

## 文字列

printf の第1引数や fprintf の第2引数で指定している "と" で囲む表現をしている情報は、**文字列定数**と呼ばれ、メモリー上に文字を並べ、その並べた先頭文字が存在する位置を表現しているデータです。(このように位置を表現しているデータはポインタと呼ばれます。)

この**文字列定数**は、" で始まり、終わりを " で指定するため、内部で直接に " の文字を表現することはできません。" の文字を内部で表現するには、`¥` と " の2つの文字を並べて、1つの " を表現します。このように文字列定数の中では、`¥`は、特別な意味を持ちます。

一つの**改行コード**を表現する場合にも `¥` の文字と `n` の文字を並べて一つの特異文字を表現しました。このように、`¥`と、それに続く文字で表現するスタイルを**エスケープシーケンス**と呼びます。`¥`の**エスケープシーケンス**には、`¥n`以外にも`¥t` で表現するタブや、`¥0` に続く8進数や、`¥x` に続いた2桁の16進数で文字コードを直接指定する方法などがあります。

また、一つの`¥`を文字列定数内に埋め込む場合も、`¥¥`と**2つ並べたエスケープシーケンス**の表現を必要とします。以下に例を示します。

<code>printf( "¥¥と¥"を2つ並べて一つの¥"になります。¥n");</code>	<code>¥</code> と"を2つ並べて一つの"になります。
<code>printf( " 8進数表現¥141,¥142,¥143,¥144¥n");</code>	8進数表現 a,b,c,d
<code>printf( "16進数表現¥x61,¥x62,¥x63,¥x64¥n");</code>	16進数表現 a,b,c,d

実行した時の画面

## printf や fprintf の便利な機能

printf の第1引数や fprintf の第2引数は、出力フォーマットの文字列と呼ばれます。

このフォーマット内では、%の直後で、次に続く引数をどのように出力するか指示しています。そのため、%の文字を出力するには、特別な指示を必要とします。

それは、%を2つ並べます。つまり%%で一つの%を出力します。以下に例を示します。

<code>fprintf(stdout, "%%は、%%を2つ並べて一つを出力させます¥n");</code>	
	出力結果
	<code>%</code> は、 <code>%</code> を2つ並べて一つを出力させます

また、

%における幅指定では、0を直後に書くと、幅に満たない空間を0で埋める指定になります。

<code>printf("値は%010.5fです¥n", 12.345);</code>	値は 0012.34500 です
<code>printf("値は%010.4fです¥n", 12.345);</code>	値は 00012.3450 です
<code>printf("値は%010.3fです¥n", 12.345);</code>	値は 000012.345 です
<code>printf("値は%010.2fです¥n", 12.345);</code>	値は 0000012.35 です
<code>printf("値は%010.1fです¥n", 12.345);</code>	値は 00000012.3 です
<code>printf("値は%010.0fです¥n", 12.345);</code>	値は 0000000012 です

また、出力フォーマットの%指定では、10進数の文字並びに変換して出力する場合に `d` の文字を使いましたが、これ以外にも16進数変換に `X`、8進数変換に `o` (オーの文字) を%の直後に指定します。次にその例を示します。

<pre>printf("%d は\t16 進数:%02X 8 進数%03o\n", 0, 0, 0); printf("%d は\t16 進数:%02X 8 進数%03o\n", 1, 1, 1); printf("%d は\t16 進数:%02X 8 進数%03o\n", 2, 2, 2); printf("%d は\t16 進数:%02X 8 進数%03o\n", 4, 4, 4); printf("%d は\t16 進数:%02X 8 進数%03o\n", 8, 8, 8); printf("%d は\t16 進数:%02X 8 進数%03o\n", 16, 16, 16); printf("%d は\t16 進数:%02X 8 進数%03o\n", 32, 32, 32); printf("%d は\t16 進数:%02X 8 進数%03o\n", 64, 64, 64); printf("%d は\t16 進数:%02X 8 進数%03o\n", 128, 128, 128);</pre>	<pre>0 は      16 進数:00 8 進数 000 1 は      16 進数:01 8 進数 001 2 は      16 進数:02 8 進数 002 4 は      16 進数:04 8 進数 004 8 は      16 進数:08 8 進数 010 16 は     16 進数:10 8 進数 020 32 は     16 進数:20 8 進数 040 64 は     16 進数:40 8 進数 100 128 は    16 進数:80 8 進数 200</pre>
---	---

### 整数定数の表現

整数の値を表現する方法は、10進数で表現する以外にいくつかの方法があります。

それは、10進数、8進数、文字定数で、以下に 109 の値を使った例で示します。

<pre>printf("%d\n", 109); /* 10 進数 */ printf("%d\n", 0x6d); /* 16 進数 */ printf("%d\n", 0155); /* 8 進数 */ printf("%d\n", 'm'); /* 文字定数 */</pre>	<pre>109 109 109 109</pre>
--	----------------------------

上記で表示しているのは、全て 109 のデータである。違いは表現方法で、

10 進数の場合は 0 (ゼロ) から始まらない表現で、

16 進数表現の場合は 0x を前に付け、

8 進数の場合は 0 (ゼロ) から始まる表現にし、

文字定数表現の場合はシングルクォーツで文字を囲んでいる。(109 の ASCII コードは m の文字)

おな、8進や16進で表現する場合は、表現している最上位ビットが1なる表現は、一般に負の値になります。

#### 問題 1

作りなさい。次にこれを、1行表示の繰り返しに変更しなさい。(ファイル名 `tF01.c`)

そして次に、繰り返し範囲を 0 から 65536 まで表示させるように変更しなさい。

(桁の表示が揃うように、%の幅指定は変更のこと。)

このプログラムを

#### 問題 2 次のプログラムを作成して、表示を確認した後で、最後のメッセージ以外の出力を

`F02.htm` のファイルへ出力してファイルを作成するプログラムへ変更しなさい。

```
#include <stdio.h> /* ファイル名 tF02.c */
main()
{
    FILE *fpw;
    int blue = 0;
    int green = 0;
    fpw = stdout;
    fprintf(fpw, "<html><head></head><body>\n");
    while(green <= 0x0fff){
        fprintf(fpw, "<span style=%s\"background-color: #00%02X%02X;%s\">&nbsp;</span>", green, blue);
        blue = blue + 0x10;
        if(blue > 0xff){
            blue = 0;
            green = green + 0x10;
        }
    }
    fprintf(fpw, "\n</body></html>\n");
    fprintf(stdout, "以上!\n");
}
```

正しく実行できて表示できたら、  
`fpw = fopen("F01.htm", "w");` へ変更

正しく表示できたら、  
`fclose(fpw);` を追加する。

**問題 3** 上記を、`http://manabu.quu.cc/up/3/F03.htm` と同じように見える htm の生成プログラムへ変更しなさい。ファイル名 `tF03.c` (余裕があればの問題 4 `F04.htm` と同じように見えるものを作る。)