# 数组的基本概念

一个int类型的变量能保存一个人的年龄，如果想保存整个班的年龄呢？

## 什么是数组

数组，从字面上看，就是一组数据的意思，没错，数组就是用来存储一组数据的

## 数组的特点

* 只能存放一种类型的数据，比如int类型的数组、float类型的数组
* 里面存放的数据称为“元素”

# 数组的定义

## 定义

* 声明数组的类型
* 声明数组的元素个数（需要多少存储空间）

## 格式

元素类型 数组名[元素个数];

比如：int ages[3];

## 简单使用

* 简单初始化：int ages[5] = {19, 19, 20, 21, 25};
* 元素有顺序之分，每个元素都有一个唯一的下标（索引），从0开始
* 数组元素的访问：a[i]

## 初始化

* 初始化方式
* int a[3] = {10, 9, 6};
* int a[3] = {10,9};
* int a[] = {11, 7, 6};
* int a[4] = {[1]=11,[0] = 7};
* 常见错误
* int a[];
* int[4] a;
* int a[b];
* a = {10, 11};
* a[4] = {10,9,8,5};

## 内存分析

* 数组存储空间的大小
* 存储空间的划分（内存的分配是从高地址到低地址进行的，但一个数组内部元素又是从低到高进行的）
* 数组名的作用，查看元素地址
* 数组越界的注意

## 其他使用

* 数组与函数参数
* 数组元素作为函数参数
* 数组作为函数参数（sizeof注意）
* 遍历数组元素
* 两种方式遍历（while循环和for循环）
* 遍历元素值和元素地址
* 字符数组的使用

## 习题

1. 提示输入5个学生的成绩，算出平均分、最高分、最低分，然后输出
2. 设计一个函数：int arrayMax(int a[], int count)，找出数组元素的最大值
3. 设计一个函数：int arraySum(int a[], int n)，求一维数组a前n个数的和
4. 设计一个函数：将一维整型数组中的元素逆序存放。比如本来是1,3,4,2，逆序存放就变成了：2，4，3，1

# 二维数组

## 什么是二维数组

* 一个数组能表示一个班人的年龄，如果想表示很多班呢？
* 什么是二维数组？int ages[3][10]; 三个班，每个班10个人
* 相当于3行10列
* 相当于装着3个一维数组
* 二维数组是一个特殊的一维数组：它的元素是一维数组。例如int a[2][3]可以看作由一维数组a[0]和一维数组a[1]组成，这两个一维数组都包含了3个int类型的元素

http://images.cnitblog.com/blog/497279/201303/15143658-9ac5ed1d9a61422aa670fc68e43d6016.png

## 存储

* 存储大小
* 存储结构和顺序
* 存储地址问题

## 初始化

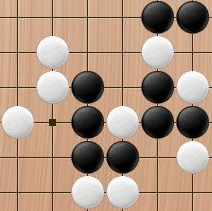
* int a[3][4] = {1,2,3,4,5,6};
* int a[3][4] = {{},{},{}};
* 数组元素简单访问
* int a[][5] = {3,21,31,2,32,1};
* 注意错误：

int a[3][4];

a[3] = {};

## 遍历

* 遍历所有的元素
* 遍历地址
* 使用场合：五子棋、俄罗斯方块

# 字符串

## 什么是字符串

* 简单的字符串”itcast”
* 一个’i’是一个字符
* 很多个字符组合在一起就是字符串了

## 字符串的初始化

* char a[] = “123”; 和 char a [] = {‘1’,’2’,’3’};的区别，可以比较大小
* “123”其实是由’1’、’2’、’3’、’\0’组成
* “123”的存储分布
* 字符串的输出”%s”，’\0’是不会输出的

## \0的作用

* 输出char a[] = {‘o’,’k’};
* 在char a[]前面搞个”mj”
* 输出”mj”
* 再输出a
* char a[] = {‘i’,’t’,’\0’,’c’};

## 常用字符串处理函数

* strlen（注意中文）

## 习题

编写一个函数char\_contains(char str[],char c)，如果字符串str中包含字符c则返回数值1，否则返回数值0

# 字符串数组

## 1.使用场合

\* 一维字符数组中存放一个字符串，比如一个名字char name[20] = "mj"

\* 如果要存储多个字符串，比如一个班所有学生的名字，则需要二维字符数组，char names[15][20]可以存放15个学生的姓名(假设姓名不超过20字符)

\* 如果要存储两个班的学生姓名，那么可以用三维字符数组char names[2][15][20]

## 2.初始化

char names[2][10] = { {'J','a','y','\0'}, {'J','i','m','\0'} };

char names2[2][10] = { {"Jay"}, {"Jim"} };

char names3[2][10] = { "Jay", "Jim" };