**数据类型**

char //字符数据类型

例:

#include <stdio.h>

int main() //main前面int(决定返回值类型)表示main函数调用后返回一个整型值

{

char ch= 'A'; //此处必须为''否则代码无法执行

printf(" % c\n",ch); //此处必须为""否则代码无法执行,%c-->char

return 0; //返回0;此处语法上(实际上一般为0)不必为0,整型即可

}

short //短整型

int //整型

例:

#include <stdio.h>

int main()

{

int age= 19;

printf(" % d\n",age); //%d-->int(10进制)

return 0;

}

long//长整型

long long//更长的长整型

float //单精度浮点型

例:

#include <stdio.h>

int main()

{

float age= 19.10;

printf(" % f\n",age); //%f-->float

return 0;

}

double//双精度浮点型

**各数据类型大小**

**计算机单位**

bit - 比特位 ---->> 一个二进制位( 1 or 0 )

byte - 字节 ---->> 八个比特位

kb ---->> 1024个byte

mb ---->> 1024个kb

gb ---->> 1024个mb

tb ---->> 1024个gb

pb ---->> 1024个tb

**使用代码了解数据大小 及 各数据大小**

#include <stdio.h>

int main()

{

printf(" % d\n",sizeof(char)); //必须为%d(输出为整型);sizeof(东西),东西可以是类型也可以是变量

return 0;

}

char1个字节

short 2

int 4

long 4(本编译器)/8 sizeof(long) 》siz**e**of(int)

long long 8

float 4

double 8

**变量 与 常量**

**变量**

**定义变量的方法**

例:

#include <stdio.h>

int main()

{

short age= 20; //此处 age 即为变量

//float weight=90.5 系统对小数默认为双精度浮点型,此时会报错(精度丢失)

float weight=90.5f; //正确写法

char ch = 'w';

return 0;

}

**变量的分类**

#include <stdio.h>

int num1 = 10; //全局变量--定义再代码块{ }外的变量

int main()

{

{

int num2 = 20; //局部变量--定义再代码块{ }内的变量

int num1 = 100; //不会报错,打印结果为100(局部优先)

return 0;

}

printf("%d\n",num2) //报错,num2未定义(局部变量不能在本代码块{ }外使用)

printf("%d\n",num1) //打印为 10 (全局变量能在本代码块{ }外使用)

}

**变量的使用**

以scanf函数为例:

#define \_CRT\_SECURE\_NO\_WARNINGS 1 //使用#def...(必须加在原文件第一排),而不是scanf\_s(部分编译器使用)使代码更通用

#include <stdio.h>

int main()

{

int num1 = 0; //不必为0,甚至可写为: int num1;

int num2 = 0;

int sum=0; //正确位置

scanf("%d%d",&num1,&num2); //使用 输入函数scanf(将变量放入指定地址);取地址符号&

//int sum=0; C语言规定变量要定义在代码块的最前面(老版本,新版本无此语法可任意放置)

sum=num1+num2;

printf("sum=%d\n",sum);

return 0;

}

**变量的作用域与生命周期**

**变量的作用域(scope)**

全局变量的作用域是整个**工程!**

例: test1.c

#include <stdio.h>

int a = 100;

test2.c

#include <stdio.h>

int main()

{

extern int a; //用extern(声明外部符号)让test1.c的全局变量在test2.c中使用

printf("%d is a num\n", a);

return 0;

}

**变量的生命周期**

局部变量的生命周期 : 进入作用域开始,出作用域结束

全局变量的生命周期 : 整个程序的生命周期(main)

**常量**

**常量的分类**

字面常量

const 修饰的常变量

#define 定义的标识符常量

枚举常量

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

enum LikeColour //枚举关键 - enum

{

red,

yellow,

blue, //'red,yellow,blue,black'为枚举常量

black //red-0, yellow-1, blue-2, black-3

}; // ';'不可缺少

int main()

{

2; //字面常量

int num=1; //变量

num=4; //变量可以改变赋值

const int num=1; //const(常属性)修饰的常变量(具有常属性,但本质为变量,即被锁定的变量)

//num=4; 无法可以改变赋值,会报错

#define MAX 10; //#define 定义的标识符常量(也可以放在{}外,与变量用法类似)

enum LikeColour I =red; //枚举常量的使用

I = blue; //变量改变赋值

//blue = 10 枚举常量不可改变,会报错

return 0;

}

**字符串 与 转义字符 与 注释**

**字符串**

用 " " 引起来的一串字符称为字符串字面值,简称字符串

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

int main()

{

"abcdef"; //用""括起来的即为字符串

""; //空字符串

char arrl1[] = "abc" ; //用 数组 存储字符串 ,arrl1--'a','b','c','\0', '\0'字符串的结束标志

char arrl2[] = {'a','b','c'} ; // X --打印结果'abc烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫涕F$p?'

char arrl3[] = {'a','b','c',0} ; // O --打印结果'abc'(用'\0'也行)

printf("%s\n",arrl1); //字符串的占位符--%s

printf("%s\n",arrl2);

printf("%s\n",arrl3);

return 0;

