**01 string**

**main.m**

/\*

NSString：不可变字符串

NSMutableString：可变字符串

\*/

#import <Foundation/Foundation.h>

int main(int argc, const char \* argv[]) {

/\*

1、字符串的创建

\*/

NSString \*s1 = @"jdshfa";

NSString \*s2 = [[NSString alloc] initWithFormat:@"string is %d", 10];

//C字符串-》OC字符串

NSString \*s3 = [[NSString alloc] initWithUTF8String:"jakljdf"];

//OC字符串-》C字符串

char \*cs = [s3 UTF8String];

//NSUTF8StringEncoding 用到中文就可以用这种编码

NSString \*s4 = [[NSString alloc] initWithContentsOfFile:@"/Users/xuhui/Desktop/iOS2016/main.m" encoding:NSUTF8StringEncoding error:nil];

//URL：资源路径

//协议头://路径

//file://

//http://

//ftp://

NSURL \*url = [[NSURL alloc] initWithString:@"http://www.baidu.com"];

NSURL \*url1 = [NSURL fileURLWithPath:@"/Users/xuhui/Desktop/iOS2016/main.m"];

//NSURL \*url = [[NSURL alloc] initWithString:@"<file:///Users/xuhui/Desktop/iOS2016/main.m>"];

NSString \*s5 = [[NSString alloc] initWithContentsOfURL:url encoding:NSUTF8StringEncoding error:nil];

NSLog(@"%@",s5);

//一般都会有一个类方法跟对象方法配对

// [NSURL URLWithString:<#(nonnull NSString \*)#>];

// [NSString stringWithFormat:@""];

// [NSString stringWithContentsOfFile:<#(nonnull NSString \*)#> encoding:<#(NSStringEncoding)#> error:<#(NSError \* \_Nullable \_\_autoreleasing \* \_Nullable)#>];

// [NSString stringWithContentsOfURL:<#(nonnull NSURL \*)#> encoding:<#(NSStringEncoding)#> error:<#(NSError \* \_Nullable \_\_autoreleasing \* \_Nullable)#>];

//字符串的导出

[@"hello world" writeToFile:@"/Users/xuhui/Desktop/iOS2016/my.txt" atomically:YES encoding:NSUTF8StringEncoding error:nil];

NSMutableString \*sm1 = [NSMutableString stringWithFormat:@"age is 10"];

//拼接内容到s1的后面

[sm1 appendString:@" 11 12"];

//获取is的范围

NSRange range = [sm1 rangeOfString:@"is"];

[sm1 deleteCharactersInRange:range];

//[sm1 deleteCharactersInRange:NSMakeRange(4, 2)];

NSLog(@"sm1 = %@",sm1);

return 0;

}

**02 NSArray**

**Person.h**

#import <Foundation/Foundation.h>

@interface Person : NSObject

@end

**Person.m**

#import "Person.h"

@implementation Person

@end

**main.m**

/\*

NSArray:不可变数组

NSMutableArray:可变数组

\*/

int main(int argc, const char \* argv[]) {

/\*

C语言数组

int ages[5] = {1,2,3,4,5};

Person \*p = [[Person alloc] init];

Person \*persons[5] = {p,[[Person alloc] init]};

\*/

//OC数组不能存放nil值

//OC数组只能存放OC对象，不能存放非OC对象类型，比如int,struct,enum等

//1、NSArray的创建爱你

//这个array永远是空数组

NSArray \*array = [NSArray array];

NSArray \*array2 = [NSArray arrayWithObject:@"Jact"];

//nil是数组元素结束的标志

NSArray \*array3 = [NSArray arrayWithObjects:@"Jack",@"Rose", nil];

//NSArray \*array4 = [NSArray arrayWithObjects:@"Jacr",@"Rose",@"jfkal",@"agr", nil];

//快速创建一个NSArray对象

NSArray \*array4 = @[@"Jacr",@"Rose",@"jfkal",@"agr"];

//2、NSArray的元素个数

//[array2 count];

NSLog(@"array3's number is %ld",array3.count);

//3、NSArray中元素的访问

NSLog(@"数组第二个元素是%@",[array3 objectAtIndex:1]);

NSLog(@"数组第一个元素是%@",array3[0]);

Person \*p = [[Person alloc] init];

NSArray \*array5 = @[p,@"Jack"];

