关于在线学习的唠叨

同学问题解答

Python中的变量与对象（名、型、值）

抽象数据类型ADT与Python类定义

慕课作业讲解

数据结构与算法（Python）-06/0310

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

目录

几个课程学习的问题

提问的艺术

学习过程中有问题比没问题好，但不要做伸手党，先努力自己找答案

课程安排相关的，Canvas公告、单元、页面/gis4g/微信群公告；

教学内容相关的，先动手试试，搜索慕课讨论区，搜索微信群；

学习过程中花费的这些时间都不会白费，你会获得知识以外的强大能力。

作业不等于学习

但这些所有事情的前提，是先要看完教学视频

学习是一个过程，作业只是一个一个评价点，会做作业并不代表学会

通过教别人加强学习

多参与慕课讨论区的讨论交流，同学的提问一般体现了课程内容的难点

其实，如果能做到“作业等于学习”挺好的

- 项目制学习
- 游戏化学习
- 数算游戏？

本课中的样例

- **为了聚焦核心算法，课程样例采用简明的约定**

  表达式采用了简明的空格分隔操作数与操作符：A + B * 5

- **在课程样例中并不涉及各种错误的处理**

  特殊的输入格式处理，如：表达式23.45+33.7这样

  由于输入数据引起的错误处理，如：23a+45这样

- **也留给同学们自己完善和扩展的空间**

问题解答

- 1. 作业题第三题感觉逻辑还不是太清楚。
- 2. 助教小哥哥都有没有npy啊
- 3. MOOC上栈的应用：括号匹配

  关于open和close开和闭，可否解释一下

- 4. H2中关于del函数的有关问题
- 5. OJ第三题洗盘子怎么用栈实现啊
- 6. 讲解一下MOOC最后一题（洗盘子）的模拟过程

```
            else:
                top = s.pop()
                if not matches(top,symbol):
                    balanced = False
        index = index + 1
    if balanced and s.isEmpty():
        return True
    else:
        return False

def matches(open,close):
    opens = "([{"
    closers = ")]}"
    return opens.index(open) == closers.index(close)
```

查找符号

问题

- 7. 栈的每个元素中存储的是地址吗？那它可以作为参数传递吗？
- 8.我们自己定义了一个类叫栈，并且定义了对他一些操作的含义，是不是我们**没有定义操作就不能使用**呢？比如栈的本质是用列表实现的，列表可以随意取值比如a=list[2]，栈没有定义这个所以不能这么做。
- 9.栈是不是不可以直接**输出**，对于作业2除了再建一个栈倒一边还有其它的方法可以顺序输出栈里面的值吗？
- 10.希望老师能具体讲一下类，它是什么，怎样才算定义了一个完整的类？

Python中的变量与对象

变量

N9527

对象1

type：华府下人

对象2

type：学警

对象3

type：囚犯

Python中的变量与对象

- **变量：名字Name**
  所有的名字都安排在层次化的命名空间Namespace里
- **名字产生**
  对变量首次赋值、import、def、class
- **名字销毁：del**
- **不存在“有名无实”**
- **必须“名至实归”（名字必须引用对象）**

