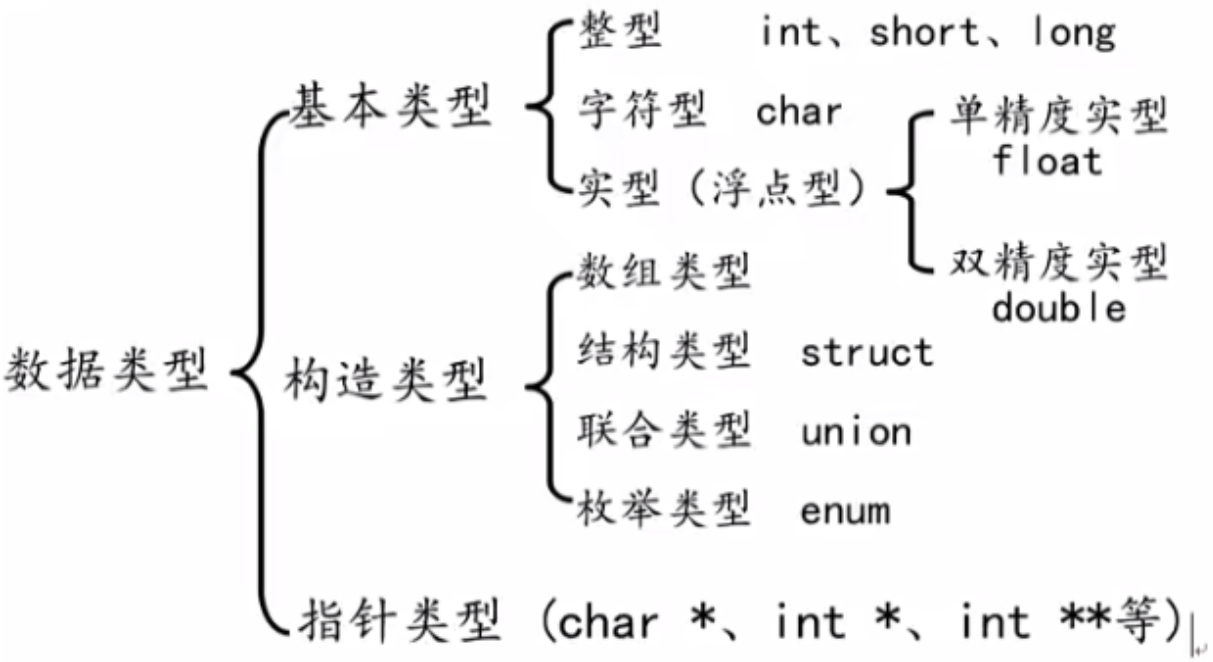
C语言基础Day2

一、变量和常量

- 数据类型：告诉编译器这个数据在内存中需要多大的空间
- 常量：程序中不能改变的量
- 变量：程序运行中可被改变的量，存在于内存中
- 定义变量时，必须以字母和下划线开始，不可以用数字开始
- 变量名不可以取关键字
- define 定义的是常量 不可以被改变的量
- const修改的变量不能被修改



- 定义：在内存中开辟空间
- 初始化：定义时赋值
- 声明：extern 告诉编译器有这个东西，但是这里不开辟空间，没办法赋值

```
extern short a;// 声明变量a 并没有开辟空间
```

```
#include<stdio.h>

int main()
{
    const int a = 10;// const 定义一个常量

    printf("%d\n",a);// 把c的值输出 c = a + b
    return 0;
```

```
}
```

