<http://www.cnblogs.com/tureno/articles/3630689.html>

1．转换说明符  
      %a(%A)     浮点数、十六进制数字和p-(P-)记数法(C99)  
      %c             字符  
      %d             有符号十进制整数  
      %f              浮点数(包括float和doulbe)  
      %e(%E)     浮点数指数输出[e-(E-)记数法]  
      %g(%G)     浮点数不显无意义的零"0"  
      %i              有符号十进制整数(与%d相同)  
      %u             无符号十进制整数  
      %o             八进制整数    e.g.     0123  
      %x(%X)      十六进制整数<?xml:namespace prefix = st1 />()   e.g.   0x1234  
      %p             指针  
      %s             字符串  
      %%            "%"

2．标志  
      左对齐："-"   e.g.   "%-20s"  
      右对齐："+" e.g.   "%+20s"  
      空格：若符号为正，则显示空格，负则显示"-"   e.g.   "% "        
      #：对c,s,d,u类无影响；对o类，在输出时加前缀o；对x类，在输出时加前缀0x；  
           对e,g,f 类当结果有小数时才给出小数点。

3．格式字符串（格式）  
      ［标志］［输出最少宽度］［．精度］［长度］类型   
     "％-md" ：左对齐，若m比实际少时，按实际输出。  
     "%m.ns"：输出m位，取字符串(左起)n位，左补空格，当n>m or m省略时m=n  
                      e.g.    "%7.2s"   输入CHINA  
                                 　           输出"     CH"  
     "%m.nf"：输出浮点数，m为宽度，n为小数点右边数位  
                      e.g.    "%"    输入3852.99  
                                               输出3853.0   
      长度：为ｈ短整形量,ｌ为长整形量

printf的格式控制的完整格式：  
% - .n l或h 格式字符  
下面对组成格式说明的各项加以说明：  
①%：表示格式说明的起始符号，不可缺少。  
②-：有-表示左对齐输出，如省略表示右对齐输出。  
③0：有0表示指定空位填0,如省略表示指定空位不填。  
④m.n：m指域宽，即对应的输出项在输出设备上所占的字符数。N指精度。用于说明输出的实型数的小数位数。为指定n时，隐含的精度为n=6位。  
⑤l或h:l对整型指long型，对实型指double型。h用于将整型的格式字符修正为short型。

－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－  
格式字符   
格式字符用以指定输出项的数据类型和输出格式。  
①d格式：用来输出十进制整数。有以下几种用法：  
%d：按整型数据的实际长度输出。  
%md：m为指定的输出字段的宽度。如果数据的位数小于m，则左端补以空格，若大于m，则按实际位数输出。  
%ld：输出长整型数据。  
②o格式：以无符号八进制形式输出整数。对长整型可以用"%lo"格式输出。同样也可以指定字段宽度用“%mo”格式输出。  
例：  
   main()  
   { int a = -1;  
     printf("%d, %o", a, a);  
   }  
运行结果：-1,177777  
程序解析：-1在内存单元中（以补码形式存放）为(1111111111111111)2，转换为八进制数为(177777)8。  
③x格式：以无符号十六进制形式输出整数。对长整型可以用"%lx"格式输出。同样也可以指定字段宽度用"%mx"格式输出。  
④u格式：以无符号十进制形式输出整数。对长整型可以用"%lu"格式输出。同样也可以指定字段宽度用“%mu”格式输出。  
⑤c格式：输出一个字符。  
⑥s格式：用来输出一个串。有几中用法  
%s：例如:printf("%s", "CHINA")输出"CHINA"字符串（不包括双引号）。  
%ms：输出的字符串占m列，如字符串本身长度大于m，则突破获m的限制,将字符串全部输出。若串长小于m，则左补空格。  
%-ms：如果串长小于m，则在m列范围内，字符串向左靠，右补空格。  
%m.ns：输出占m列，但只取字符串中左端n个字符。这n个字符输出在m列的右侧，左补空格。  
%-m.ns：其中m、n含义同上，n个字符输出在m列范围的左侧，右补空格。如果n>m，则自动取n值，即保证n个字符正常输出。  
⑦f格式：用来输出实数（包括单、双精度），以小数形式输出。有以下几种用法：  
%f：不指定宽度，整数部分全部输出并输出6位小数。  
%m.nf：输出共占m列，其中有n位小数，如数值宽度小于m左端补空格。   
%-m.nf：输出共占n列，其中有n位小数，如数值宽度小于m右端补空格。  
⑧e格式：以指数形式输出实数。可用以下形式：  
%e：数字部分（又称尾数）输出6位小数，指数部分占5位或4位。  
%m.ne和%-m.ne：m、n和”-”字符含义与前相同。此处n指数据的数字部分的小数位数，m表示整个输出数据所占的宽度。  
⑨g格式：自动选f格式或e格式中较短的一种输出，且不输出无意义的零。

