

Lesson3

Monkey by 51reboot

大纲



- 上节回顾
- 上节进阶
- 经典面试题
- 字典
- 文件操作
- 模块
- 错误和异常
- Q&A

上节回顾



示例1



```
'''  
    变量互换  
'''  
x = 1  
y = 2  
print(x, y)  
  
y, x = x, y  
print(x, y)  
  
'''  
    解包  
'''  
t = (1, 2, 3)  
x1, x2, x3 = t  
print(x1, x2, x3)
```

示例2



- break or exit区别?
- 如果跳出多层循环?

上节进阶



- 字符串
- 列表
- 元组

字符串



- 单引号和双引号使用完全相同
- 使用三引号(`'''` 或 `"""`)可以指定一个多行字符串
- 转义符(反斜杠 `\`)可以用来转义, 使用 `r` 可以让反斜杠不发生转义, 如 `r"this is a line with \n"` , 则 `\n` 会显示, 并不是换行
- 按字面意义级联字符串, 如 `"this " "is " "string"` 会被自动转换为 `this is string`
- 字符串可以用 `+` 运算符连接在一起, 用 `*` 运算符重复
- 字符串有两种索引方式, 从左往右以 0 开始, 从右往左以 -1 开始
- 字符串不能改变
- 没有单独的字符类型, 一个字符就是长度为 1 的字符串
- 字符串的截取的语法格式如下: `变量[头下标:尾下标]`

[string func](<https://segmentfault.com/a/1190000004598007>)

[string format](<https://zhuanlan.zhihu.com/p/37316452>)

列表



列表定义



- 列表是一种无序的、可重复的数据序列，可以随时添加、删除其中的元素。
- 列表页的每个元素都分配一个数字索引，从 0 开始
- 列表使用方括号创建，使用逗号分隔元素
- 列表元素值可以是任意类型，包括变量
- 使用方括号对列表进行元素访问、切片、修改、删除等操作，开闭合区间为 `[]` 形式
- 列表的元素访问可以嵌套
- 方括号内可以是任意表达式

列表方法



- `list.append(obj)`
在列表末尾添加新的对象
- `list.count(obj)`
返回元素在列表中出现的次数
- `list.extend(seq)`
在列表末尾一次性追加另一个序列中的多个值
- `list.index(obj)`
返回查找对象的索引位置，如果没有找到对象则抛出异常
- `list.insert(index, obj)`
将指定对象插入列表的指定位置
- `list.pop([index=-1])`
移除列表中的一个元素（默认最后一个元素），并且返回该元素的值
- `list.remove(obj)`
移除列表中某个值的第一个匹配项
- `list.reverse()`
反向排序列表的元素
- `list.sort(cmp=None, key=None, reverse=False)`
对原列表进行排序，如果指定参数，则使用比较函数指定的比较函数
- `list.clear()`
清空列表
还可以使用 `del list[:]`、`li = []` 等方式实现
- `list.copy()`
复制列表
默认使用等号赋值给另一个变量，实际上是引用列表变量。如果要实现

元组



- 元组与列表类似，不同之处在于元组的元素不能修改
- 元组使用小括号，列表使用方括号
- 元组创建很简单，只需要在括号中添加元素，并使用逗号隔开即可
- 没有 `append()`，`insert()` 这样进行修改的方法，其他方法都与列表一样
- 字典中的键必须是唯一的，同时不可变的，值则没有限制
- 元组中只包含一个元素时，需要在元素后面添加逗号，否则括号会被当作运算符使用

经典面试题



示例1



```
list = [1, 3, 3, 3, 3]
print("Before list {}".format(list))

c = 0
for i in list:
    if i == 3:
        list.remove(i)
        c += 1

print('After list {}'.format(list))
print('delete count of 3: {}'.format(c))
```

示例2



?? True or False ??

```
list01 = [1,2,3]
list02 = list01
list03 = [1,2,3]
```

```
id(list01)
id(list02)
id(list03)
```

```
list01 is list02
list02 is list01
list01 == list02
```

```
list03 is list01
list03 is list02
list03 == list01
list03 == list02
```

- 结论

== 判断的是值是否相等

is 判断的是在内存空间存储的位置是否相同

字典



- 定义
- 操作

定义



- 字典是另一种可变容器模型，可存储任意类型对象
- 字典的每个键值(key=>value)对用冒号(:)分割，每个对之间用逗号(,)分割，整个字典包括在花括号({})中
- 键必须是唯一的，但值则不必
- 值可以是任意数据类型
- 键必须是不可变的，例如：数字、字符串、元组可以，但列表就不行
- 如果用字典里没有的键访问数据，会报错
- 字典的元素没有顺序，不能通过下标引用元素，通过键来引用
- 字典内部存放的顺序和 key 放入的顺序是没有关系的

16

操作

- 定义字典
- 访问字典
- 修改字典
- 删除字典
- 字典函数
- 字典方法
- 字典推导式



示例



- 如下列表，求每种语言出现的次数

- ['python', 'go', 'go', 'java', 'c', 'python', 'ruby', 'python']
- {'python' : 3, 'go' : 2, 'c' : 1, 'java' : 1, 'ruby' : 1}
- 尽情发挥，可以用多种方式实现；

18

文件操作



- Read
- Write

[file link](https://github.com/467754239/python/blob/master/basic/file.md) (<https://github.com/467754239/python/blob/master/basic/file.md>)

示例



- Nginx日志处理

模块



- 标准模块

- os
- sys
- json
- datetime

- 第三方模块

- CSV

语法



```
'''
    方式1:
'''
import sys
import json
import datetime

'''
    方式2:
'''
import sys, json, datetime

'''
    方式3:
'''
from datetime import datetime

import json as pjson
import json as cjson
```

示例



错误和异常



- 语法错误
- 异常

语法错误



SyntaxError 类表示语法错误;
语法错误是无法通过try...except捕获的;

```
>>> print:
      File "<stdin>", line 1
        print:
            ^
SyntaxError: invalid syntax
```

25

异常



运行时发生的错误被称为异常；

```
>>> 1 + '0'
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

26

示例



```
s1 = '1'
print(int(s1))

'''
    异常1
'''

s2 = 'hello'
print(int(s2))

l = [1, 2, 3]
print(l[3])

print(address)

print("End")
```

语法



```
'''Example 1:'''
try:
    pass
except ValueError:
    pass
except IndexError:
    pass
except NameError:
    pass
except:
    pass

'''Example 2:'''
try:
    pass
except (ValueError, IndexError, NameError):
    pass
except Exception as e:
    pass
else:
    pass
finally:
    pass
```

作业



- 用户管理系统V2

- 数据结构：字典
- 分页
- 持久化
- 异常处理
- 导出csv

Q&A



zhengyscn by 51reboot(<https://www.51reboot.com/>)

30

Thank you

zhengyscn@gmail.com (mailto:zhengyscn@gmail.com)

<http://github.com/zhengyscn> (http://github.com/zhengyscn)

