

文字列のフォーマット

宮田 賢一

表1 書式文字列のパラメータの意味

パラメータ	意味	指定値	設定の意味
fill	パディング文字	任意の文字	パディング幅に足りないときに埋める文字
align	文字の配置方法	"<"	左寄せ
		">"	右寄せ
		"="	符号の後ろをゼロ埋め
		"^"	中央寄せ
sign	符号の付け方	"+"	正の場合 "+", 負の場合 "-"
		"-"	正の場合なし, 負の場合 "-"
		" "	正の場合 " ", 負の場合 "-"
#	typeと連動して別形式で表示	"#"	"b", "o", "x", "X" のとき "0b", "0o", "0x", "0X" を前置 "f", "F" のとき必ず小数点を付加 "g", "G" のとき, 最後の0を除去しない
		0	ゼロパディング
		width	最小幅
,	3桁区切り表示	" , "	文字列を割り当てる最小幅 整数部を3桁で区切ってカンマを挿入
.precision	小数点以下の桁数	整数	小数点以下の桁数
type	表示型	"b"	2進数
		"c"	1文字
		"d"	10進数
		"e", "E"	浮動小数の指数表記
		"f", "F"	浮動小数の固定小数表記
		"g", "G"	指数表記と固定小数表記の併用
		"n"	"d"と同じ
		"o"	8進数
		"s"	文字列
		"x", "X"	16進数
"%"	パーセント表記		

● 2.1 変数値の埋め込み

変数の値を文字列に埋め込みたいとき、文字列中に波括弧で囲まれた置換フィールドを埋め込んでおき、この文字列に対して `format()` 関数を使って置換フィールドを実際の値で置換できます。例えば次のように使います。

```
>>> "{} + {} = {}".format(1, 3, 4)
'1 + 3 = 4'
```

この例では文字列中に3つの置換フィールドが含まれています。これに対して「文字列 `.format()`」という形で呼び出すと、`format` 関数に与えた引数と

同じ順番で文字列中の置換フィールドに引数の値が埋め込まれます。

置換フィールドを {番号} という形で指定すると、`format` 関数の引数の番号を使って任意の順序、かつ同じ引数を複数回使って置換することもできます。

```
>>> "{1} = {0} * {0}".format(9, 81)
'81 = 9 * 9'
```

● 2.2 書式文字列

波括弧で囲われた置換フィールドに対して、{書式文字列} または {番号: 書式文字列} のようにコロン