

python基础_格式化输出（%用法和format用法）

目录

%用法

format用法

%用法

1、整数的输出

%o —— oct 八进制
%d —— dec 十进制
%x —— hex 十六进制

```
1 >>> print('%o' % 20)
2 24
3 >>> print('%d' % 20)
4 20
5 >>> print('%x' % 20)
6 14
```

2、浮点数输出

(1) 格式化输出

%f ——保留小数点后面六位有效数字
%.3f, 保留3位小数位
%e ——保留小数点后面六位有效数字，指数形式输出
%.3e, 保留3位小数位，使用科学计数法
%g ——在保证六位有效数字的前提下，使用小数方式，否则使用科学计数法
%.3g, 保留3位有效数字，使用小数或科学计数法

```
1 >>> print('%f' % 1.11) # 默认保留6位小数
2 1.110000
3 >>> print('%.1f' % 1.11) # 取1位小数
4 1.1
5 >>> print('%e' % 1.11) # 默认6位小数，用科学计数法
6 1.110000e+00
7 >>> print('%.3e' % 1.11) # 取3位小数，用科学计数法
8 1.110e+00
9 >>> print('%g' % 1111.1111) # 默认6位有效数字
10 1111.11
11 >>> print('%.7g' % 1111.1111) # 取7位有效数字
12 1111.111
13 >>> print('%.2g' % 1111.1111) # 取2位有效数字，自动转换为科学计数法
14 1.1e+03
```

(2) 内置round()

公告

昵称: fat39
园龄: 2年2个月
粉丝: 23
关注: 24
+加关注

2019年4月						
日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Python笔记(1)

随笔分类

Python(3)

随笔档案

2019年3月 (7)
2019年2月 (3)
2019年1月 (4)
2018年12月 (8)
2018年11月 (15)
2018年10月 (2)
2018年9月 (14)
2018年8月 (10)
2018年7月 (4)
2018年6月 (17)
2018年5月 (22)
2018年4月 (16)
2018年3月 (16)
2018年2月 (1)
2017年8月 (1)
2017年7月 (2)

round(number[, ndigits])

参数：

number - 这是一个数字表达式。

ndigits - 表示从小数点到最后四舍五入的位数。默认值为0。

返回值

该方法返回x的小数点舍入为n位数后的值。

round()函数只有一个参数，不指定位数的时候，返回一个整数，而且是最靠近的整数，类似于四舍五入，当指定取舍的小数点位数的时候，一般情况也是使用四舍五入的规则，但是碰到.5的情况时，如果要取舍的位数前的小数是奇数，则直接舍弃，如果是偶数则向上取舍。

注：“.5”这个是一个“坑”，且python2和python3出来的接口有时候是不一样的，尽量避免使用round()函数吧

```
1 >>> round(1.1125) # 四舍五入，不指定位数，取整
2 1
3 >>> round(1.1135,3) # 取3位小数，由于3为奇数，则向下“舍”
4 1.113
5 >>> round(1.1125,3) # 取3位小数，由于2为偶数，则向上“入”
6 1.113
7 >>> round(1.5) # 无法理解，查阅一些资料是说python会对数据进行截断，没有深究
8 2
9 >>> round(2.5) # 无法理解
10 2
11 >>> round(1.675,2) # 无法理解
12 1.68
13 >>> round(2.675,2) # 无法理解
14 2.67
15 >>>
```



3、字符串输出

%s

%10s——右对齐，占位符10位

%-10s——左对齐，占位符10位

%.2s——截取2位字符串

%10.2s——10位占位符，截取两位字符串

```
1 >>> print('%s' % 'hello world') # 字符串输出
2 hello world
3 >>> print('%20s' % 'hello world') # 右对齐，取20位，不够则补位
4          hello world
5 >>> print('%-20s' % 'hello world') # 左对齐，取20位，不够则补位
6 hello world
7 >>> print('%.2s' % 'hello world') # 取2位
8 he
9 >>> print('%10.2s' % 'hello world') # 右对齐，取2位
10          he
11 >>> print('%-10.2s' % 'hello world') # 左对齐，取2位
12 he
```



4、其他

(1) 字符串格式代码

阅读排行榜

1. python基础_格式化输出（%用法和format用法）(127393)
2. 目录(305)
3. （转）pycharm 安装 markdown 的三种方法，实测第三种有效(287)
4. 算法_贪心算法找零问题(242)
5. ASCII码(225)

推荐排行榜

1. python基础_格式化输出（%用法和format用法）(13)

符号	说明
%s	字符串
%c	字符
%d	十进制（整数）
%i	整数
%u	无符号整数
%o	八进制整数
%x	十六进制整数
%X	十六进制整数大写
%e	浮点数格式1
%E	浮点数格式2
%f	浮点数格式3
%g	浮点数格式4
%G	浮点数格式5
%%	文字%

