**Runtime**

RunTime简称运行时,就是系统在运行的时候的一些机制，其中最主要的是消息机制。对于C语言，函数的调用在编译的时候会决定调用哪个函数，编译完成之后直接顺序执行，无任何二义性。OC的函数调用成为消息发送。属于动态调用过程。在编译的时候并不能决定真正调用哪个函数（事实证明，在编 译阶段，OC可以调用任何函数，即使这个函数并未实现，只要申明过就不会报错。而C语言在编译阶段就会报错）。只有在真正运行的时候才会根据函数的名称找 到对应的函数来调用。

**Runtime实际应用场景**

**1.将某些OC代码转为运行时代码，探究底层，比如block的实现原理**

2.**拦截系统自带的方法调用（Swizzle 黑魔法）,也可以说成对系统的方法进行替换**，比如拦截imageNamed:、viewDidLoad、alloc

3.实现分类也可以增加属性；

分类中声明属性，只会生成setter和getter方法的声明，不会生成带“\_”的成员变量和setter和getter方法的实现

添加属性，在setter方法中调用：objc\_setAssociatedObject

获取属性，在getter方法中调用：objc\_getAssociatedObject

4.实现NSCoding的自动归档和自动解档；(不用对每个属性edcode和decode了,如果几十个属性一个个的encode和decode真的很麻烦啊,使用运行时可以遍历出每个对象的属性,数组的方式遍历eccode,decode)

5.实现字典和模型的自动转换(核心就是可以遍历出字典中的每个属性,json解析中大牛框架都用了这个特性,包括MJEXtension,YYModel，jsonModel都是将json转换为字典,再遍历字典中的每个属性来进行modle的转换)

6.动态增加方法  (动态的为某个类或对象增加一个方法)

7.动态变量控制  (动态对某个对象的变量的值进行替换)

8.实现万能控制器跳转产品来一

变态需求,推送过来的消息,要跳转到任意控制器.利用runtime动态生成对象、属性、方法这特性，我们可以先跟服务端商量好，定义跳转规则，比如要跳转到A控制器，需要传属性id、type，那么服务端返回字典给我，里面有控制器名，两个属性名跟属性值，客户端就可以根据控制器名生成对象，再用kvc给对象赋值，这样就搞定了

9.插件开发

通过头文件方法名猜测方法的作用，swizzle 这些方法，插入自己的代码实现插件逻辑

10.Jspath 热更新 也是使用运行时，jspatch 基本上算是黑科技，在线修复版本bug