

# 基础内容

MP4 1920 X 1080 [2/16] 2-1.类型11.mp4 腾讯课堂

## 变量

是装数据的容器，数据变，容器不变。  
变量在使前要先定义，基本格式为：  
**变量类型名 变量名1, 变量名2 ...;**  
也就是左例中a、b、c三个变量的定义可写为：

```
int a, b, c;  
int a = 0, b, c = 0; //初始化
```

## 变量名

- 是由字母（'\_'被看作字母）和数字组成的序列，但第一个字符必须为字母；
- 大小写字母是有区别的；
- 如int等关键字，已被语言本身占有，不能做为变量名。

天勤率辉考研2021

00:05:53 00:13:08

MP4 1920 X 1080 [2/16] 2-1.类型11.mp4 腾讯课堂

## 施加在基本类型上的限定符

```
short int a; //int可省略不写  
long int a; //int可省略不写
```

short限定符尽可能的使int变短，long尽可能的使int变长。  
short和int类型不小于16位，int型可以是16位和32位，long类型不小与32位，具体由所在机器而定。

signed与unsigned可用来限定char型和所有整型（包括被short或long限定过的整型）。  
signed **整型** 等价于 **整型**。

unsigned **整型**使得原有整型长度不变，符号位变成了数据位，仅能表示0和正数。  
signed char强制char可以存储有符号整数，unsigned char强制char可以存储无符号整数，char 不加任何限定，则是否有符号根据所在机器而定。

天勤率辉考研2021

00:12:46 00:13:08

\a	响铃符	\\	反斜杠
\b	回退符	\?	问号
\f	换页符	\'	单引号
\n	换行符	\"	双引号
\r	回车符	\ooo	八进制数
\t	横向制表符	\xhh	十六进制数
\v	纵向制表符	\0	字符串结束的标识

## 进制

0开头表示八进制

0x开头表示十六进制

## ++/--用法

++/--用法相同

x++ 先返回x的值，然后自增

++x 先自增，然后返回x的值

## x>y ? 'a' : 'b'

x>y ? 'a' : 'b' 如果x>y,那么返回a,否则返回b

## 函数的格式

返回值类型 函数名（参数列表）

```
{  
  声明和语句;  
  return x;  
}
```

## 作用域

内部作用域的赋值可以影响到外部作用域  
而定义不可以

## static静态变量

```
static静态变量  
{  
  {  
    static int i=0;  
  }  
}
```

等同于把局部变量i 改为全局变量



## 指针型数组

### 一维数组

当以数组名带入函数时，数组名表示该数组第一个元素的指针

### 二维数组

`int b[i][j]={}`, `i`表示行, `j`表示列

```
9  #include <iostream>
10 using namespace std;
11 void array2D(int a[][3], int n)
12 {
13     for(int i=0; i<n; ++i)
14     {
15         for(int j=0; j<3; ++j)
16             cout<<a[i][j]<<"\t";
17         cout<<endl;
18     }
19 }
20 int main()
21 {
22     int b[4][3] = {
23         {1, 2, 3},
24         {4, 5, 6},
25         {7, 8, 9},
26         {10, 11, 12}
27     };
28     array2D(b, 4);
29     return 0;
30 }
31
```

1	2	3
4	5	6
7	8	9
10	11	12

### 字符型数组



```

9  #include <iostream>
10 using namespace std;
11 void array2D(char *s2D[])
12 {
13     for(int i=0; i<4; ++i)
14     {
15         cout<<s2D[i]<<endl;
16     }
17 }
18 int main()
19 {
20     char *s2D[] = {
21         "I",
22         "am",
23         "a",
24         "string"
25     };
26     array2D(s2D);
27     return 0;
28 }

```

```

I
am
a
string

```

指向函数的指针

MP4 1920 X 1080 [15/17] 5-3 数组21.mp4

### ● 指向函数的指针

```
9 #include <iostream>
10 using namespace std;
11 int add(int a, int b)
12 {
13     return a + b;
14 }
15 int minu(int a, int b)
16 {
17     return a - b;
18 }
19 int main()
20 {
21     int (*p)(int, int);
22     char op = '+';
23     if(op == '+')
24         p = add;
25     else
26         p = minu;
27     cout<<p(3, 4)<<endl;
28     return 0;
29 }
```

```
9 #include <iostream>
10 using namespace std;
11 int add(int a, int b)
12 {
13     return a + b;
14 }
15 int minu(int a, int b)
16 {
17     return a - b;
18 }
19 int main()
20 {
21     int (*p)(int, int);
22     char op = '-';
23     if(op == '+')
24         p = add;
25     else
26         p = minu;
27     cout<<p(3, 4)<<endl;
28     return 0;
29 }
```

