



“const” 关键字的用法

1) 定义常量

```
const int MAX_VAL = 23;
```

```
const string SCHOOL_NAME = "Peking University"  
;
```

2) 定义常量指针

□ 不可通过常量指针修改其指向的内容

```
int n,m;  
const int * p = & n;  
* p = 5;  
n = 4;  
p = &m;
```

2) 定义常量指针

□ 不可通过常量指针修改其指向的内容

```
int n,m;  
const int * p = & n;  
* p = 5; //编译出错  
n = 4;  
p = &m;
```

2) 定义常量指针

□ 不可通过常量指针修改其指向的内容

```
int n,m;  
const int * p = & n;  
* p = 5; //编译出错  
n = 4;   //ok  
p = &m;
```

2) 定义常量指针

□ 不可通过常量指针修改其指向的内容

```
int n,m;
```

```
const int * p = &n;
```

```
*p = 5; //编译出错
```

```
n = 4; //ok
```

```
p = &m; //ok, 常量指针的指向可以变化
```

2) 定义常量指针

- 不能把常量指针赋值给非常量指针，反过来可以

```
const int * p1; int * p2;
```

```
p1 = p2; //ok
```

```
p2 = p1; //error
```

```
p2 = (int * ) p1; //ok, 强制类型转换
```

2) 定义常量指针

- 函数参数为常量指针时，可避免函数内部不小心改变参数指针所指地方的内容

```
void MyPrintf( const char * p )  
{  
    strcpy( p, "this"); //编译出错  
    printf("%s", p);    //ok  
}
```


3) 定义常引用

- 不能通过常引用修改其引用的变量

```
int n;
```

```
const int & r = n;
```

```
r = 5; //error
```

```
n = 4; //ok
```



下一小节：动态内存分配