}

注 **:** 字符串的结束标志是一个 \0 的转义字符。在计算字符串长度时 \0 不算字符串内容

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

int main()

{

char arrl1[] = "abc" ;

char arrl2[] = {'a','b','c'} ;

printf("%d\n",strlen(arrl1)); //strlen -- string lenth -- 计算字符串长度, 结果 3

printf("%d\n",strlen(arrl2)); //结果 42(2022VC社区版),abc后无\0没有结束一直随机(近似随机)到0

return 0;

}

**转义字符**

例:

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

int main()

{

printf("abcn");

printf("abc\n"); // '\'转变了'n'的含义-->>换行

printf("C:\test\32\test.c"); //结果 'C: est␦ est.c'

printf("C:\\test\\32\\test.c"); //结果 "C:\test\32\test.c"

return 0;

}

strlen("C:\test\32\test.c") --->13

\32(\ddd) ---> ascii码第26位

转[义字符对照表 - TabKey9 - 博客园 (cnblogs.com)](https://www.cnblogs.com/tabkey9/p/15930390.html)

[Ascii完整码表（256个）\_ascii码表-CSDN博客](https://blog.csdn.net/ttmice/article/details/50978054)

**注释**

/\*

\*/ (有缺陷,-不能嵌套,只到第一个 \*/ 为止)

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

/\*

int main()

{

/\*

printf("废物");

\*/ //注释到此结束,以下内容未被注释

return 0;

}

\*/

**语句分类**

**选择语句**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

int main()

{

int input =0;

printf("你要学C语言吗?(1/0)>:") ;

scanf("%d",&input); //导入,让你输入 东西 (这里指1/0)

if(input==1) //选择语句

printf("祝你成功!\n");

else

printf("O.o ?\n") ;

return 0;

}

**循环语句**

**while循环**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

int main()

{

int line=0;

printf("开始学C语言\n");

while(line<20000)

{

printf("敲一行代码,这是第%d行代码\n",line);

line++;

}

printf("已经循环两万次了");

return 0;

}

**for循环(我还没学到,后面学)**

**do...while循环(后面学)**

**函数**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

int Add(int x, int y)

{

int z =x+y;

return z;

}

int main()

{

int num1 = 10;

int num2 = 20;

int a =100;

int b =200;

int sum=0;

//sum=num1+num2;

sum = Add(num1,num2);

printf("sum=%d\n",sum);

//sum=a+b;

sum = Add(a,b);

printf("sum=%d\n",sum);

return 0;

}

**数组**

**--------**  一组相同类型元素集合

**数组的定义方式**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

int main()

{

int a = 1;

int b = 2;

int c = 3; //太过麻烦,使用数组

int arr[10]={1,2,3,4,5,6,7,8,9,0}; //定义一个存放10个整数数字的数组,arr是数组名称

return 0;

}

**数组的使用**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

int main()

{

int arr[10]={1,2,3,4,5,6,7,8,9,0}; //数组的使用是通过下标来实现的,下标从0开始(如:arr[4]为3)

int stare=0;

while(stare<10)

{

printf("%d ",arr[stare]);

stare++;

}

return 0;

}

**操作符**

算数操作符

+  **- \*** / (商)%(取模,余数)

移位操作符

>> <<

位操作符

&按位与 | 按位或 ^ 按位异或(相同为0,不同为1)

赋值操作符

= += **-**= \*= /= ^= |= >>= <<=

单目操作符

!逻辑反操作

-/+负值/正值

&取地址

sizeof操作数的类型长度(以字节为单位)

~对一个二进制取反

--/++前置(先--/++,再使用),后置(先使用,再--/++)

\*间接访问操作符(解引用操作符)

(类型)强制类型转换

关系操作符

>

>=

<

<=

!=用于测试 ' 不相等 '

==用于测试 ' 相等 '

逻辑操作符

&&逻辑 与

**| |逻辑** 或

条件操作符(三目操作符)

exp1 ? exp2 : exp3 (解释: 1如果真,执行2;否则3)

int main()

{

int a = 10;

int b=20;

int max = (a>b? a : b)

printf("%d \n",max)

return 0;

}

逗号表达式

exp1,exp2,exp3......expn

下标引用 [ ] ,函数调用 ( ) ,结构成员 . ->

[ ] ( ) . ->

**关键字**

register , signed , unsigned , struct , static(局部变量,全局变量,函数) , define...........

**指针......**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

int main()

{

int a = 10;

printf("%p \n",&a) //取地址(%p--取地址占位符,&取地址操作符)

int\* p = &a; //int\* 地址变量

\*p = 20; //\*p--->a(\* -- 解地址符)

}

**结构体**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <string.h>

struct Book

{

char name[20]

short price;

}; //";"不可缺少

int main()

{

struct Book b1 = {"C语言入门",30};

printf("书名:%s\n",b1.name)

b1.price = 15;

struct Book\* pb = &b1;

printf("书名:%s\n",(\*pb).name);

printf("书名:%s\n",pb->name) ;

}