

python基础_格式化输出（%用法和format用法）

%用法

format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()

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。

公告

昵称: [Penphy](#)
 园龄: [3年4个月](#)
 粉丝: [7](#)
 关注: [6](#)
[+加关注](#)

<			2021年10月
日	一	二	三
26	27	28	29
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27
31	1	2	3

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

[Java基本数据类型\(4\)](#)
[键盘接收用户输入案例练习\(1\)](#)
[Java8种基本类型分类\(1\)](#)
[dict 基本操作方法-->增 删 改](#)
[list 基本操作方法\(1\)](#)
[str操作方法\(1\)](#)
[format与%用法\(1\)](#)
[python基本数据类型\(1\)](#)
[赋值运算等。。。 \(1\)](#)
[编码种类\(1\)](#)
[更多](#)

随笔分类

[Apollo\(1\)](#)
[JAVA\(80\)](#)
[travel in python\(17\)](#)
[计算机基础\(1\)](#)
[学习java也要翻墙的网站\(1\)](#)

随笔档案

[2020年4月\(5\)](#)
[2020年3月\(10\)](#)
[2019年6月\(2\)](#)
[2019年5月\(20\)](#)
[2019年4月\(15\)](#)
[2019年3月\(9\)](#)
[2019年2月\(3\)](#)
[2019年1月\(21\)](#)
[2018年12月\(12\)](#)
[2018年11月\(3\)](#)
[2018年9月\(1\)](#)
[2018年7月\(2\)](#)
[2018年5月\(10\)](#)

阅读排行榜

1. [python基础_格式化输出（%用法） \(37513\)](#)
 2. [RocketMQ消费者-重复消费](#)
 3. [方法内存分析（进栈（压栈）） \(5140\)](#)

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

```
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
```

3.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
```

4. 对对象中的属性赋值（两种方法）
5. Java实现对zip和rar文件的解

推荐排行榜

1. python基础_格式化输出（9种方法）(2)
2. RocketMQ消费者-重复消费

```
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
```



```
1 >>> '{0}, {1}, {2}'.format('a', 'b', 'c')
2 'a, b, c'
3 >>> '{} {}, {}'.format('a', 'b', 'c') # 3.1+版本支持
4 'a, b, c'
5 >>> '{2}, {1}, {0}'.format('a', 'b', 'c')
6 'c, b, a'
7 >>> '{2}, {1}, {0}'.format(*'abc') # 可打乱顺序
8 'c, b, a'
9 >>> '{0}{1}{0}'.format('abra', 'cad') # 可重复
10 'abracadabra'
```



```
1 'Coordinates: {latitude}, {longitude}'.format(latitude='37.24N', longitude='-115.81W')
2 'Coordinates: 37.24N, -115.81W'
3 >>> coord = {'latitude': '37.24N', 'longitude': '-115.81W'}
4 >>> 'Coordinates: {latitude}, {longitude}'.format(**coord)
5 'Coordinates: 37.24N, -115.81W'
```



```
1 >>> c = 3-5j
2 >>> ('The complex number {0} is formed from the real part {0.real} '
3 ... 'and the imaginary part {0.imag}.').format(c)
4 'The complex number (3-5j) is formed from the real part 3.0 and the imaginary part -5.0.'
5 >>> class Point:
6 ...     def __init__(self, x, y):
7 ...         self.x, self.y = x, y
8 ...     def __str__(self):
9 ...         return 'Point(({self.x}, {self.y}))'.format(self=self)
10 ...
11 >>> str(Point(4, 2))
12 'Point(4, 2)'
```



```
1 >>>
2 >>> coord = (3, 5)
3 >>> 'X: {0[0]}; Y: {0[1]}'.format(coord)
4 'X: 3; Y: 5'
5 >>> a = {'a': 'test_a', 'b': 'test_b'}
6 >>> 'X: {0[a]}; Y: {0[b]}'.format(a)
7 'X: test_a; Y: test_b'
```



format的用法变形



```
1 # a.format(b)
2 >>> "{0} {1}".format("hello","world")
3 'hello world'
4
5
6 # f"xxxx"
7 # 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:
8
9 >>> a = "hello"
10 >>> b = "world"
11 >>> f"{a} {b}"
12 'hello world'
```

```
13
14
15
16 name = 'jack'
17 age = 18
18 sex = 'man'
19 job = "IT"
20 salary = 9999.99
21
22 print(f'my name is {name.capitalize()}')
23 print(f'I am {age:^10} years old.')
24 print(f'I am a {sex}')
25 print(f'My salary is {salary:10.3f}')
26
27 # 结果
28 my name is Jack.
29 I am ****18**** years old.
30 I am a man
31 My salary is 9999.990
```

坎坷困难会让你不断的强大起来 -- 前提是你别怂

分类: [travel in python](#)

标签: [format与%用法](#)

好文要顶

关注我

收藏该文



Penphy

关注 - 6

粉丝 - 7

2

推荐

0

反对

+加关注

« 上一篇: [python spilt\(\)函数的使用方法](#)

» 下一篇: [str 操作方法](#)

posted @ 2018-11-27 19:46 [Penphy](#) 阅读(37513) 评论(0) [编辑](#) [收藏](#) [举报](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】并行超算云面向博客园粉丝推出“免费算力限时申领”特别活动

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载!

【推荐】和开发者在一起: 华为开发者社区, 入驻博客园科技品牌专区



编辑推荐:

- [在 Unity 中渲染一个黑洞](#)
- [理解 ASP.NET Core - 配置\(Configuration\)](#)
- [CSS 奇技淫巧 | 妙用 drop-shadow 实现线条光影效果](#)
- [详细分析 JDK 中 Stream 的实现原理](#)
- [全链路压测: 认识全链路压测](#)

最新新闻:

- [我国研制的“海斗一号”无人潜水器跨入万米科考应用新阶段](#) (2021-10-11 10:11)
- [特斯拉逃离加州解析: “世界第五大经济体”是怎么逼走马斯克的?](#) (2021-10-11 10:07)
- [研究揭示美国首个古老文明遗迹由技术娴熟的工程师们创造](#) (2021-10-11 09:55)
- [NVIDIA为Linux 5.16提供Tegra NVDEC支持](#) (2021-10-11 09:45)
- [联想回应科创板IPO终止: 财务信息可能过期失效](#) (2021-10-11 09:35)

» [更多新闻...](#)