**python format 格式化输出**

2017年03月28日 21:01:53

阅读数：1986

**0.前言**

在python2.7及以上的版本，str.format()的方式为格式化提供了非常大的便利。与之前的%型格式化字符串相比，他显得更为方便与优越。下面我们就来看看format的具体用法。

**1.常见的用法**

二话不说，首先上代码，看看format的一些常用方法。

print "{:.2f}".format(3.1415926) #3.14,保留小数点后两位

print "{:+.2f}".format(3.1415926) #+3.14 带符号保留小数点后两位

print "{:+.2f}".format(-10) #-10.00 带符号保留小数点后两位

print "{:+.0f}".format(-10.00) #-10 不带小数

print "{:0>2d}".format(1) #01 数字补零 (填充左边, 宽度为2)

print "{:x<2d}".format(1) #1x 数字补x (填充右边, 宽度为4)

print "{:x<4d}".format(10) #10xx 数字补x (填充右边, 宽度为4)

print "{:,}".format(1000000) #1,000,000 以逗号分隔的数字格式

print "{:.2%}".format(0.12) #12.00% 百分比格式

print "{:.2e}".format(1000000) #1.00e+06 指数记法

print "{:<10d}".format(10) #10 左对齐 (宽度为10)

print "{:>10d}".format(10) # 10 右对齐 (默认, 宽度为10)

print "{:^10d}".format(10) # 10 中间对齐 (宽度为10)

**1.格式符**

‘f’表示浮点数   
‘d’表示十进制整数. 将数字以10为基数进行输出   
‘%’表示百分数. 将数值乘以100然后以fixed-point(‘f’)格式打印, 值后面会有一个百分号   
‘e’表示幂符号. 用科学计数法打印数字, 用’e’表示幂.

**2.对齐与填充**

^、<、>分别是居中、左对齐、右对齐，后面带宽度   
:后面带填充字符，只能是一个字符，不指定的话默认就是空格。

**2.format基础字符串替换**

format中的字符串参数可以使用{num}来表示。0表示第一个，1表示第二个，以此类推。   
为了更好了解上面的用法，首先我们来看看format的源码

def format(self, \*args, \*\*kwargs): # known special case of str.format

"""

S.format(\*args, \*\*kwargs) -> string

Return a formatted version of S, using substitutions from args and kwargs.

The substitutions are identified by braces ('{' and '}').

"""

pass

给大家翻译一把：   
使用args和kwargs的替换返回S的格式化版本，替换由大括号（’{‘和’}’）标识。

再来看看实际的例子：

print "{0} and {1} is good for big data".format("python","java")

print "{} and {} is good for big data".format("python","java")

print "{1} and {0} and {0} is good for big data".format("python","java")

让代码run起来以后的结果：

python and java is good for big data

python and java is good for big data

java and python and python is good for big data

还可以为参数制定名字：

print "{language1} is as well as {language2}".format(language1="python",language2="java")

* 1

效果如下：

python is as well as java

* 1

**3.通过集合下标的方式访问**

下面的例子也可以达到目的

languages = ["python","java"]

print "{0[0]} is as well as {0[1]}".format(languages)

* 1
* 2

最后的效果：

python is as well as java

* 1

**4.通过对象属性**

format还经常使用在对象属性中。请看下面的例子：

class Person(object):

def \_\_init\_\_(self,name,age):

self.name = name

self.age = age

def \_\_str\_\_(self):

return "name is: {current.name}, age is: {current.age}".format(current=self)

p = Person("leilei",18)

print p

* 1最后的效果：

name is: leilei, age is: 18

（4）其他类型  
主要就是进制了，b、d、o、x分别是二进制、十进制、八进制、十六进制。

[复制代码](javascript:void(0);)

>>> print("{:b}".format(123456))

11110001001000000

>>> print("{:d}".format(123456))

123456

>>> print("{:o}".format(123456))

361100

>>> print("{:x}".format(123456))

1e240

[复制代码](javascript:void(0);)

（5）用“，”号作为金额的千位分隔符的格式化。

>>> print("{:,}".format(123e10))

1,230,000,000,000.0