

其实当[receiver message]调用方法时，系统会在运行时偷偷地动态传入两个参数分别是self和_cmd。

之所以称它们为隐藏参数，是因为在源代码中并没有声明和定义这两个参数，self在这里就是指代当前的接收者对象，而_cmd则在这里标示当前的方法选择器，类型是SEL,即当前方法的唯一标识符，通常在使用objc_getAssociatedObject / objc_setAssociatedObject 中唯一标识键名。

当然了，这里有一点是需要注意的，那就是self是在方法实现中访问消息接收者对象的实例变量的途径。当方法中的super关键字接收到消息时，编译器会创建一个objc_super结构体：

```
/// Specifies the superclass of an instance.
struct objc_super {
    /// Specifies an instance of a class.
    __unsafe_unretained id receiver;

    /// Specifies the particular superclass of the instance to message.
#if !defined(__cplusplus) && !__OBJC2__
    /* For compatibility with old objc-runtime.h header */
    __unsafe_unretained Class class;
#else
    __unsafe_unretained Class super_class;
#endif
    /* super_class is the first class to search */
};
```

这个结构体指明了消息应该被传递给特定超类的定义，虽然如此，receiver接收者仍然是self本身。

因为当我们想通过[super class]获取超类时，super只是编译器的标识符，只会将指向self的id指针和class的SEL传递给objc_msgSendSuper函数，因为只有在NSObject类中找到class方法，然后class方法调用object_getClass(),接着调用objc_msgSend(objc_super->receiver,@selector (class)),传入的第一个参数是指向self的id指针，与调用[self class]相同，那么得到的还是self的类型。