



链滴

字符串

作者: [Harrison](#)

原文链接: <https://ld246.com/article/1673603707041>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1 【单选题】设str1='python', 语句print(str1.center(10,*))的执行结果是_____。

- A、 SyntaxError
- B、 **** python
- C、 ** python**
- D、 python****

考点：字符串的center方法

答案：A

解析：

首先对于*这种特殊符号，没用于特殊用法下，你就把他弄成字符串，非字符的*能表示数字运算的乘法和序列的重复，所以此题报错

然后以下为官方文档原文：

```
str.center(width[, fillchar])
```

返回长度为 *width* 的字符串，原字符串在其正中。使用指定的 *fillchar* 填充两边的空位（默认使用 ASCII 空格符）。如果 *width* 小于等于 `len(s)` 则返回原字符串的副本。

2 【单选题】以下代码的输出结果是_____。
print('{:@>10.6}'.format('Fog'))

- A、 Fog@@@@
- B、 Fog@@@@@@
- C、 @@@@@@Fog
- D、 @@@@Fog

考点: format格式化

答案: C

解析:

详见群文件个人整理《1-2.字符串.py》

此题解释为: 以@填充, 右对齐, 输出字符串宽度为10, 只使用"Fog"的前6个字符 (对此题无意义有意义见下图)

```
[]>>> print("{:@>10.2}".format("Fog"))
@@@@@@@@Fo
[]
```

3 【单选题】以下代码的输出结果是_____。
print('{:0<10.2f}'.format(1234.5678))

- A、 1234.57
- B、 1234.57000
- C、 0001234.57
- D、 1234.567800

考点: format格式化

答案: B

解析:

详见群文件个人整理《1-2.字符串.py》

此题解释为: 以0填充, 左对齐, 输出字符串宽度为10 (注意小数点算一个字符), 保留2位小数 (四五入), 数据结果转化为浮点数 (对此题无意义, 有意义见下图)

```
[]>>> "{:0<10f}".format(20)
'20.00000000'
[]>>> "{:0<10}".format(20)
'2000000000'
[]
```

4 【单选题】以下代码的输出结果是_____。
`print('{0:*^10x},{0:#>10o}'.format(27))`

A、 `****1B****,#####33`

B、 `****1B****,#####27`

C、 `****1b****,#####33`

D、 `****1b****,#####27`

□

考点: format格式化

答案: C

解析:

详见群文件个人整理《1-2.字符串.py》

第一个解释为: 以*填充, 居中对齐, 输出字符串宽度为10, 数据结果转化为十六进制小写

第二个解释为: 以#填充, 右对齐, 输出字符串宽度为10, 数据结果转化为八进制

□

5 【单选题】以下代码的输出结果是_____。
`print('{:\^10b}'.format(10))`

A、 出错

B、 1010

C、 `\\10\\`

D、 `\\1010\\`

□

考点: format格式化

答案: D

解析:

详见群文件个人整理《1-2.字符串.py》

此题解释为: 以\\填充, 居中对齐, 输出字符串宽度为10, 数据结果转化为二进制

□

6 【单选题】以下代码的输出结果是_____。
`print('{:->8.4}'.format('python程序设计'))`

A、 ----pyth

B、 pyth

C、 python程序

D、 python程序设计

□

考点: format格式化

答案: A

解析:

详见群文件个人整理《1-2.字符串.py》

此题解释为: 以-填充, 右对齐, 输出字符串宽度为8, 只使用'python程序设计'中的前4个字符'pyth'

□ 7 【单选题】设str1=' py thon *', 语句`print(str1.strip())`的执行结果是_____。

A、 py thon*

B、 python*

C、 python *

D、 py thon *

□

考点: 字符串的strip方法

答案: D

解析:

`strip`只剔除头尾的字符组合, 没有参数时剔除空格, 即剔除字符串头尾的空格, 以下为官方文档原文
此时可涉及三个相关方法:

