## 相关说法

* 1. 取地址符 &
  2. 解引用 \*

## 指针的\*位置

int \*p;

int \* p;

int\* p;

均可

## 指针的\*两种作用

* 1. 用于创建指针

int \*myPointer = &myInt;

* 1. 对指针进行解引用

\*myPointer = 3394;

## 无类型指针

void \*p //p只是存放一个地址

对它进行解引用之前，应当先转换为一种适当的数据类型

## 图说指针

int a =456;

char b = ‘c’;

int \*aPointer = &a;

int \*bPointer = &b;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量名 | a | b | aPointer | bPointer |
| 内存内容 | 456 | 67 | 0 | 4 |
| 内存地址 | 0123 | 4567 | 89AB | CDEF |

## 指针与数组

以下两句做同样地事情

int \*ptr1 = &myArray[0];

int \*ptr1 = &myArray;

## 指针++

数组上的指针++ ===+sizeof(指针类型);

## 进阶思考

既然指针变量存放地址，为什么不用一个整型变量来存储而发明指针变量呢？

编译器为了方便你写代码而指定的，编译器编译完后，仍旧是一个整型变量