OC学习笔记

数据类型:

NSInteger,	%li,	Signed integer
NSUInteger,	%lu,	Unsigned integer
BOOL,	%I,	Boolean (YES/NO)
CGFloat,	%f,	Floating point

基本类型NSObject:

NSString(不可变长度字符串类)， 字符串内存地址在堆和栈和全局区域都有可能，具体放在哪个区域可以根据isa指针查看父类。
NSMutableString(可变长度字符串类)，可以在字符串任意位置插入字符。
NSNumber(数字类)，是一个面向对象的数字类型，包括int，float。
NSArray(可变长度数组类)，用于存储NSObject对象的扩容数组。
NSDictionary(字典类)，用于存储键值对映射关系的哈希表。

面向对象

类定义

```
@interface MyObj : NSObject

@public(公有)

@protected(保护)

@private(私有)

@end
```

类别

```
@interface MyObj(Categories)
    只能有方法，没有属性
@end
```

类扩展

```
@interface MyObj()
    私有属性
    私有方法
@end
```

类实现

```
@implementation MyObj
    方法实现
@end

@implementation MyObj(Categories) 类别的实现
    方法实现
```

@end

类成员方法
- (Return Type) 方法名字: 参数

类方法
+ (Return Type) 方法名字: 参数

类属性
@property(标记属性关键字，还会声明方法和成员变量，如果在类别里定义只会声明方法)
@synthesize(直接生成 getter 和 setter 函数，可以重写)
Readwrite(可读可写，getter 和 setter 方法)
Readonly(只能读，只有 getter 方法)
strong(强引用)
weak(弱引用)
assign(不会释放对象，引用计数为 1)
Copy(赋值之后是深拷贝，创建一个对象，计数为 1)
retain(只有 setter 方法，直接释放旧对象)

协议 (类似于 C++ 纯虚函数，用作接口)
@protocol MyInterface<NSObject>

@required(必须要实现)
- (void)Fun1

@optional(可选择实现)
- (void)Fun2;

@end

@interface MyObj : NSObject<MyInterface1, MyInterface2>

@end

OC 函数调用 (消息传递)
函数调用可以这么写
- id returnValue = [someObj messageName:parameter];

他的实质是，someObj 叫做接收者，messageName 是选择器,选择器及 parameter 参数，一起被称为消息 Message，会把他转化为标准 C 语言调用 objc_msgSend，即下面函数。
- void objc_msgSend(id self,SEL cmd,...);

编译器有个尾调用优化机制，如果某个函数里面是调用另一个函数，且某个函数最后一项操作也是调用另一个函数，并且调用的函数不作为返回值另做他用，则在递归操作是，这某个函数的栈帧会重复利用。
下面递归返回可以进行尾调用优化
- return [self message:someMsg];

下面递归返回则不能优化，因为虽然调用了函数，但最终调用的函数是为返回值做准备的。

- return [self message:someMsg]+1;

Block 语法

和 C 函数指针声明类似，类似于 C++ 的 lambda 表达式

- returnType (^blockName)(parameterTypes) = ^returnType(parameters) {...};