编 程 学 习 笔 记

C

Code

作 者 姓 名： 颜佳

目 录

[1 数组 1](#_Toc84795339)

[1.1 数组的初始化 1](#_Toc84795340)

[1.2 声明数组的大小必须满足一下条件： 1](#_Toc84795341)

[1.3 初始化二维数组 1](#_Toc84795342)

[1.4 指针与数组 1](#_Toc84795343)

[1.5 函数、数组和指针 1](#_Toc84795344)

[1.5.1 在函数定义中，以下是等价的 2](#_Toc84795345)

[1.5.2 在函数原型中，以下是等价的 2](#_Toc84795346)

[1.6 #define指令和const 2](#_Toc84795347)

[1.7 指针与多维数组 3](#_Toc84795348)

[1.8 函数与多维数组 3](#_Toc84795349)

[1.9 字符串 3](#_Toc84795350)

[1.9.1 声明 3](#_Toc84795351)

[1.9.2 字符串数组 3](#_Toc84795352)

# 数组

## 数组的初始化

int a[2]={0,1};

const int a[2]= {0,1};//只读数组

数组必须初始化才能使用；

当初始化列表中值得个数小于数组元素个数时，自动初始化为0；

**当初始化列表中值得个数大于数组元素个数时，报错**；

只能在定义时使用花括号进行初始化；

最好使用符号常量声明数组大小

**指定初始化器（C99）**

在编译时，下标越界不报错？？？？

## 声明数组的大小必须满足一下条件：

大于0；

整形常量表达式

sizeof表达式：sizeof(int)+1

不能是变量

## 初始化二维数组

int a[2][2]={{0,0},{1,2}}

int a[2][2]={0,1,2}

## 指针与数组

数组名是数组首元素的地址；

int \*p=a=&a[0]

dates+2 == &dates[2]//地址

\*（dates+2）== dates[2]//数据

## 函数、数组和指针

### 在函数定义中，以下是等价的

int sum (int \*array, int n);//更通用

int sum (int array[], int n);

### 在函数原型中，以下是等价的

int sum (int \*array, int n);

int sum (int \*, int);

int sum (int array[], int n);

int sum (int [], int);

如果函数的意图不是修改数组的数据内容，那么在函数原型和函数定义中应使用const关键字

int sum (const int \*array, int n);

int sum (const int \*, int);

int sum (const int array[], int n);

int sum (const int [], int);

假设start是一个指针

\*start++ == \*start ;start++

\*（start++） == \*start ;start++

(\*start)++ == \*start; (\*start)++//值递增

\*++start == start++ ;\*start

## #define指令和const

#define可创建符号常量；

const关键字更灵活，可以创建变量，数组，指针，指向const的指针；

## 指针与多维数组

int zippo [4][2];

zippo == &zippo[0]

zippo[0] == &zippo[0][0]

zippo == zippo[0]

zippo+1 != zippo[0]+1

\*(zippo[0]) == zippo[0][0]

\*zippo == &zippo[0][0]

\*\*zippo == zippo[0][0]

int (\*p)[2];//数组指针，声明了一个指向一个内含两个int类型的数组的指针

int \*p[2];//指针数组，声明了一个含有两个指针元素的数组，每个元素都指向int类型

## 函数与多维数组

void function(int (\*pt)[4]);

void function(int pt[][4]);

## 字符串

puts()函数只显示字符串，并在字符串末尾加上换行符

### 声明

char words[] = ”abcdefghijklmnopqrstuvwxyz”;

const char \*words = ”abcdefghijklmnopqrstuvwxyz”;

### 字符串数组

char words[5][50] ={ ”abcdefghijklmnopqrstuvwxyz”,

”abcdefghijklmnopqrstuvwxyz”,

”abcdefghijklmnopqrstuvwxyz”};

const char \*words[5] = {”abcdefghijklmnopqrstuvwxyz”,

”abcdefghijklmnopqrstuvwxyz”,

”abcdefghijklmnopqrstuvwxyz”};