－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－  
关于printf函数的进一步说明：  
如果想输出字符"%",则应该在“格式控制”字符串中用连续两个%表示，如:  
printf("%f%%", 1.0/3);  
输出0.333333%。

－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－  
对于单精度数，使用%f格式符输出时，仅前7位是有效数字，小数6位．  
对于双精度数，使用%lf格式符输出时，前16位是有效数字，小数6位．

######################################拾遗 ########################################  
由高手指点  
对于m.n的格式还可以用如下方法表示（例）  
char ch[20];  
printf("%\*.\*s\n",m,n,ch);  
前边的\*定义的是总的宽度，后边的定义的是输出的个数。分别对应外面的参数m和n 。我想这种方法的好处是可以在语句之外对参数m和n赋值，从而控制输出格式。   
--------------------------------------------------------------------------------  
今天()又看到一种输出格式 %n 可以将所输出字符串的长度值赋绐一个变量, 见下例:

int slen;

printf("hello world%n", &slen);

执行后变量被赋值为11

另外：

1、格式输出函数的一般形式

      函数原型： int printf(char \* format[,argument,…]);

      函数功能：按规定格式向输出设备（一般为显示器）输出数据，并返回实际输出的字符数，若出错，则返回负数。

      它使用的一般形式为：

               printf("格式控制字符串",输出项列表);

      语句中“输出项列表”列出要输出的表达式（如常量、变量、运算符表达式、函数返回值等），它可以是0个、一个或多个，每个输出项之间用逗号（,）分隔。输出的数据可以是整数、实数、字符和字符串。

      "格式控制字符串"必须用英文的双引号括起来，它的作用是 控制输出项的格式和输出一些提示信息，例如

                    int i=97; printf("i=%d,%c\n",i,i);

                输出结果为：i=97,a

        语句“printf("i=%d,%c\n",i,i);”中的两个输出项都是变量i，但却以不同的格式输出，一个输出整型数97，另一个输出的却是字符a，其格式分别由“%d”与“%c”来控制。

        语句“printf("i=%d,%c\n",i,i);”的格式控制字符串中“i=”是普通字符，他将照原样输出；“%d”与“%c”是格式控制符；"\n"是转义字符，它的作用是换行。

2、格式控制

      格式控制由格式控制字符串实现。格式控制字符串由3部分组成：普通字符、转义字符、输出项格式说明。

      （1）普通字符。普通字符在输出时，按原样输出，主要用于输出提示信息。

      （2）转义字符。转义字符指明特定的操作，如"\n"表示换行，"\t"表示水平制表等。

      （3）格式说明部分由“%”和“格式字符串”组成，他表示按规定的格式输出数据。

        格式说明的形式为：

 %[flags][width][.prec][F|N|h|I][type]

