

元组,列表, 别名, 可变性,克隆

(下载幻灯片和 .py 文件并跟随!)

6.0001 第 5 讲

上次

功能

分解 创建结构

抽象 抑制细节

从现在开始将大量使用函数

今天

见过变量类型: int、float、bool、string

引入新的复合数据类型 · 元组 · 列表

别名的概念

可变性的概念

克隆的想法

元组

元素的有序序列,可以混合元素类型不能改变元素值,不可变



remember
strings?

用括号表示

= ()



empty
tuple

t = (2, 与 ,3)

t[0]

计算为 2

(2, mit ,3) + (5,6) 计算结果为 (2, mit ,3,5,6)

t[1:2] 切片元组,计算结果为 (mit ,)

t[1:3] 切片元组,计算结果为 (mit ,3)

len(t) 计算结果为 3

t[1] = 4 给出错误,不能修改对象



extra comma
means a tuple
with one element

元组

方便地用于交换变量值

`x = y`

`y = x`



温度 = x

`x = y`

`y = 温度`



`(x, y) = (y, x)`



用于从一个函数返回多个值

`def quotient_and_remainder(x, y):`

`q = x //` 是



`r = x % y`

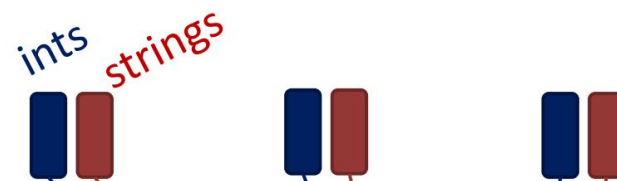
返回 (q, r)

integer
division

`(quot, rem) = quotient_and_remainder(4, 5)`

操作元组

元组:((),(),())



可以遍历元组

```
def get_data(aTuple):
```

数字 = ()

话 = ()

对于 aTuple 中的 t:

数字 = 数字 + (t[0],)

如果 t[1] 不是单词:

单词 = 单词 + (t[1],)

min_n = min(nums)

max_n = max(nums)

unique_words = len (单词)

返回 (min_n, max_n, unique_words)

数 (

)

字 (

)

?

?

?

如果还没有在单词中,即来自 aTuple 的唯一字符串

empty tuple

singleton tuple

列表

有序的信息序列,可通过索引访问


列表用**方括号**表示, []

一个列表包含**元素**

- 通常是齐次的 (即所有整数)
- 可以包含混合类型 (不常见)

列表元素可以更改,因此列表是**可变**的

指数和排序

`a_list = []`  *empty list*

`L = [2, a, 4, [1,2]]`

`len(L)` 计算结果为 4

`L[0]` 计算结果为 2

`L[2]+1` 计算结果为 5

`L[3]` 计算结果为 [1,2], 另一个列表!

`L[4]` 给出错误

`我 = 2`

`L[i-1]` 求值为 `a`, 因为上面的 `L[1]= a`

改变元素

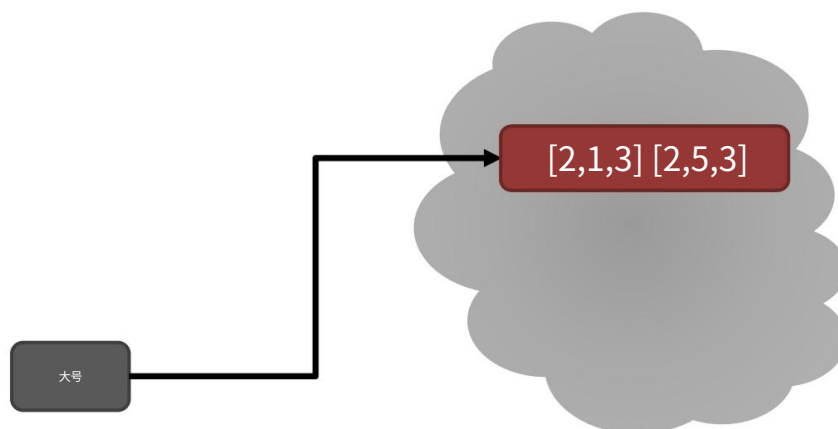
列表是**可变的**！

分配给索引处的元素会改变值

$L = [2, 1, 3]$

$L[1] = 5$

L 现在是 $[2, 5, 3]$, 注意这是**同一个对象**L



遍历列表

计算列表元素的总和

通用模式,遍历列表元素

总计 = 0

对于我在范围内 (len (L)) :

