

为了我们可以更好的去感受runtime，我们可以在终端使用clang编译器去看看到底在runtime的作用下我们平时写的OC代码变成了什么？

在终端中，使用clang编译器的命令：clang -rewrite-objc main.m
(main.m指的是你要编译的代码文件)

```
int main(int argc, const char * argv[]) {
    /* @autoreleasepool */ { __AtAutoreleasePool __autoreleasepool;

        Person *p1 = ((Person (*)(id, SEL))(void *)objc_msgSend)((id)((Person (*)(id, SEL))
            )(void *)objc_msgSend)((id)objc_getClass("Person"), sel_registerName("alloc")),
            sel_registerName("init"));
        ((void (*)(id, SEL))(void *)objc_msgSend)((id)p1, sel_registerName("sendHello"));

    }
    return 0;
}
```

很显然，我们的OC代码在执行过程中被转成了C的代码，在这里大家可以注意观察一个函数objc_msgsend,没错，这个正是我接下来要讲的runtime的核心——消息传递。

那么何为消息呢？

消息：在维基百科中即：人与人之间沟通的载体、内容。

那么我们这里的消息是啥呢？

其实在这里指的就是：在OC中[self XXX]调用方法内部的真正实现原理，所谓的消息直白点可以理解为对调用某一个方法的内部实现过程的描述。