        各部分说明如下：

1. “[]”表示该项为可选项，即可有可无，如 printf("%d",100);
2. flags为可选择的标志字符，常用的标志字符有：
   * -   ——左对齐输出，默认为右对齐输出；
   * +   ——正数输出加号(+)，负数输出减号(-);
   * 空格 ——正数输出空格代替加号(+)，负数输出减号(-)。
3. width为可选择的宽度指示符。
   * 用十进制正整数表示设置输出值得最少字符个数。不足则补空格，多出则按实际输出，默认按实际输出，例如：
   * printf("%8d\n",100);       printf("%6d\n",100);       printf("%-8d\n",100);        printf("%+8\n",100);
   * 输出结果为：└┘└┘└┘└┘└┘100           
     └┘└┘└┘100          100└┘└┘└┘└┘└┘     └┘└┘└┘└┘+100
4. [.prec]为可选的精度指示符
   * 用“小数点”加“十进制正整数”表示，对“整数”、“实数”和“字符串”的输出有如下功能：对“整数”，表示至少要输出的数字个数，不足补数字0，多则原样输出；对“实数”，表示小数点后至多输出的数字个数，不足则补数字0，多则做舍入处理；对“字符串”，表示最多输出的字符个数，不足补空格，多则丢弃。
   * 例如：printf("%8.2f\n",3.14159);                       
     printf("%8.5f\n",3.14159);
   * 输出结果为：└┘└┘└┘└┘3.14                                  
     └┘3.14159
5. [F|N|h|I]为可选的输出长度修饰符，其功能如下：
   * F ——输出远指针存放的地址；
   * N——输出近指针存放的地址；
   * h——输出短整型数据的值；
   * l——输出长整型或双精度型数据的值。
   * 例如：long  n=40000;
   * printf("%8ld\n",n);/\*因为200\*200是长整型数据\*/
6. type为可选的格式字符，用来进行格式转换。

#include<cstdio>  
int main()  
{  
//for int  
int i=30122121;  
long i2=309095024l;  
short i3=30;  
unsigned i4=2123453;  
printf("%d,%o,%x,%X,%ld,%hd,%u/n",i,i,i,i,i2,i3,i4);//如果是：%l,%h，则输不出结果   
printf("%d,%ld/n",i,i2);//试验不出%ld和%d之间的差别，因为long是4bytes  
printf("%hd,%hd/n/n/n",i,i3);//试验了%hd和%d之间的差别，因为short是2bytes  
//for string and char  
char ch1='d';  
unsigned char ch2=160;  
char \*str="Hello everyone!";  
printf("%c,%u,%s/n/n/n",ch1,ch2,str);//unsigned char超过128的没有字符对应  
  
//for float and double,unsigned and signed can not be used with double and float  
float fl=2.566545445F;//or 2.566545445f  
double dl=265.5651445;  
long double dl2=2.5654441454;  
//%g没有e格式，默认6位包括小数点前面的数,  
//%f没有e格式，默认6位仅只小数点后面包含6位  
//%e采用e格式，默认6位为转化后的小数点后面的6位  
printf("%f,%e,%g,%.7f/n",fl,dl,dl,dl);  
printf("%f,%E,%G,%f/n",fl,dl,dl,dl);//%F is wrong  
printf("%.8f,%.10e/n",fl,dl);  
printf("%.8e,%.10f/n/n/n",fl,dl);  
//for point   
int \*iP=&i;  
char \*iP1=new char;  
void \*iP2;//dangerous!  
printf("%p,%p,%p/n/n/n",iP,iP1,iP2);  
  
//其他知识：负号，表示左对齐（默认是右对齐）；%6.3，6表示宽度，3表示精度  
char \*s="Hello world!";  
printf(":%s: /n:%10s: /n:%.10s: /n:%-10s: /n:%.15s: /n:%-15s: /n:%15.10s: /n:%-15.10s:/n/n/n",  
s,s,s,s,s,s,s,s);  
double ddd=563.908556444;  
printf(":%g: /n:%10g: /n:%.10g: /n:%-10g: /n:%.15g: /n:%-15g: /n:%15.10g: /n:%-15.10g:/n/n/n",  
ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd);  
//还有一个特殊的格式%\*.\* ,这两个星号的值分别由第二个和第三个参数的值指定  
printf("%.\*s /n", 8, "abcdefgggggg");  
printf("%\*.\*f /n", 3,3, 1.25456f);  
  
return 0;  
}