNumber number = method\_getNumberOfArguments(method);