(2) 常用转义字符

转义字符	描述
\ (在行尾时)	续行符
\\	反斜杠符号
\'	单引号
\"	双引号
\a	响铃
\b	退格 (Backspace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数yy代表的字符，例如：\o12代表换行
\xyy	十进制数yy代表的字符，例如：\x0a代表换行
\other	其它的字符以普通格式输出

format用法

相对基本格式化输出采用“%”的方法，format()功能更强大，该函数把字符串当成一个模板，通过传入的参数进行格式化，并且使用大括号“{}”作为特殊字符代替“%”

位置匹配

- (1) 不带编号，即“{}”
- (2) 带数字编号，可调换顺序，即“{1}”、“{2}”
- (3) 带关键字，即“{a}”、“{tom}”


1 >>> print('{} {}'.format('hello','world')) # 不带字段
2 hello world
3 >>> print('{0} {1}'.format('hello','world')) # 带数字编号
4 hello world
5 >>> print('{0} {1} {0}'.format('hello','world')) # 打乱顺序
6 hello world hello
7 >>> print('{1} {1} {0}'.format('hello','world'))
8 world world hello
9 >>> print('{a} {tom} {a}'.format(tom='hello',a='world')) # 带关键字
10 world hello world




- ☐ 通过位置匹配
- ☐ 通过名字匹配
- ☐ 通过对象属性匹配
- ☐ 通过下标或key匹配参数

格式转换

- 'b' - 二进制。将数字以2为基数进行输出。
- 'c' - 字符。在打印之前将整数转换成对应的Unicode字符串。
- 'd' - 十进制整数。将数字以10为基数进行输出。
- 'o' - 八进制。将数字以8为基数进行输出。
- 'x' - 十六进制。将数字以16为基数进行输出，9以上的位数用小写字母。
- 'e' - 幂符号。用科学计数法打印数字。用'e'表示幂。
- 'g' - 一般格式。将数值以fixed-point格式输出。当数值特别大的时候，用幂形式打印。
- 'n' - 数字。当值为整数时和'd'相同，值为浮点数时和'g'相同。不同的是它会根据区域设置插入数字分隔符。
- '%' - 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。



```
1 >>> print('{0:b}'.format(3))
2 11
3 >>> print('{:c}'.format(20))
4 □
5 >>> print('{:d}'.format(20))
6 20
7 >>> print('{:o}'.format(20))
8 24
9 >>> print('{:x}'.format(20))
10 14
11 >>> print('{:e}'.format(20))
12 2.000000e+01
13 >>> print('{:g}'.format(20.1))
14 20.1
15 >>> print('{:f}'.format(20))
16 20.000000
17 >>> print('{:n}'.format(20))
18 20
19 >>> print('{:%}'.format(20))
20 2000.000000%
21 >>>
```



进阶用法

进制转换

☐ 2进制、8进制、10进制、16进制

左中右对齐及位数补全

- (1) < (默认) 左对齐、> 右对齐、^ 中间对齐、= (只用于数字) 在小数点后进行补齐
- (2) 取位数"{:4s}"、"{:.2f}"等

☐ 左中右对齐及位数补齐

正负符号显示

☐ 正负符号显示 %+f, %-f, 和 % f的用法

百分数%

☐ 百分数%

时间

☐ 时间

逗号","分隔金钱，没以前进位

☐ 逗号","分隔金钱

占位符嵌套

+


占位符嵌套

占位符%s和%r

+

占位符%s和%r

format的用法变形



```
# a.format(b)
>>> "{0} {1}".format("hello","world")
'hello world'

# f"xxxx"
# 可在字符串前加f以达到格式化的目的，在{}里加入对象，此为format的另一种形式：

>>> a = "hello"
>>> b = "world"
>>> f"{a} {b}"
'hello world'

name = 'jack'
age = 18
sex = 'man'
job = "IT"
salary = 9999.99

print(f'my name is {name.capitalize()}')
print(f'I am {age:^10} years old.')
print(f'I am a {sex}')
print(f'My salary is {salary:10.3f}')

# 结果
my name is Jack.
I am ***18*** years old.
I am a man
My salary is 9999.990
```



参考or转发

官方资料

分类: Python

标签: Python笔记


好文要顶

关注我

收藏该文





fat39

关注 - 24

粉丝 - 23

+加关注

» 下一篇: ASCII码

posted @ 2017-07-13 14:28 fat39 阅读(127403) 评论(0) 编辑 收藏

(评论功能已被禁用)

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

【推荐】腾讯云新客户无门槛领取总价值高达2860元代金券

【活动】2019第四届全球人工智能大会解码“智能+时代”

