

# 指针

## for+遍历

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};  
//1、  
for(int i=0;i<10;i++)  
    cout << a[i] <<" ";  
//2、  
for(int* p=a;p<a+10;p++)  
    cout << *p <<" ";  
cout <<endl;  
输出 1 2 3 4 5 6 7 8 9 10
```

## 深度理解数组和指针的关系

```
//int变量类型的指针  
int num = 1;  
int* m = &num;  
  
//int数组的指针  
int a[10] = { 0,1,2,3,4,5,6,7,8,9 };  
int* p = a;//p的值为a的数组首地址  
cout << a << endl;  
cout << p << endl;  
cout << &a << endl;//前3个都是一样的地址  
cout << &p << endl;//输出的是 p指针 的地址 （不和前面的一样）  
  
//cout << ++a << endl;数组是常量不可以改变  
cout << a + 1 << endl;//但可以这样 不会改变a  
  
cout << ++p << endl;//而指针可以改变 ++指向下一个索引的 地址  
cout << &a[1] << endl;  
  
p = a;//初始化  
cout << *(p + 5) << endl; //输出5  
  
p = a;//初始化  
cout << a[3] << endl;  
cout << p[3] << endl;//p可以像数组一样用[]  
输出  
3 3  
  
p++//让p偏移  
cout << a[3] << endl;  
cout << p[3] << endl;  
输出  
3 4
```

## 指针常量与常量指针

## 1. 指针常量与常量指针的概念

指针常量就是指针本身是常量，换句话说，就是指针里面所存储的内容（内存地址）是常量，不能改变。但是，内存地址所对应的内容是可以经过指针改变的。

常量指针就是指向常量的指针，换句话说，就是指针指向的是常量，它指向的内容不能发生改变，不能通过指针来修改它指向的内容。但是，指针自身不是常量，它自身的值可以改变，从而指向另一个常量。

## 2. 指针常量与常量指针的声明

指针常量的声明：数据类型 \* **const 指针变量**。

常量指针的声明：数据类型 **const \* 指针变量**或者 **const 数据类型 \* 指针变量**。

常量指针常量的声明：数据类型 **const \* const 指针变量** 或者 **const 数据类型 \* const 指针变量**。

## 指针相减

结论：指针相减 = (地址1-地址2) / sizeof(数据类型)

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};
int* p = a;
int* m = p++;
cout << "p= " << p << endl;
cout << "m= " << m << endl;
cout << "m - p = " << m - p << endl;
cout << "(char*)m - (char*)p = " << (char*)m - (char*)p << endl;
```

输出

```
p= 0x6bfeb0
m= 0x6bfeb4//地址相差 4个字节 (int)
m - p = 1
(char*)m - (char*)p = 4
```

```
double g[] = { 1,2 };
double* buf= &g[1];
cout << "buf - g = " << buf - g << endl;
cout << "(int*)buf - (int*)g = " << (int*)buf - (int*)g << endl;
cout << "(double*)buf - (double*)g = " << (double*)buf - (double*)g << endl;
cout << "(char*)buf - (char*)g = " << (char*)buf - (char*)g << endl;
```

输出

```
buf - g = 1
(int*)buf - (int*)g = 2
(double*)buf - (double*)g = 1
(char*)buf - (char*)g = 8
```

##