□ `str.lstrip([chars])`

返回原字符串的副本, 移除其中的前导字符。 `chars` 参数为指定要移除字符的字符串。如果省略或为 `None`, 则 `chars` 参数默认移除空白符。实际上 `chars` 参数并非指定单个前缀; 而是会移除参数值的所有组合:

```
>>> '  spacious  '.lstrip()
'spacious'
>>> 'www.example.com'.lstrip('cmowz.')
'example.com'
```

□ `str.rstrip([chars])`

返回原字符串的副本, 移除其中的末尾字符。 `chars` 参数为指定要移除字符的字符串。如果省略或为 `None`, 则 `chars` 参数默认移除空白符。实际上 `chars` 参数并非指定单个后缀; 而是会移除参数值的所有组合:

```
>>> '  spacious  '.rstrip()
'spacious'
>>> 'mississippi'.rstrip('ipz')
'mississ'
```

□ `str.strip([chars])`

返回原字符串的副本, 移除其中的前导和末尾字符。 `chars` 参数为指定要移除字符的字符串。如果省略或为 `None`, 则 `chars` 参数默认移除空白符。实际上 `chars` 参数并非指定单个前缀或后缀; 而是会移除参数值的所有组合:

```
>>> '  spacious  '.strip()
'spacious'
>>> 'www.example.com'.strip('cmowz.')
'example'
```

最外侧的前导和末尾 `chars` 参数值将从字符串中移除。开头端的字符的移除将在遇到一个未包含于 `chars` 所指定字符集的字符时停止。类似的操作也将在结尾端发生。例如:

```
>>> comment_string = '#..... Section 3.2.1 Issue #32 .....'
>>> comment_string.strip('.#! ')
'Section 3.2.1 Issue #32'
```

8 【单选题】设s1=["1","2","3","4"], 语句print("-".join(s1))的执行结果是_____。

- A、 1-2-3-4
- B、 ["1234"]
- C、 "1234"
- D、 [1-2-3-4]

□

考点：字符串的join方法

答案：A

解析：

以下为官方文档原文，可以注意到参数必须是序列，且序列的元素必须都是字符串，然后才会通过指字符把序列里的所有字符串元素拼接起来并返回一个新的字符串

□ str. `join(iterable)`

返回一个由 `iterable` 中的字符串拼接而成的字符串。如果 `iterable` 中存在任何非字符串值包括 `bytes` 对象则会引发 `TypeError`。调用该方法的字符串将作为元素之间的分隔。

□

□ 9 【单选题】设s="abcdef", 现要将s字符串中的“c”字符去除，下面_____可以完成该操作。

- A、 `del s[2]`
- B、 `s=s[:2]+s[3:]`
- C、 `s[:3]="ab"`
- D、 `list(s)[3]="ab"`

□

考点：字符串的操作

答案：B

解析：

A: `del`操作为可变序列的操作，会对对象本身进行修改，不适用于作为不可变序列的字符串

B: `s = s[:2] + s[3:] = "ab" + "def" = "abdef"`，实现了题目要求

C: 字符串作为不可变序列，不能进行索引替换和切片替换，这两个操作都会对对象本身进行修改

D: 首该选项凭空创建了一个列表, 无返回值, 没有任何实际影响

10 【单选题】

设str1='pythonpythonpython', 语句print(str1.replace("th","l",2))的执行结果是_____。

- A、 pythonpylonpylon
- B、 pyonpyonpython
- C、 pylonpylonpython
- D、 pylonpylonpylon

考点: 字符串的replace方法

答案: C

解析:

以下为官方文档原文, 此题解释为将字符中前2次出现的"th"都替换为 "l" :

□ `str.replace(old, new[, count])`

返回字符串的副本, 其中出现的所有子字符串 *old* 都将被替换为 *new*。如果给出了可选参数 *count*, 则只替换前 *count* 次出现。