编程小翁

博客园 首页 新随笔 联系 订阅 管理

随笔-182 文章-1 评论-43

目前最细致清晰的NSDictionary以及NSMutableDictionar v用法总结

做过Java语言 或者 C语言 开发的朋友应该很清楚 关键字map 吧,它可以将数据以键值对儿的形式储存起来,取值的时候通过KEY就可以直接拿到对应的值,非常方便。在Objective-C语言中 词典对象就是做这个事情的,不过在同一个词典对象中可以保存多个不同类型的数据,不像Java与C 只能保存声明的相同类型的数据,它的关键字为NSDictionary与NSMutableDictionary。阅读过我之前文章的朋友应该从关键字的 结构就可以看出这两个的区别。很明显前者为不可变词典,或者为可变词典。

NSDictionary *dict;

```
for(NSString * akey in dict)
{
//......
```

}很好用

1.创建不可变词典

[NSDictionary dictionaryWithObjectsAndKeys:..]:使用键值对儿直接创建词典对象,结尾必需使用nil标志结束。

[NSDictionary initWithObjectsAndKeys:..]:使用键值对儿初始化词典对象,结尾必需使用nil标志结束。

[dictionary count]: 得到词典的长度单位。

[dictionary keyEnumerator]: 将词典的所有KEY储存在NSEnumerator中,NSEnumerator很像Java语言 中的迭代器,使用快速枚举可以遍历词典中所有储存KEY值。

[dictionary objectEnumerator]: 将词典的所有value储存在NSEnumerator中,用法和上面差不多可用来遍历KEY对应储存的Value值。

[dictionary objectForKey:key]: 通过传入KEY对象可以拿到当前KEY对应储存的值。

```
#import <UIKit/UIKit.h>
#import "MyClass.h"
int main(int argc, char *argv[])
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    //添加我们的测试代码
   NSDictionary *dictionary = [NSDictionary dictionaryWithObjectsAndKeys:@"雨松
MOMO", @"name", @"15810463139", @"number", nil];
    //得到词典的数量
    int count = [dictionary count];
    NSLog(@"词典的数量为: %d",count);
    //得到词典中所有KEY值
   NSEnumerator * enumeratorKey = [dictionary keyEnumerator];
    //快速枚举遍历所有KEY的值
    for (NSObject *object in enumeratorKey) {
       NSLog(@"遍历KEY的值: %@",object);
    //得到词典中所有Value值
    NSEnumerator * enumeratorValue = [dictionary objectEnumerator];
    //快速枚举遍历所有Value的值
    for (NSObject *object in enumeratorValue) {
       NSLog(@"遍历Value的值: %@",object);
    //通过KEY找到value
    NSObject *object = [dictionary objectForKey:@"name"];
    if (object != nil) {
       NSLog(@"通过KEY找到的value是: %@",object);
    int retVal = UIApplicationMain(argc, argv, nil, nil);
```

昵称:编程小翁 园龄:3年3个月 粉丝:145 关注:10 +加关注

<		20	012年3	月		>
日	_	=	Ξ	四	五	<u>'\</u>
26	27	28	29	1	2	3
4	5	<u>6</u>	<u>7</u>	8	<u>9</u>	10
<u>11</u>	12	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
18	19	20	21	<u>22</u>	23	24
25	26	27	28	<u>29</u>	30	31
1	2	3	4	5	6	7



常用链接	
我的随笔	
我的评论	
我的参与	
最新评论	
我的标签	
更多链接	J

我的标签
ios(30)
OpenCV(9)
内存管理(6)
c++(5)
MFC(4)
SQLite3(4)
总结(4)
文件操作(3)
UITableView(3)
NSDate(3)
更多

随笔分类	
C/C++(28)	
IOS开发(67)	
MFC(5)	
My GitHub	
OpenCV(8)	
机器学习(4)	
计算机视觉(6)	
经典资源收藏(1)	

```
return retVal;
}
```

```
All Output +

Attaching to process 633.
2011-09-26 21:43:54.853 HelloWorld[633:207] 词典的数量为: 2
2011-09-26 21:43:54.858 HelloWorld[633:207] 適历KEY的值: name
2011-09-26 21:43:54.858 HelloWorld[633:207] 適历KEY的值: number
2011-09-26 21:43:54.859 HelloWorld[633:207] 適历Value的值: 雨松MOMO
2011-09-26 21:43:54.859 HelloWorld[633:207] 適历Value的值: 15810463139
2011-09-26 21:43:54.860 HelloWorld[633:207] 通汀KEY找到的value是: 雨松MOMO
```