//id obj代表着数组中的每一个元素

for ( id obj in array5) {

//找出obj元素在数组中的位置

NSUInteger i = [array5 indexOfObject:obj];

NSLog(@"%ld-%@",i,obj);

}

//每遍历到一个元素，就会调用一次block，并且当前元素和索引位置当做参数传给block

[array5 enumerateObjectsUsingBlock:^(id \_Nonnull obj, NSUInteger idx, BOOL \* \_Nonnull stop) {

NSLog(@"%ld-%@",idx,obj);

if (idx == 0) {

//停止遍历

\*stop = YES;

}

}];

return 0;

}

**03 NSMutableArray**

**Person.h**

#import <Foundation/Foundation.h>

@interface Person : NSObject

@end

**Person.m**

#import "Person.h"

@implementation Person

@end

**Main.m**

#import <Foundation/Foundation.h>

#import "Person.h"

int main(int argc, const char \* argv[]) {

//@[]只创建不可变数组

// NSMutableArray \*arry1 = @[@"hello",@"world"];

// [arry1 addObject:@"jack"];

//NSMutableArray \*array = [NSMutableArray array];

NSMutableArray \*array = [NSMutableArray arrayWithObjects:@"hello",@"world", nil];

//添加元素

[array addObject:[[Person alloc] init]];

[array addObject:@"Jack"];

[array removeObjectAtIndex:1];

//删除元素

//[array removeAllObjects];

//删除指定对象

//[array removeObject:@"Jack"];

NSLog(@"array's number is %ld",array.count);

NSLog(@"%@",array);

return 0;

}

**04 NSSet**

**main.m**

/\*

NSSet和NSArray的对比

1、共同点

\*都是集合，都能存放多个OC对象

\*只能存放OC对象，不能存放非OC对象类型，如基本数据类型

\*都有一个可变的子类，本身都是不可变的

2、不同点

\*NSArray是有顺序的，NSSet是无序的

\*/

#import <Foundation/Foundation.h>

int main(int argc, const char \* argv[]) {

NSSet \*s = [NSSet set];

NSSet \*s2 = [NSSet setWithObjects:@"Hello",@"World",@"Jack",@"Mary",@"Rose",nil];

NSLog(@"s2's count is %ld",s2.count);

//随机拿出一个元素

NSString \*str = [s2 anyObject];

NSLog(@"%@",str);

NSMutableSet \*ms = [NSMutableSet set];

//添加元素

[ms addObject:@"hello"];

//删除元素

[ms removeObject:@"hello"];

return 0;

}

**05 NSDictionary**

**main.m**

#import <Foundation/Foundation.h>

int main(int argc, const char \* argv[]) {

/\*

字典：

key--value

索引--文字内容

里面存储的东西都是键值对

字典不允许有相同的key

字典里面是无序的

\*/

NSArray \*keys = @[@"name",@"address"];

NSArray \*objs = @[@"Jack",@"cumt"];

NSDictionary \*dic = [NSDictionary dictionaryWithObjects:objs forKeys:keys];

NSDictionary \*dic1 = [NSDictionary dictionaryWithObjectsAndKeys:@"Jack",@"name",

@"cumt",@"address",

@"785956678",@"qq",

nil];

NSDictionary \*dic2 = @{@"name":@"Jack",@"address":@"cumt"};

//id obj = [dic1 objectForKey:@"qq"];

id obj = dic2[@"name"];

NSLog(@"%@",obj);

//返回键值对的个数

NSLog(@"%ld",dic2.count);

[dic1 enumerateKeysAndObjectsUsingBlock:^(id \_Nonnull key, id \_Nonnull obj, BOOL \* \_Nonnull stop) {

NSLog(@"dic1:%@-%@",key,obj);

}];

NSMutableDictionary \*mdic = [NSMutableDictionary dictionary];

//添加键值对，键值对没有顺序之分

[mdic setObject:@"Jack" forKey:@"name"];

[mdic setObject:@"cumt" forKey:@"address"];

//覆盖掉相同key的值

[mdic setObject:@"Rose" forKey:@"name"];

NSLog(@"%@",mdic);

//移除键值对

[mdic removeObjectForKey:@"name"];

//错误

// NSMutableDictionary \*mdic2 = @{@"name":@"Jack"};

// [mdic2 setObject:@"cumt" forKey:@"address"];

return 0;

}