

小白必备内置函数练习

50 看不懂官档的函数说明怎么办？

有些朋友平时反映，看不懂官方文档中介绍函数的说明，比如：

```
max(iterable,*[, key, default])
```

[复制](#)

max 函数的几个形参，为什么有 * 符号，又有 [] ？

函数形参列表中符号 * 表示，后面的形参只能为关键字参数 (keyword argument)，不能为位置参数(positional argument)，也就是说，max 函数要这么用：

```
In [5]: a = [1,2,3,4,2,2,3]

In [6]: max(a,key=lambda x: a.count(x), default=1)
Out[6]: 2
```

[复制](#)

定义函数 f，参数 b 位于 * 后面，只能为关键字参数

```
In [116]: def f(a,*b):
...:     pass

In [117]: f(a,b=1)

In [118]: f(a,1) # 这种调用是错误的
TypeError: f() takes 1 positional argument but 2 were given
```

[复制](#)

再看一个内置函数 sum：

```
sum(iterable, /, start=0)
```

[复制](#)

看到形参列表中有一个 /，它表示 / 前的参数只能是位置参数，不能是关键字参数。

因此，以下调用是合法的：

```
In [18]: a = [1,3,2,1,4,2]

In [19]: sum(a,2) # start=2表示求和的初始值为2
Out[19]: 15
```

[复制](#)

以下调用是非法的，iterable 参数不能被赋值为关键字实参：

```
In [23]: sum(iterable=a,start=2)

TypeError: sum() takes no keyword arguments
```

[复制](#)

平时大家更多看到的是这么使用 max 函数：

```
In [7]: max([1,2,3,4,2,2,3])
Out[7]: 4
```

[复制](#)

[] 表示里面的形参是可选项，max 函数可被如下几种形式调用：

- max(iterable)
- max(iterable,*[, key])
- max(iterable,*[,default])
- max(iterable,*[, key, default])

不能这么被调用：

- max(*[, key])

iterable 没有默认值，所以它是不能被省略的，必须要给出一个实参。

关于 Python 中五类函数参数，也会单独有介绍。

在弄懂这些基础知识后，下面开始总结内置函数的用法，同时学会 Python 中函数如何定义，如何使用等。

51 max 函数使用例子详解解释

max(iterable,*[, *key*, *default*])，返回最大值：

```
In [99]: max(3,1,4,2,1)
Out[99]: 4

In [100]: max((),default=0)
Out[100]: 0

In [89]: di = {'a':3,'b1':1,'c':4}
In [90]: max(di)
Out[90]: 'c'

In [102]: a = [{'name':'xiaoming','age':18,'gender':'male'},{'name':
...: xiaohong','age':20,'gender':'female'}]
In [104]: max(a,key=lambda x: x['age'])
Out[104]: {'name': 'xiaohong', 'age': 20, 'gender': 'female'}
```

[复制](#)

如果已知多个列表，找出列表更长的，使用 max 方法：

```
In [12]: def max_length(*lst):
...:     return max(*lst, key=lambda v: len(v))

In [13]: max_length([1, 2, 3], [4, 5, 6, 7], [8])
Out[13]: [4, 5, 6, 7]
```

[复制](#)

50 看不懂官档的函...

51 max 函数使用例...

52 求次幂再求余的...

53 四舍五入保留三...

54 初始值为10的列...

55 一次求商和余数...

56 Python 的 id 函...

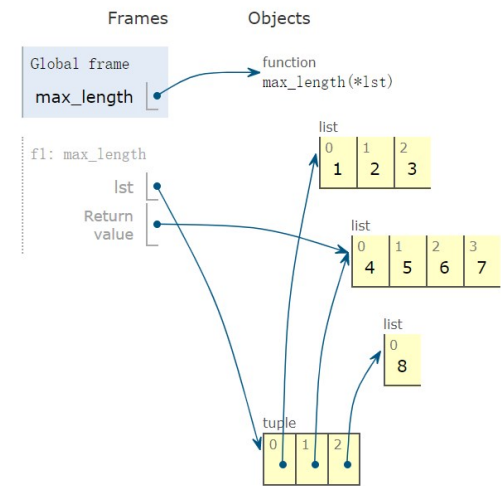
57 列表所有元素是...

58 列表内元素是否...

59 十进制转二进制...

60 如何测试对象是...

