

## printf の変換指定について <sup>(1)</sup> <sup>(2)</sup>

関数のプロトタイプ宣言 `int printf(const char * restrict format, ...);`

```
例 1 float x = 1.2;    printf("x= %f %n", x);    // x= 1.200000
例 2 double y = 3.14; printf("y= %5.2f %n", y); // y= 3.14
例 3 char *cmd="hoge"; int lineNo=20; char *fmt = "%2$d 行目のコマンド %1$s は不正です";
    printf(fmt, cmd, lineNo); // 20 行目のコマンド hoge は不正です
```

第 1 引数 format での変換指定

`%[引数順][フラグ][最小フィールド幅][.精度][長さ修飾子]変換指定子`

### 変換指定子 f

double 型の引数を丸めて [-]mmm.ddd の形の 10 進表現に変換する。小数点の後の桁数は、精度で指定された値となる。精度が指定されていない場合には 6 として扱われる。精度として明示的に 0 が指定されたときには、小数点以下は表示されない。小数点を表示する際には、小数点の前に少なくとも一桁は数字が表示される <sup>(3)</sup>。

第 2 引数：可変引数！（引数の個数が決まっていない）

規定の実引数拡張

関数のプロトタイプ宣言で引数の型の宣言がない場合や可変引数の場合は、関数の呼び出し時に「規定の実引数拡張」が行われ、float 型は double 型に変換される。従って、printf は実引数の型 float と double を区別しないで double 型と見なす。変換指定子が f の場合は[長さ修飾子]l(long)は本来的に不要である。

```
double x = 12.345;
printf( "%f" , x); //12.345000
x → printf( "%f" , x) → stdout(標準出力 : ディスプレイ) 12.345000 (文字列)
```

#### 既定の実引数拡張

- ・ int よりも順位の低い型の整数引数は、その型の値をすべて int で保持できるならば、int に格上げされる。そうでなければ、unsigned int に格上げされる（「整数の格上げ」）。
- ・ float 型の引数は double に格上げされる。

### 比較 : scanf

```
float x; double y;
scanf( "%f" , &x);    scanf( "%lf" , &y);
```

#### メモリ

stdin (標準入力 : キーボード) 12.345 (文字列) → scanf( "%f" , &x) → x = 12.345000 (4 byte)

stdin (標準入力 : キーボード) 12.345 (文字列) → scanf( "%lf" , &y) → y = 12.345000 (8 byte)

[http://www.k-cube.co.jp/wakaba/server/floating\\_point.html](http://www.k-cube.co.jp/wakaba/server/floating_point.html)

#### 浮動小数点数の内部表現

##### float(32bit)

0 011 1111 1 000 0000 0000 0000 0000

符号(1) 指数部(8)

仮数部(23)

##### double(64bit)

0 011 1111 1111 0000 0000 0000 0000 0000 0000 0000

符号(1) 指数部(11)

仮数部(52)

参考

- (1) <https://ja.wikipedia.org/wiki/Printf>
- (2) <http://www.c-tipsref.com/reference/stdio/printf.html>
- (3) [https://linuxjm.osdn.jp/html/LDP\\_man-pages/man3/printf.3.html](https://linuxjm.osdn.jp/html/LDP_man-pages/man3/printf.3.html)
- (4) <http://chaste.web.fc2.com/index.html>