

对象初始化

Objective-C有两种创建新对象的不同方法：

1. [类名 new];
2. [[类名 alloc] init];

这两种方法是等价的，不过，通常的 Cocoa 惯例是使用alloc和init，而不使用 new。

- 分配对象（为对象分配内存）

“

分配（allocation）是一个新对象诞生的过程。是从操作系统获得一块内存并将其指定为存放对象的实例变量的位置。向某个类发送 alloc 消息的结果，就是为该类分配一块足够大的内存，以存放该类的全部实例变量。

“

分配内存时，alloc 方法将这块内存区域全部初始化为0。所有的 BOOL 类型变量被初始化为 NO，所有的 int 类型变量被初始化为 0，所有的 float 类型变量被初始化为 0.0，所有的指针被初始化为 nil。

- 初始化对象

“

刚刚分配（alloc）的对象需要经过初始化（initialization）才能使用

```
Car *car = [[Car alloc] init];      // right
V.S.
Car *car = [Car alloc];[car init];  // wrong
```

因为初始化方法返回的对象可能与分配的对象不同

“

为什么init方法会返回不同的对象？

诸如NSString和NSArray这样的类实际上是类簇，由于init方法可以接收参数，因此，其代码能检查接收的参数，并决定返回类簇中哪一个类的对象更合适。比如，参数是文本和参数是阿拉伯数字，返回的可能就是不同的类

- 编写初始化方法

```
if (self = [super init]) {
```

- 便利初始化方法

“

许多类包含便利初始化方法（convenience initializer），它们是用来完成某些额外工作的初始化方法，可以减少你完成这些工作的麻烦。

```
- (id) initWithFormat: (NSString *) format, ...;
- (id) initWithContentOfFile: (NSString *) path;
```

- 指定初始化方法

“

指定初始化方法（designated initializer），指在类中出现2个或2个以上便利初始化方法时，把类中的某个初始化方法被指派为指定初始化方法。该类的所有初始化方法使用指定初始化方法执行初始化操作。子类使用其超类的指定初始化方法实现超类的初始化操作。通常，接受参数最多的初始化方法最终成为指定初始化方法。

- 初始化方法规则

1. 如果不需要设置任何状态，则不需要为你自己的类创建初始化方法
2. 如果初始化方法不止一个，则需要选择一个作为指定初始化方法
3. 如果创建了一个指定初始化方法，则一定要在你自己的指定初始化方法中调用超类的指定初始化方法。其他的初始化方法使用指定初始化方法来实现。