代码可视化图：



可看到，`max`、`min` 函数都有一个参数 `key`，它们也被称为 `key` 函数，`key` 函数一般结合更紧凑的 `lambda` 函数。

`max` 有一个 `default` 参数：

- 当传入的列表为空时，若参数 `default` 被赋值，则返回 `default`。
- 否则，会抛空序列的异常(`empty sequence`)

```
In [4]: max([],default='0')
Out[4]: '0'

In [5]: max([])
ValueError: max() arg is an empty sequence
```

52 求次幂再求余的极简实现

`pow(x, y, z=None, /)`

`x`为底的`y`次幂，如果`z`给出，取余

```
In [149]: pow(3, 2, 4)
Out[149]: 1
```

53 四舍五入保留三位小数

`round(number[, ndigits])`

四舍五入，`ndigits` 代表小数点后保留几位：

```
In [157]: round(10.0222222, 3)
Out[157]: 10.022
```

54 初始值为10的列表求和

`sum(iterable, /, start=0)`

求和：

```
In [181]: a = [1,4,2,3,1]

In [182]: sum(a)
Out[182]: 11

In [185]: sum(a,10) #求和的初始值为10
Out[185]: 21
```

55 一次求商和余数

`divmod(a,b)`

分别取商和余数

```
In [97]: divmod(10,3)
Out[97]: (3, 1)
```

56 Python 的 `id` 函数怎么用？

`id(object)`

返回对象的内存地址

```
In [30]: class Student():
...:     def __init__(self,id,name):
...:         self.id = id
...:         self.name = name
...:     def __repr__(self):
...:         return 'id = '+self.id +', name = '+self.name

In [33]: xiaoming = Student('001','xiaoming')
In [115]: id(xiaoming)
Out[115]: 98234208
```

57 列表所有元素是否都为真值

`all(iterable)`

接受一个可迭代对象，如果其所有元素都为真，返回 True，否则返回 False

复制

```
In [2]: all([1,0,3,6])
Out[2]: False

In [3]: all([1,2,3])
Out[3]: True
```

58 列表内元素是否至少有一个为真值

any(iterable)

接受一个可迭代对象，如果可迭代对象里有一个元素为真，返回 True，否则返回 False

复制

```
In [4]: any([0,0,0,[]])
Out[4]: False

In [5]: any([0,0,1])
Out[5]: True
```

59 十进制转二进制、转八进制、转十六进制

bin(x)

将十进制转换为二进制

复制

```
In [35]: bin(10)
Out[35]: '0b1010'
```

oct(x)

将十进制转换为八进制

复制

```
In [36]: oct(9)
Out[36]: '0o11'
```

hex(x)

将十进制转换为十六进制

复制

```
In [37]: hex(15)
Out[37]: '0xf'
```

60 如何测试对象是否为真值

bool(x)

测试一个对象是 True, 还是 False.

复制

```
In [38]: bool([0,0,0])
Out[38]: True

In [39]: bool([])
Out[39]: False

In [40]: bool([1,0,1])
Out[40]: True
```

61 如何将一个字符串转换成字节类型

使用 bytes([source[, encoding[, errors]])

将一个 字符串 转换成 字节 类型

复制

```
In [44]: s = "apple"

In [45]: bytes(s,encoding='utf-8')
Out[45]: b'apple'
```

62 十进制和 ASCII 码如何互转？

chr(i)

查看 十进制 整数对应的 ASCII 字符

复制

```
In [54]: chr(65)
Out[54]: 'A'
```

ord(c)

查看某个 ASCII 字符对应的十进制数

复制

```
In [60]: ord('A')
Out[60]: 65
```

63 怎么转化为 int 或 float ？

int(x)

int(x, base=10), x 可能为字符串或数值，将 x 转换为一个整数。

复制

```
In [16]: int('12')
Out[16]: 12
In [120]: int('12',16)
Out[120]: 18
```

若 x 不能转化为整数，则抛出 ValueError 异常

复制

```
In [3]: int('ws')
ValueError: invalid literal for int() with base 10: 'ws'
```

float(x)

将一个字符串或整数转换为浮点数

```
In [103]: float('30')
Out[103]: 30.0
```

复制

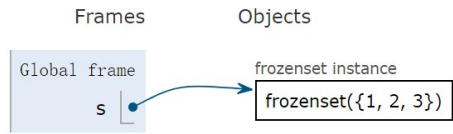
64 如何创建一个不可修改的集合？

frozenset(iterable)

