

extern、static、const

<https://www.jianshu.com/p/3fa703e80720>

const

只读，修饰基本数据变量p，指针变量*p。

```
1  int a = 20;
2  a = 30;
3
4  const int b = 20;
5  int const b = 20;
6  b = 30; // 错误，不可修改
7
8  int *p = &a;
9  int c = 40;
10 p = &c; // 允许修改指向的地址
11 *p = 20; // 允许修改指向的地址的值
12
13 const int *p1; // *p1: 常量 p1:变量
14 int const *p1; // *p1: 常量 p1:变量
15 // const修饰指针变量p1
16 int * const p1; // *p1:变量 p1:常量
17 // 第一个const修饰*p1 第二个const修饰 p1
18 // 两种方式一样
19 const int * const p1; // *p1: 常量 p1: 常量
20 int const * const p1; // *p1: 常量 p1: 常量
21
```

static

- 修饰全局变量

在全局变量添加static，就把全局变量变成了静态全局变量，在整个工程执行期间存在，小说里有的sharedInstance就是static修饰单例。

小说里的单例一般都是HTSService，保持不被回收就要isServicePersistent = YES。
static放在shareInstance的目的，返回对象只有这一个，那么它的生命周期就是跟该项目生命周期相同。

存储区：静态存储区，在整个程序运行期间都存在

作用域：全局静态变量在声明他的文件之外是不可见的。准确地讲从定义之处开始到文件结尾。非静态全局变量的作用域是整个源程序（多个源文件可以共同使用）；而静态全局变量则限制了其作用域，即只在定义该变量的源文件内有效，在同一源程序的其它源文件中不能使用它。

- 修饰局部变量

存储区：由栈变成了静态存储区，生存周期变化。

作用域：不变。

- 修饰函数

静态函数只能被本文件可见，不能被其他文件呢所用。

extern

只能用来修饰全局变量。使得可以跨文件访问。

- 与.h文件的关系

```
1  #import <Foundation/Foundation.h>
2  @interface ExternModel : NSObject
3      // 这个是不合法的：因为OC的类会将这个全局变量当做成员属性来处理，而成员属性是需要加{}的，
   所以不合法
4      NSString *lhString; //声明全局变量的时候默认带有extern，这里必须显式声明
5  @end
6
7  #import <Foundation/Foundation.h>
8  @interface ExternModel : NSObject
9      extern NSString *lhString; //这里由于带有extern所以会被认为是全局变量
10 @end
11
12
```

