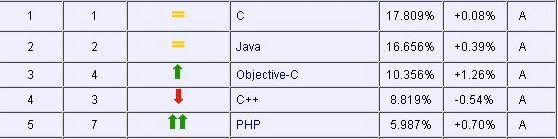
# OC简介

* C语言的基础上，增加了一层最小的面向对象语法
* 完全兼容C语言
* 可以在OC代码中混入C语言代码，甚至是C++代码
* 可以使用OC开发Mac OS X平台和iOS平台的应用程序



# OC语法预览

## 关键字

* **基本上所有关键字都是以@开头**
* 下面列举一些常见的关键字，看一眼就行了，不用去翻资料查找它们的含义

@interface、@implementation、@end

@public、@protected、@private、@selector

@try、@catch、@throw、@finally

@protocol、@optional、@required、@class

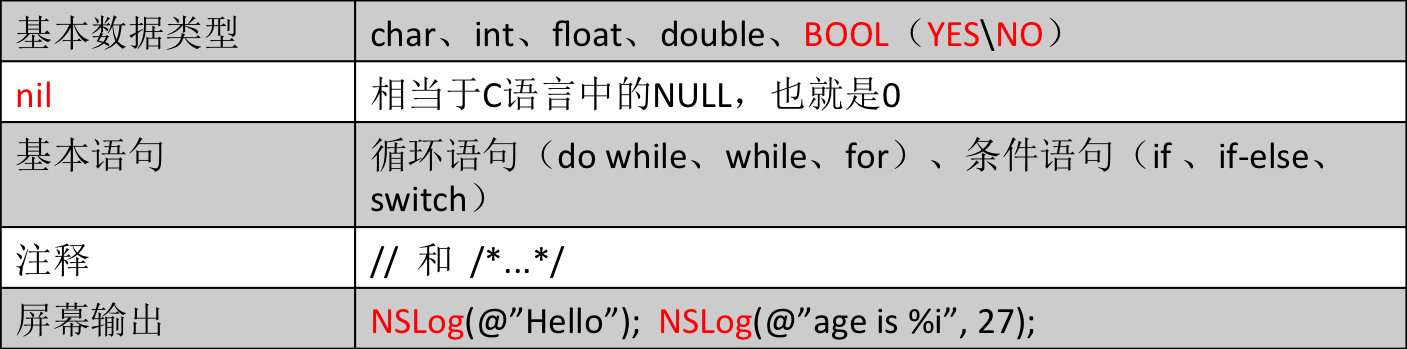
@property、@synthesize、@dynamic

self、super、id、\_cmd、\_\_block、\_\_strong、\_\_weak、

## 字符串以@开头

比如@"Hello"是OC中的字符串，而"Hello"则是C语言中的字符串

## 其他语法



# OC程序的开发过程

与C语言类似：



# 第1个OC程序

## 代码编写

跟C语言一样，OC程序的入口依然是main函数，只不过写到一个.m文件中。比如这里写到一个main.m文件中(文件名可以是中文)

#include <stdio.h>

int main()

{

printf("第1个OC程序\n");

return 0;

}

## 终端指令

* 编译cc –c main.m
* 链接cc main.o
* 运行./a.out

# 第2个OC程序

## 代码编写

来点跟C语言不一样的，使用NSLog函数输出内容

#import <Foundation/Foundation.h>

int main()

{

NSLog(@"第2个OC程序");

return 0;

}

## 终端指令

* 编译cc –c main.m
* 链接cc main.o –framework Foundation
* 运行./a.out

## NSLog与printf的区别

* NSLog接收OC字符串作为参数，printf接收C语言字符串作为参数
* NSLog输出后会自动换行，printf输出后不会自动换行
* 使用NSLog需要#import <Foundation/Foundation.h>
* 使用printf需要#include <stdio.h>

## #import的作用

* 跟#include一样，用来拷贝某个文件的内容
* 可以自动防止文件内容被拷贝多次，也就以为着头文件中不用加入下面的预处理指令了

#ifndef \_STDIO\_H\_

#define \_STDIO\_H\_

#endif

## Foundation框架的作用

* 开发OC、iOS、Mac程序必备的框架
* 此框架中包含了很多常用的API（应用编程接口）
* 框架中包含了很多头文件，若想使用整个框架的内容，包含它的主头文件即可

#import <Foundation/Foundation.h>

## BOOL的使用

* BOOL类型的本质

typedef signed char BOOL;

* BOOL类型的变量有2种取值：YES、NO

#define YES (BOOL)1

#define NO (BOOL)0

* BOOL的输出(当做整数来用)

NSLog(@"%d %d", YES, NO);

# 第3个OC程序

## 多个.m文件的开发

跟C语言中多个.c文件的开发是一样的

1. 编写3个文件

* main.m

#import "one.h"

int main()

{

test();

return 0;

}

* one.h

void test();

* one.m

#import <Foundation/Foundation.h>

void test()

{

NSLog(@"调用了test函数");

}

1. 终端指令

* 编译：cc –c main.m test.m
* 链接：cc main.o test.o –framework Foundation
* 运行：./a.out

## .m文件和.c文件混用开发

1. 编写3个文件

* main.m

#import "one.h"

int main()

{

test();

return 0;

}

* one.h

void test();

* one.c

#include <stdio.h>

void test()

{

printf("调用了test函数\n");

}

1. 终端指令

* 编译：cc –c main.m test.m
* 链接：cc main.o test.o
* 运行：./a.out

(没有使用Foundation框架的话，就不用-framework Foundation)