创建一个不可修改的冻结集合，一旦创建不允许增删元素。

```
In [30]: s = frozenset([1,1,3,2,3])
In [31]: s
Out[31]: frozenset({1, 2, 3})
```

复制

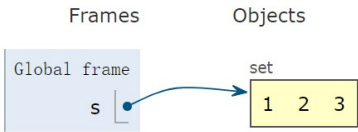


但是，普通集合 `set` 创建后，仍然可以增删元素。

创建一个普通集合 `s`：

```
In [26]: s = {1,2,3}
```

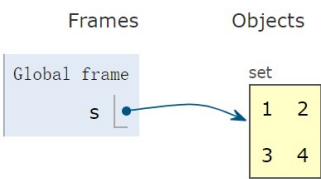
复制



创建 `s` 后，仍然能通过 `add` 方法，再插入元素：

```
In [28]: s.add(4)
In [29]: s
Out[29]: {1, 2, 3, 4}
```

复制

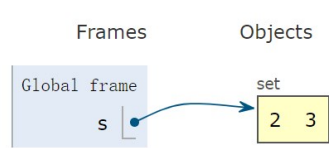


普通集合也能删除元素，使用 `pop` 方法移除集合的第一个元素：

```
In [35]: s = {1,2,3}
In [36]: s.pop()
Out[36]: 1

In [37]: s
Out[37]: {2, 3}
```

复制



65 range 函数的两种调用方法

`range(stop)`；

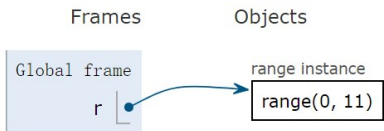
`range(start, stop[, step])`

Python 提供两个内置的 `range` 函数，生成不可变序列：

```
In [153]: range(11) # 只有一个参数，默认初始值为0，步长为1
Out[153]: range(0, 11)

In [154]: range(0,11,1) #三个参数都提供，分别是开始，终止，步长值
Out[154]: range(0, 11)
```

复制

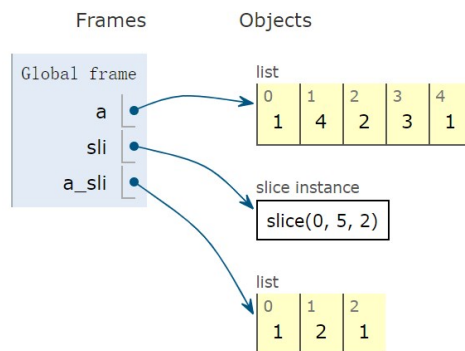


66 slice对象创建和使用

`slice(stop)`；`slice(start, stop[, step])`

返回一个由 `range(start, stop, step)` 所指定索引集的 `slice` 对象

```
In [170]: a = [1,4,2,3,1]
In [171]: a[slice(0,5,2)] #等价于a[0:5:2]
Out[171]: [1, 2, 1]
```



67 zip 函数打包使用总结

zip(*iterables)

创建一个迭代器，聚合每个可迭代对象的元素。

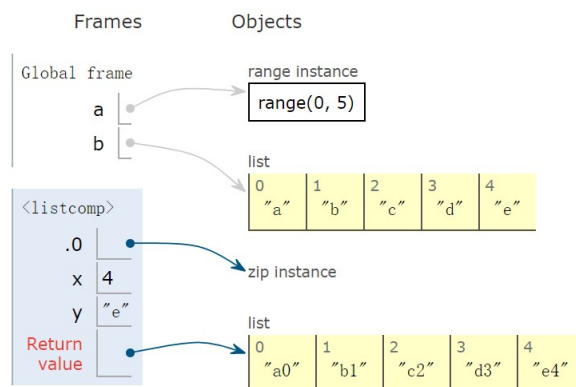
参数前带 *，意味着是可变序列参数，可传入 1 个，2 个或多个参数。

传入 1 个参数

```
In [68]: for i in zip([1,2]):
...:     print(i)
...:
(1,)
(2,)
```

传入 2 个参数：

```
In [191]: a = range(5)
In [192]: b = list('abcde')
In [193]: b
Out[193]: ['a', 'b', 'c', 'd', 'e']
In [194]: [str(y) + str(x) for x,y in zip(a,b)]
Out[194]: ['a0', 'b1', 'c2', 'd3', 'e4']
```



下一章

互动评论



说点什么

评论



The Scrapper

2个月前

day4,习题质量还是不错的

鼓掌



存

评论

<

>