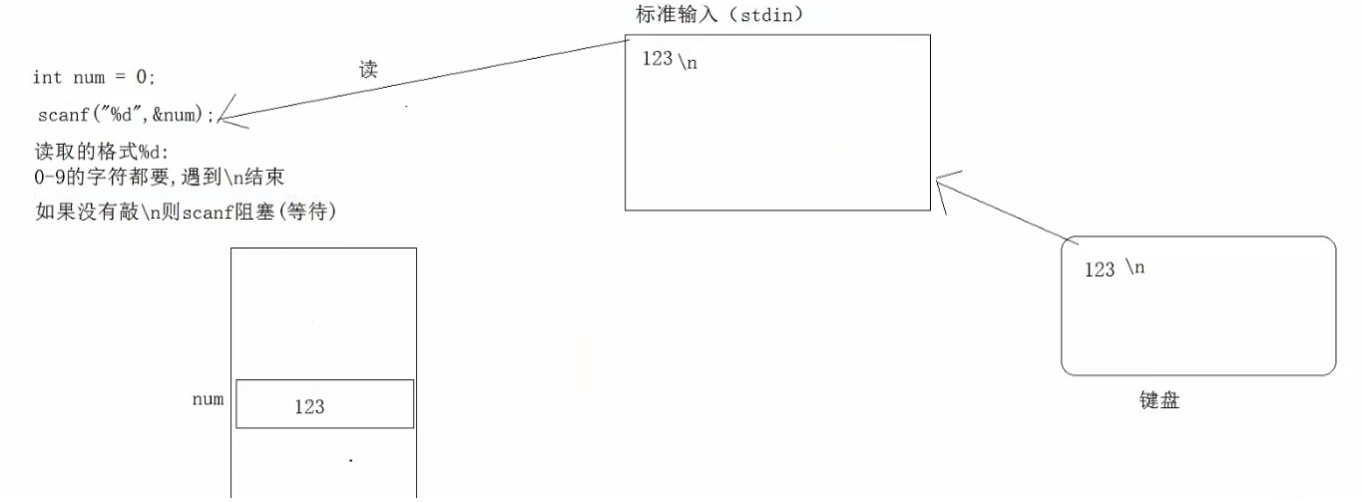
二、int类型

打印格式：

- %d **输出一个有符号的十进制int类型**
- %o 输出八进制int类型
- %x 输出十六进制的int类型，字母小写形式输出
- %X 输出十六进制的int类型，字母以大写形式输出
- %u **输出一个十进制的无符号数**

三、数据的输入scanf

通过scanf语句进行数据的输入，scanf("%d",&a);这里表示使用有符号十进制的形式向a的内存地址中输入一个数。



scanf()函数的读取流程：

- 首先键盘输入数据进行标准输入文件中(stdin)
- 然后scanf函数以有符号十进制数据的形式从标准输入文件中读取数据，通过&num进行取地址

四、short、int、long、long long

- short(短整型) 2个字节
- int（整形）4个字节
- long(长整型) windows为四个字节，Linux为四个字节（32位），八个字节（64位）
- long long (长长整形) 八个字节

注意：

- 整形数据在内存中所占的字节数和所选取的操作系统有关，long类型的整数长度不可以小于int类型，short类型整数的长度不可以大于int类型
- 当一个小的数据类型赋值给一个大的数据类型，不会出错，因为编译器可以自动转化，但是当一个大的类型赋值给一个小的数据类型，就会发生精度丢失

可以使用sizeof测试数据类型的长度

五、整数打印格式

```
sizeof(a) = 2
sizeof(b) = 4
sizeof(c) = 4
sizeof(d) = 8
short a = 10
int b = 10
long c = 10
long long d = 10
unsigned short a2 = 20
unsigned int b2 = 20
unsigned long c2 = 20
unsigned long long d = 20
```

```c
#include<stdio.h>

int main()
{
    short a = 10;
    int b = 10;
    long c = 10l;// 可以写10 因为会自动转换数据类型
    long long d = 10ll;

    printf("sizeof(a) = %u\n",sizeof(a));
    printf("sizeof(b) = %u\n",sizeof(b));
    printf("sizeof(c) = %u\n",sizeof(c));
    printf("sizeof(d) = %u\n",sizeof(d));

    printf("short a = %hd\n",a);
    printf("int b = %d\n",b);
    printf("long c = %ld\n",c);
    printf("long long d = %lld\n",d);

    unsigned short a2 = 20u;
    unsigned int b2 = 20u;
    unsigned long c2 = 20ul;
    unsigned long long d2 = 20ull;

    printf("unsigned short a2 = %hu\n", a2);
    printf("unsigned int b2 = %u\n", b2);
    printf("unsigned long c2 = %lu\n", c2);
    printf("unsigned long long d = %llu\n", d2);


    return 0;
}
```

六、字符型：char

2.4.1 字符变量的定义和输出

字符型变量用于存储一个单一的字符，在C语言中使用char进行表示，其中每一个字符变量都会占用一个字节。在给字符型变量赋值的时候，需要使用一对英文半角格式的单引号将其括起来。**字符变量实际上并不是把该字符本身放到变量的内存单元中去，而是将该字符对应的ASCII编码放到变量的存储单元中去**，char的本质就是一个字节大小的整形。

比如字符'0'存入内存中，存储的是他的ASCII48.