总计 += L[i]

打印总计

总计 = 0

对于 L 中的 i:



总计 += 我

打印总计

like strings,
can iterate
over list
elements
directly

通知

- 列表元素的索引为 0 到 len(L)-1
- range(n) 从 0 到 n-1

列表操作 - 添加

使用 `L.append(element)` 将元素添加到列表末尾

改变列表！

```
L = [2,1,3]
```

```
L.append(5)    L 现在是 [2,1,3,5]
```



点是什么？

- 列表是 Python 对象,Python 中的一切都是对象
- 对象有数据
- 对象具有方法和功能 ·通过

`object_name.do_something()` 访问此信息

- 稍后将详细了解这些内容

列表操作 - 添加

将列表组合在一起使用`连接`， + 运算符,为您提供一个新列表

使用 `L.extend(some_list)``改变`列表

`L1 = [2,1,3]`

`L2 = [4,5,6]`

`L3 = L1 + L2`

`L3` 为 `[2,1,3,4,5,6]`

`L1`、`L2` 不变

`L1.extend([0,6])` 将 `L1` 突变为 `[2,1,3,0,6]`

清单上的操作 - 消除

使用 `del(L[index])` 删除特定索引处的元素

使用 `L.pop()` 删除列表末尾的元素, 返回删除的元素

使用 `L.remove(element)` 删除特定元素 · 查找元素并将其删除

· 如果元素多次出现, 则删除第一次出现 · 如果元素不在列表中, 则给出错误

all these
operations
mutate
the list

`L = [2,1,3,6,3,7,0]` # 按照以下顺序执行 `L.remove(2)` 变异 `L = [1,3,6,3,7,0]`
`L.remove(3)` 改变 `L = [1,6,3,7,0]` `del(L[1])` 改变 `L = [1,3,7,0]`
`L.pop()` 返回 0 并改变 `L = [1,3,7]`

将列表转换为字符串

然后回来

使用列表将字符串转换为列表,返回一个列表,其中包含s 中的每个字符以及 L 中的元素

可以使用 s.split(),在字符参数上拆分字符串,如果在没有参数的情况下调用,则在空格上拆分

使用 .join(L) 将字符列表转为字符串,可以在引号中给一个字符以在每个元素之间添加 char

s = 我<3 cs

列表

s.split(<)

L = [a , b , c] L 是一个列表

.join(L) 返

_ .join(L) 回 “a_b_c”

s 是一个字符串

返回 [| , < , 3 , , c , s]

返回 [| , 3 cs]

返回 “abc”

其他列表操作

排序 ()和排序 ()

反向 ()

还有更多! <https://docs.python.org/3/tutorial/datastructures.html>

L=[9,6,0,3]

排序(L)

返回排序列表,不改变L

L.sort()

变异L=[0,3,6,9]

L.reverse() 变异L=[9,6,3,0]

突变、别名、克隆



重要和

棘手！

再次,Python Tutor 是你最好的朋友来帮助解决这个问题！

<http://www.pythontutor.com/>

内存列表

列表是**可变**的

一个类比

一个人的属性

◦歌手,有钱人

他有很多名字

所有昵称都指向同一个人

·为一个昵称添加新属性……

贾斯汀比伯歌手

有钱的麻烦制造者

·……他所有的昵称都是指旧属性和所有新属性

比布

歌手

有钱的麻烦制造者

JBeebs

歌手

有钱的麻烦制造者

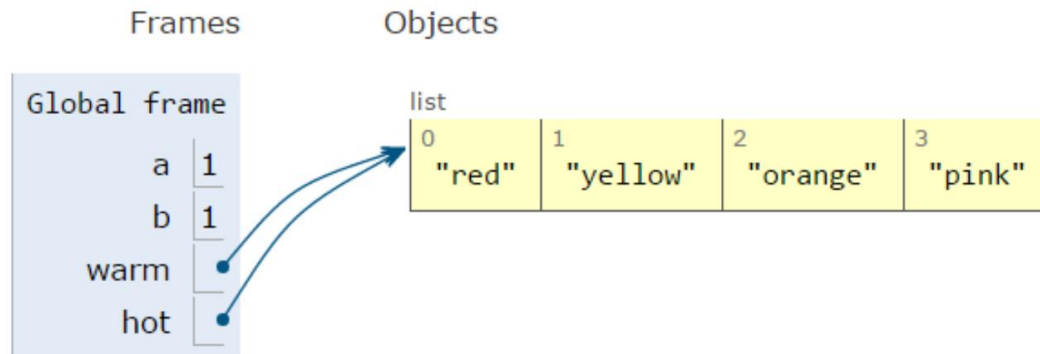
别名

热是温暖的**别名** 改变一个改变另一个！

append() 有副作用

```
1 a = 1
2 b = a
3 print(a)
4 print(b)
5
6 warm = ['red', 'yellow', 'orange']
7 hot = warm
8 hot.append('pink')
9 print(hot)
10 print(warm)
```

```
1
1
['red', 'yellow', 'orange', 'pink']
['red', 'yellow', 'orange', 'pink']
```

