链滴

golang 之 ` 打印函數 `

作者：269812428

原文链接：https://ld246.com/article/1521804499143

来源网站：链滴

golang fmt格式“占位符”

golang 的fmt 包实现了格式化I/O函数，类似于C的 printf 和 scanf。

定义示例类型和变量
```
type Human struct {
    Name string
}

var people = Human{Name:"zhangsan"}
```

普通占位符

占位符	说明	举例	输出
%v	相应值的默认格式。	Printf("%v", people)	{zhangsan},
%+v	打印结构体时，会添加字段名	Printf("%+v", people)	{Name:zhangsan}
%#v	相应值的Go语法表示	Printf("#v", people)	main.Human{Name:"zhangsan"}
%T	相应值的类型的Go语法表示	Printf("%T", people)	main.Human
%%	字面上的百分号，并非值的占位符	Printf("%%")	%

布尔占位符

占位符	说明	举例	输出
%t	true 或 false。	Printf("%t", true)	true

整数占位符

占位符	说明	举例	输出
%b	二进制表示	Printf("%b", 5)	101
%c	相应Unicode码点所表示的字符	Printf("%c", 0x4E2D)	中
%d	十进制表示	Printf("%d", 0x12)	18
%o	八进制表示	Printf("%d", 10)	12
%q	单引号围绕的字符字面值，由Go语法安全地转义	Printf("%q", 0x4E2D)	'中'
%x	十六进制表示，字母形式为小写 a-f	Printf("%x", 13)	d
%X	十六进制表示，字母形式为大写 A-F	Printf("%x", 13)	D
%U	Unicode格式：U+1234，等同于 "U+%04X"	Printf("%U", 0x4E2D)	U+4E2D

浮点数和复数的组成部分（实部和虚部）

占位符	说明	举例	输出
%b	无小数部分的，指数为二的幂的科学计数法，与 strconv.FormatFloat 的 'b' 转换格式一致。例如 -123456p-78		
%e	科学计数法，例如 -1234.456e+78	Printf("%e", 10.2)	1.020000e+01
%E	科学计数法，例如 -1234.456E+78	Printf("%e", 10.2)	1.020000E+01
%f	有小数点而无指数，例如 123.456	Printf("%f", 10.2)	10.200000
%g	根据情况选择 %e 或 %f 以产生更紧凑的（无末尾的0）输出	Printf("%g", 10.20)	10.2
%G	根据情况选择 %E 或 %f 以产生更紧凑的（无末尾的0）输出	Printf("%G", 10.20+2i)	(10.2+2i)

字符串与字节切片

占位符	说明	举例	输出

%s　　输出字符串表示（string类型或[]byte）　Printf("%s", []byte("Go语言"))　Go语言
%q　　双引号围绕的字符串，由Go语法安全地转义　Printf("%q", "Go语言")　　"Go语言"
%x　　十六进制，小写字母，每字节两个字符　　Printf("%x", "golang")　　676f6c616e67
%X　　十六进制，大写字母，每字节两个字符　　Printf("%X", "golang")　　676F6C616E67

其它标记
占位符　　说明　　举例　　输出
+　　总打印数值的正负号；对于%q（%+q）保证只输出ASCII编码的字符。
　　Printf("%+q", "中文")　"\u4e2d\u6587"
-　　在右侧而非左侧填充空格（左对齐该区域）
#　　备用格式：为八进制添加前导 0（%#o）　　Printf("%#U", '中')　　U+4E2D
　　为十六进制添加前导 0x（%#x）或 0X（%#X），为 %p（%#p）去掉前导 0x；
　　如果可能的话，%q（%#q）会打印原始 （即反引号围绕的）字符串；
　　如果是可打印字符，%U（%#U）会写出该字符的
　　Unicode 编码形式（如字符 x 会被打印成 U+0078 'x'）。
' '　(空格)为数值中省略的正负号留出空白（% d）；
　　以十六进制（% x, % X）打印字符串或切片时，在字节之间用空格隔开
0　　填充前导的0而非空格；对于数字，这会将填充移到正负号之后

golang没有 '%u' 点位符，若整数为无符号类型，默认就会被打印成无符号的。

宽度与精度的控制格式以Unicode码点为单位。宽度为该数值占用区域的最小宽度；精度为小数点之的位数。
操作数的类型为int时，宽度与精度都可用字符 '*' 表示。

对于 %g/%G 而言，精度为所有数字的总数，例如：123.45，%.4g 会打印123.5，（而 %6.2f 会打123.45）。

%e 和 %f 的默认精度为6

对大多数的数值类型而言，宽度为输出的最小字符数，如果必要的话会为已格式化的形式填充空格。

而以字符串类型，精度为输出的最大字符数，如果必要的话会直接截断。

整理自https://studygolang.com/articles/2644