2.创建可变词典对象

NSMutableDictionary 是NSDictionary的子类,所以继承了NSDictionary的方法。

[NSMutableDictionary dictionaryWithCapacity:10]: 创建一个可变词典初始指定它的长度为10.,动态的添加数据如果超过10这个词典长度会自动增加,所以不用担心数组越界。推荐用这种方式

[NSMutableDictionary initWithCapacity:10] :只是初始化一个词典的长度为10。

[dictionary setObject:@"雨松MOMO" forKey:@"name"]:向可变的词典动态的添加数据, 这里的<u>key是name, 值是雨松MOMO</u>。如果词典中存在这个KEY的数据则直接替换这个KEY的值。(易混的地方,慎重!)

[dictionary removeAllObjects..]:删除掉词典中的所有数据。

[dictionary removeObjectForKey..]:删除掉词典中指定KEY的数据。

```
#import <UIKit/UIKit.h>
#import "MyClass.h"
int main(int argc, char *argv[])
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    //添加我们的测试代码
    //创建词典对象, 初始化长度为10
    NSMutableDictionary *dictionary = [NSMutableDictionary dictionaryWithCapacity:10];
    //向词典中动态添加数据
    [dictionary setObject:@"雨松MOMO" forKey:@"name"];
    [dictionary setObject:@"15810463139" forKey:@"number"];
    //通过KEY找到value
    NSObject *object = [dictionary objectForKey:@"name"];
    if (object != nil) {
       NSLog(@"通过KEY找到的value是: %@",object);
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal:
```

All Output ‡ GDB is free software, covered by the GNU General Public License, and you are welcome to change it and/or distribute copies of it under certain conditions. Type "show copying" to see the conditions. There is absolutely no warranty for GDB. Type "show warranty" for details. This GDB was configured as "x86_64-apple-darwin".sharedlibrary apply-load-rules all Attaching to process 699. 2011-09-26 22:00:19.450 HelloWorld[699:207] 通过KEY找到的value是: 兩松MOMO

词典类的存在就是为了解决在大量数据中查找方便,因为它是通过key直接找到value所以速度很快,避免一个个的遍历寻找造成的效率低下,善用字典类会帮你的程序提速。

文章出处: http://blog.csdn.net/xys289187120/article/details/682373

标签: NSDictionary, NSMutableDictionary

绿色通道: 好文要顶」 (关注我) (收藏该文) 与我联系

```
跨平台(1)
设计模式(2)
视频分析(1)
数据库(1)
算法(2)
```

linkInBlog

iOS事件分发机制(一) hit-Testing

最新评论

1. Re: 【原】PSD图标素材的全自动切图方法,适用于IOS、安卓、web前端等领域

html5+css3切图 www.qietu.com

--psd2html

2. Re:【原】iOS中动态添加属性的方法——关联(e.g. 向Category添加成员变量)

收获很多(>_<)

--车宝大人

3. Re:【转】UIWebView获取当前页面url的 两种方法

你好请问如何实时监听uiwebview的url变化,我这边遇到一个需求是网页是单页面应用,就是点链接后网页还是原来那个网页地址,但是url后面的参数会发生变化。我需要获取这些不断变化的url地址。

--前端小尚

4. Re: 【原】IOS合并lib(.a)库的终极可用方法(可用于解决duplicate symbol静态库冲突)

博主您好: ar -x libavcodec_i386.a分离.o没问题,但是ar -x libavcodec_arm64.a时提示ar: libavcodec_arm64.a: Inappropria......

--西江月al

5. Re:【原】iOS运行时runtime初探(强制获取并修改私有变量,强制增加及修改私有方法等)

mark

--Json

阅读排行榜

- 1. 目前最细致清晰的NSDictionary以及NSM utableDictionary用法总结(43694)
- 2. UITextField的总结(43594)
- 3. 【转】APNs消息推送完整讲解(17913)
- 4. 【转】IOS数据库操作SQLite3使用详解(1 7749)
- 5. 【转】好东西! sqlite3中BLOB数据类型存储大对象运用示例(16984)

评论排行榜

- 1. 【原】就IOS发布app时如何保护文本资源的一个方法(6)
- 2. UITextField的总结(3)
- 3. 目前最细致清晰的NSDictionary以及NSM utableDictionary用法总结(3)
- 4. (转) UIWebView的基本用法,适合新 手(2)
- 5. UIPopoverController的使用(2)