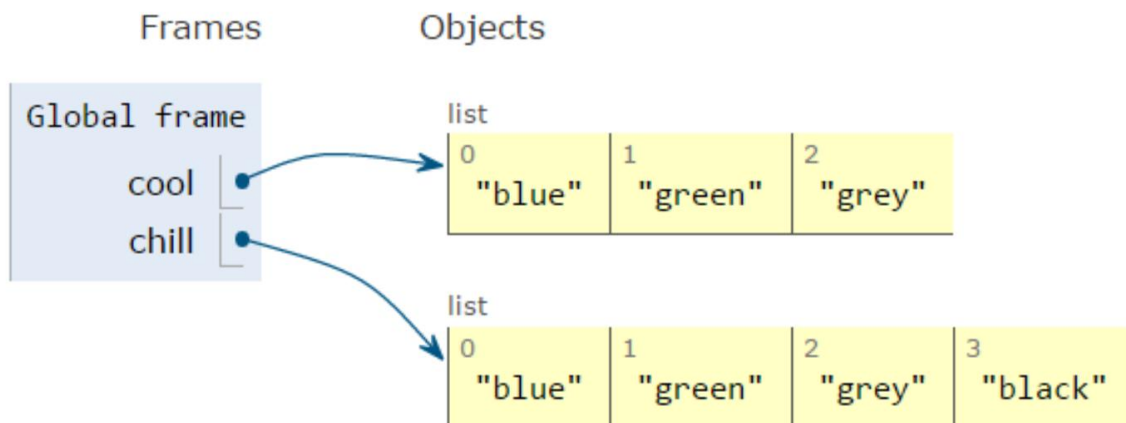


克隆列表

创建一个新列表并使用 `chill = cool[:]` 复制每个元素

```
1 cool = ['blue', 'green', 'grey']
2 chill = cool[:]
3 chill.append('black')
4 print(chill)
5 print(cool)
```

```
['blue', 'green', 'grey', 'black']
['blue', 'green', 'grey']
```