后续课程添加微信号: kaoyan6886  
欢迎关注微信公众号: 大开眼界

天勤率辉考研2021

00:13:01 00:14:06

# 结构体和类

## 结构体

### 类型定义

typedef用来给数据类型取新的名字

ey: typedef(空格)int(空格)MYINT

意为: 把int重新命名为MYINT

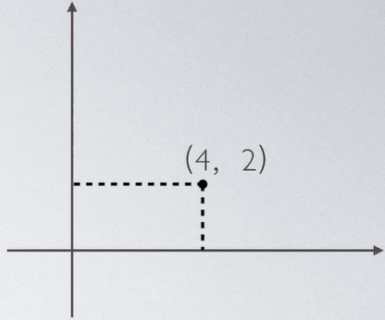
### 结构体定义坐标

before:

● 结构体

```
26 #include <stdio.h>
27 int main()
28 {
29     struct {int x; int y;} point;
30     point.x = 10;
31     point.y = 11;
32     printf("%d, %d", point.x, point.y);
33     return 0;
34 }
```

10, 11



int point\_x;  
int point\_y;

腾讯课堂

天勤率辉考研2021

00:05:21 00:09:39

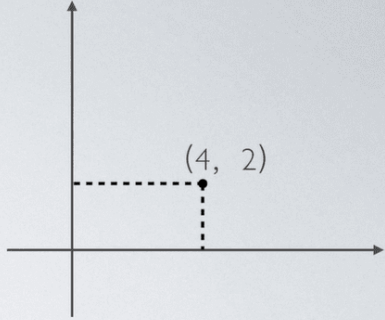
after:

(1)

暂停

```
24 #include <stdio.h>
25 typedef struct
26 {
27     int x;
28     int y;
29 }Point;
30
31 int main()
32 {
33     Point point;
34     point.x = 10;
35     point.y = 11;
36     printf("%d, %d", point.x, point.y);
37     return 0;
38 }
```

10, 11



int point\_x;  
int point\_y;

腾讯课堂

天勤率辉考研2021

00:08:31 00:09:39

(2)

```
24 #include <stdio.h>
25 struct Point
26 {
27     int x;
28     int y;
29 };
30
31 int main()
32 {
33     struct Point point;
34     point.x = 10;
35     point.y = 11;
36     printf("%d, %d", point.x, point.y);
37     return 0;
38 }
```



10, 11

指向结构体的指针



```
24 #include <stdio.h>
25 typedef struct
26 {
27     int x;
28     int y;
29 }Point;
30
31 int main()
32 {
33     Point point;
34     Point *p;
35     p = &point;
36     p->x = 10;
37     p->y = 11;
38     printf("%d, %d", p->x, p->y);
39     return 0;
40 }
```

10, 11

6886

## 自引用结构

暂停

### ● 自引用结构

```
graph TD
    p1["p1 | 1 | 0"] --> p2["p2 | 4 | 1"]
    p2 --> p3["p3 | 2 | 4"]
    p3 --> p4["p4 | 3 | 2"]
    p4 --> p5["p5 | 1 | 6"]
    p5 --> null["^p"]
```

后续课程添加微信号: kaoyan6886  
欢迎关注微信公众号: 大开研界

```
8 #include <stdio.h>
9 typedef struct Point
10 {
11     int x;
12     int y;
13     struct Point* next;
14 }Point;
15
16 int main()
17 {
18     Point p1, p2, p3, p4, p5;
19     Point *p;
20     p1.x = 1; p1.y = 0;
21     p2.x = 4; p2.y = 1;
22     p3.x = 2; p3.y = 4;
23     p4.x = 3; p4.y = 2;
24     p5.x = 1; p5.y = 6;
25
26     p1.next = &p2;
27     p2.next = &p3;
28     p3.next = &p4;
29     p4.next = &p5;
30     p5.next = NULL;
31
32     for(p = &p1; p != NULL; p = p->next)
33         printf("(%d, %d)\n", p->x, p->y);
34
35     return 0;
36 }
```

```
(1, 0)
(4, 1)
(2, 4)
(3, 2)
(1, 6)
```

天勤率辉考研2021