- 字符'0' ascii 值是48
- 字符'1' ascii值是49
- 'A' ascii值是65
- 'B' ascii值是66
- 'a' ascii值是97
- 空字符'' ascii值是32

因为字符所对应的最大ascii值是127，所以用char类型就可以寸的下所有的字符

```
char ch = 97;
printf("%c\n",ch);// 打印ch的值（ascii）所对应的字符a
```

```
char ch = 'A';// 65

printf("%c\n",ch + 32);// 打印ch的值（ascii）所对应的字符
```

```
    char ch = 'a';
    printf("%d\n",sizeof(ch));// 打印1
    printf("%d\n", sizeof('a'));// 打印4  这里看成int类型 c常量97
```

```
    // 将字符型数字转换成int类型的数字
    char ch = '8';
    int a = ch - '0';
    printf("a=%d",a);
```

2.4.2 转义字符

转义字符	含义	ASCII 码值（十进制）
\a	警报	007
\b	退格(BS)，将当前位置移到前一列	008
\f	换页(FF)，将当前位置移到下页开头	012
\n	换行(LF)，将当前位置移到下一行开头	010
\r	回车(CR)，将当前位置移到本行开头	013
\t	水平制表(HT) （跳到下一个 TAB 位置）	009
\v	垂直制表(VT)	011
\\	代表一个反斜线字符"\"	092
\'	代表一个单引号（撇号）字符	039
\"	代表一个双引号字符	034
\?	代表一个问号	063
\0	数字 0	000
\ddd	8 进制转义字符，d 范围 0~7	3 位 8 进制
\xhh	16 进制转义字符，h 范围 0~9，a~f，A~F	3 位 16 进制

七、浮点型

实型变量也可以称为浮点型变量，浮点型变量是用来存储小数数值的。浮点型变量分为两种：单精度浮点数(float)、双精度浮点数（double）,但是double型变量所表示的浮点数比float型变量更加精确。

由于浮点型变量是由有限的存储单元组成的，因此只能提供有限的有效数字，在有效位之外的数字将会被舍掉，将会产生一些误差。

不以f结尾的常量是double类型，以f结尾的常量是float类型

打印时，默认输出六个小数点，同时对于float来说最多只能保证七位有效数字，double类型最多可以保证15位有效数字

```
float a = 2.13121545555573;// 最多保证七位有效数字
double b = 3.144444456346;// 最多保证十五位有效数字
printf("a = %.8f\n",a);// 保留八位小数
printf("b = %.8lf\n",b);// 保留八位小数
```

八、类型限定符

- extern 声明一个变量，extern声明的变量没有建立存储空间。 extern int a;// 变量没有存储空间
- const 定义一个常量，常量的值不可以被修改 const int a = 10;

- volatile 防止编译器优化代码
- register 定义寄存器变量，提高效率。register是建议性的指令，并不是命令型的指令，如果cpu有空闲的寄存器，那么register就生效，如果没有空闲的寄存器，那么register就无效

九、字符串的格式化输出和输入

9.1 字符串常量

- 字符串是内存中一段连续的char空间，以'\0'(ASCII = 数字0)结尾
- 字符串常量是由双引号括起来的字符序列
- 字符串常量和字符常量不同，每一个字符串的结尾，编译器会自动地添加一个结束标志位'\0'
- %s用来打印字符串 printf("%s\n","hello");

注意以下区别：

```
数字0   内存中存的是0
'0'     内存中存的是48
'\0'    内存中存的是0
```

除了使用printf("%c\n",ch); 输出字符ch之外，还有一个格式化输出字符putchar(ch),**没有换行**，

格式化输出：

打印格式	对应数据类型	含义
%d	int	接受整数值并将它表示为有符号的十进制整数。
%hd	short int	短整数。
%hu	unsigned short	无符号短整数。
%o	unsigned int	无符号 8 进制整数。
%u	unsigned int	无符号 10 进制整数。
%x,%X	unsigned int	无符号 16 进制整数，x 对应的是 abcdef，X 对应的是 ABCDEF。
%f	float	单精度浮点数。
%lf	double	双精度浮点数。
%e,%E	double	科学计数法表示的数，此处"e"的大小写代表在输出时用的"e"的大小写。
%c	char	字符型。可以把输入的数字按照 ASCII 码相应转换为对应的字符。
%s	char *	字符串。输出字符串中的字符直至字符串中的空字符（字符串以'\0'结尾，这个'\0'即空字符）。
%p	void *	以 16 进制形式输出指针。
%%	%	输出一个百分号。

9.2 格式化输出的占位符

m.n 格式 m指的是总的位数 n指的是小数部分位数

```
double a = 3.12321;
printf("%10.4lf\n",a);// 总共输出十位  小数部分4位

int b = 10;
printf("%10d\n",b);// 输出十位

printf("%-10d\n",b);// - 左对齐
printf("%010d\n",b);// 不足的地方 使用0进行补全
```

补充:getchar() 函数读取一个字符（从终端）