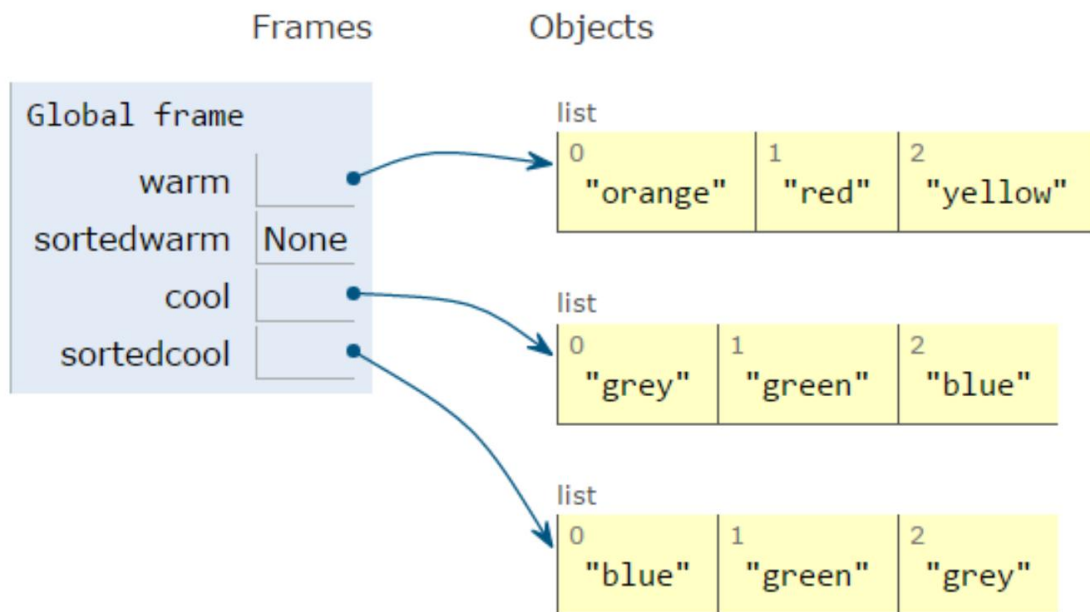
排序列表

调用 `sort()` **改变** 列表, 不返回任何内容

调用 `sorted()` **不会改变**
列表, 必须将结果分配
给变量

```
['orange', 'red', 'yellow']
None
['grey', 'green', 'blue']
['blue', 'green', 'grey']
```

```
1 warm = ['red', 'yellow', 'orange']
2 sortedwarm = warm.sort()
3 print(warm)
4 print(sortedwarm)
5
6 cool = ['grey', 'green', 'blue']
7 sortedcool = sorted(cool)
8 print(cool)
9 print(sortedcool)
```



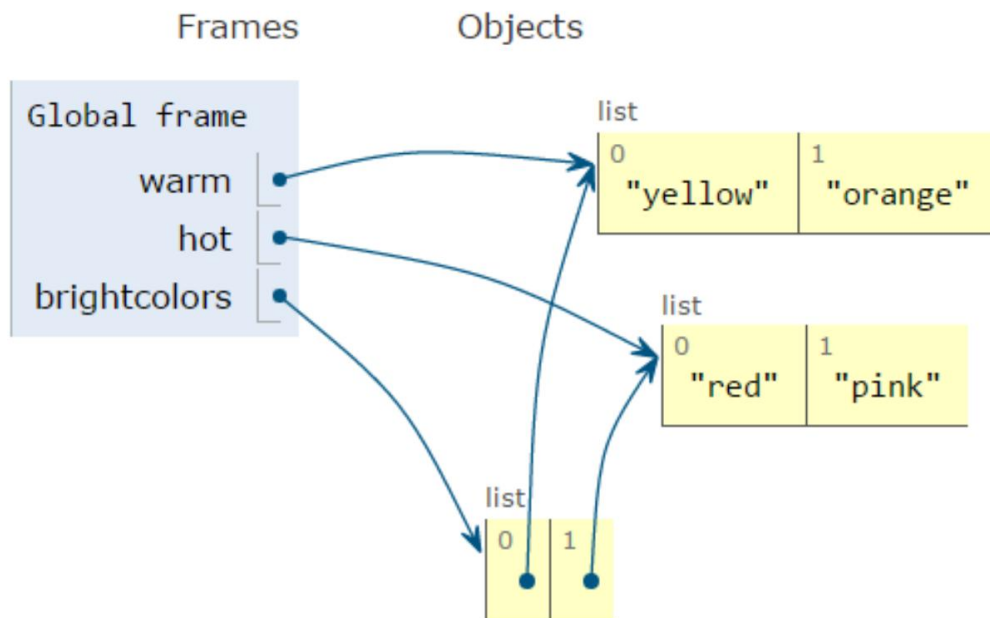
列表的列表...的列表。

可以有嵌套列表

突变后仍有可能产生副作用


```
[['yellow', 'orange'], ['red']]  
['red', 'pink']  
[['yellow', 'orange'], ['red', 'pink']]
```

```
1 warm = ['yellow', 'orange']  
2 hot = ['red']  
3 brightcolors = [warm]  
4 brightcolors.append(hot)  
5 print(brightcolors)  
6 hot.append('pink')  
7 print(hot)  
8 print(brightcolors)
```



变异和迭代在 Python Tutor 中 试试这个！

避免在迭代列表时改变列表

对于 L1 中的 e:
如果 e 在 L2 中:
 L1.remove(e)

```
L1 = [1, 2, 3, 4]
L2 = [1, 2, 5, 6]
remove_dups(L1, L2)
```

L1 是 [2,3,4] 而不是 [3,4] 为什么？

- Python 使用内部计数器来跟踪它在循环中的索引 · mutating 更改列表长度,但 Python 不更新计数器 · 循环永远不会看到元素 2

remove_dups(L1, L2):

L1_copy = L1[:] for e in

L1_copy: 如果 e 在 L2:

L1.remove(e)

*clone list first, note
that L1_copy = L1
does NOT clone*

麻省理工学院开放课件[https://
ocw.mit.edu](https://ocw.mit.edu)

6.0001 计算机科学和 Python 编程简介
2016 年秋季

有关引用这些材料或我们的使用条款的信息,请访问: <https://ocw.mit.edu/terms>。