

title: Block的使用

date: 2016-06-06 21:02:09

tags:

1.什么是Block?

- Block是iOS中的数据类型（定义变量、形参、返回值）。
- Block是苹果官方推荐使用的数据类型。
- 动画。
- 多线程。
- 集合遍历。
- 网络请求的回调。
- Block的作用
- 用来保存某一段代码，可以在恰当的时间再取出来调用。
- 功能类似于函数和方法。

2.Block的格式

- Block的定义格式

...

返回值类型(block变量名)(形参列表) = 形参列表 {

};

- 指向函数的指针

void printRose()

```
{  
printf(" {@}\n");  
printf(" | \n");  
printf(" \/\n");  
printf(" \n");  
}
```

//void代表指向的函数没有返回值

//()代表指向的函数没有形参

/*(* roseP)代表roseP是一个指向函数的指针

void (*roseP) ();

roseP = printRose;

roseP();

- 定义一个Block变量(和定义指向函数的指针很像)

```
//void代表Block将来保存的代码没有返回值
//()代表Block将来的保存的代码没有形参
//(roseBlock)代表roseBlock是一个Block变量，可以用于保存一段Block代码
void (^roseBlock) ();
roseBlock = {
printf(" {@}\n");
printf(" | \n");
printf(" \\ \n");
printf(" \n");
};
//调用Block才会执行Block保存的代码
roseBlock();
...
```

- 带参数的Block

```
void (^roseBlock) (int);
roseBlock = ^(int num){
for (int i; i < num; i++) {
printf(" {@}\n");
printf(" | \n");
printf(" \\ \n");
printf(" \n");
}
};
roseBlock(2);
```

- typedef 和 Block

```
//利用typedef给Block起别名，和函数的指针一样，Block变量的名称就是别名
typedef int (^caculte) (int,int);
//实现
caculte sum = ^(int value1, int value2){
return value1 + value2;
};

NSLog(@"sum = %i",sum(5,7));

caculte minus = ^(int value1, int value2){
return value1 - value2;
};
NSLog(@"sum = %i",minus(5,7));
```