Python中的变量与对象

对象：值Value/Object

对象的类型：type：class，对象根据其内容和可实施的操作分类

对象的标识：id，对象通过id来标识

所有的对象都存储在堆heap中

对象产生

字面值、表达式计算、类型转换、class调用、函数调用

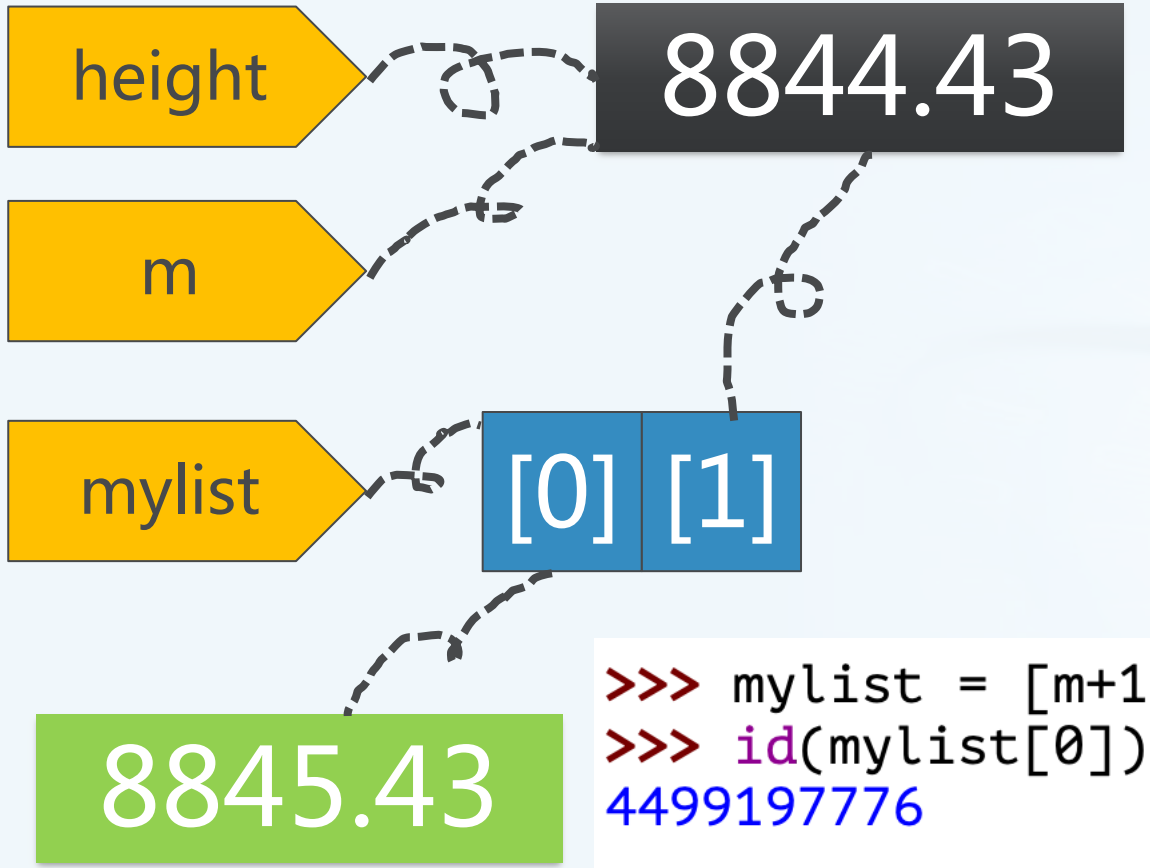
def、lambda、class

对象销毁

del（有名字引用，通过del名字来销毁对象）

垃圾回收（没有任何名字引用的对象）

Python中的变量与对象




```
>>> height = 8844.43
>>> type(height)
<class 'float'>

>>> id(height)
4503840496

>>> m = height
>>> type(m)
<class 'float'>

>>> id(m)
4503840496
```

```
>>> mylist = [m+1, height]
>>> id(mylist[0])
4499197776

>>> id(mylist[1])
4503840496
```

```
>>> type(mylist)
<class 'list'>

>>> id(mylist)
4492984800
```

Python中的变量与对象

- **类型type是对象的分类，包括了：**
  对象包含哪些属性、对象可以做什么操作

  dir函数可以列出类型所包含的属性和操作

```
>>> type(mylist)
<class 'list'>
>>> id(mylist)
4492984800
>>> dir(mylist)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__'
, '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__'
, '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__re
duce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__',
'__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', '
copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
>>> dir(list)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__'
, '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__'
, '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__',
'__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__re
duce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__',
'__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', '
copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

抽象数据类型ADT

- **抽象数据类型是仅定义了接口的数据类型**
  - 只说它操作起来会是什么结果
  - 不说明这些接口具体如何实现
- **栈是个抽象数据类型（ADT Stack）**
- **我们用Python的类定义来实现ADT Stack**
  - 可以有多种实现方法
  - 不同实现方法的复杂度不同
  - 所以不会说ADT Stack的某个接口有时间复杂度

作为子类继承list的一种实现

```
class Stack(list):
    def push(self, item):
        self.append(item)
    def peek(self):
        return self[-1]
    def isEmpty(self):
        return self==[]
    def size(self):
        return len(self)

st = Stack()
```

Python类定义

如何定义一个类？怎么就算是个完整的类了？

class语句定义类

只要有个名字，就算是完整的类啦

```
>>> class MyClass:
        pass

>>> a = MyClass()
>>> type(a)
<class '__main__.MyClass'>
```

Python类定义

怎么输出一个类？

确切的说，是把对象内容以字符串形式展现出来

通过定义__str__和__repr__特殊方法

把对象的什么属性，以什么格式展现，你说了算

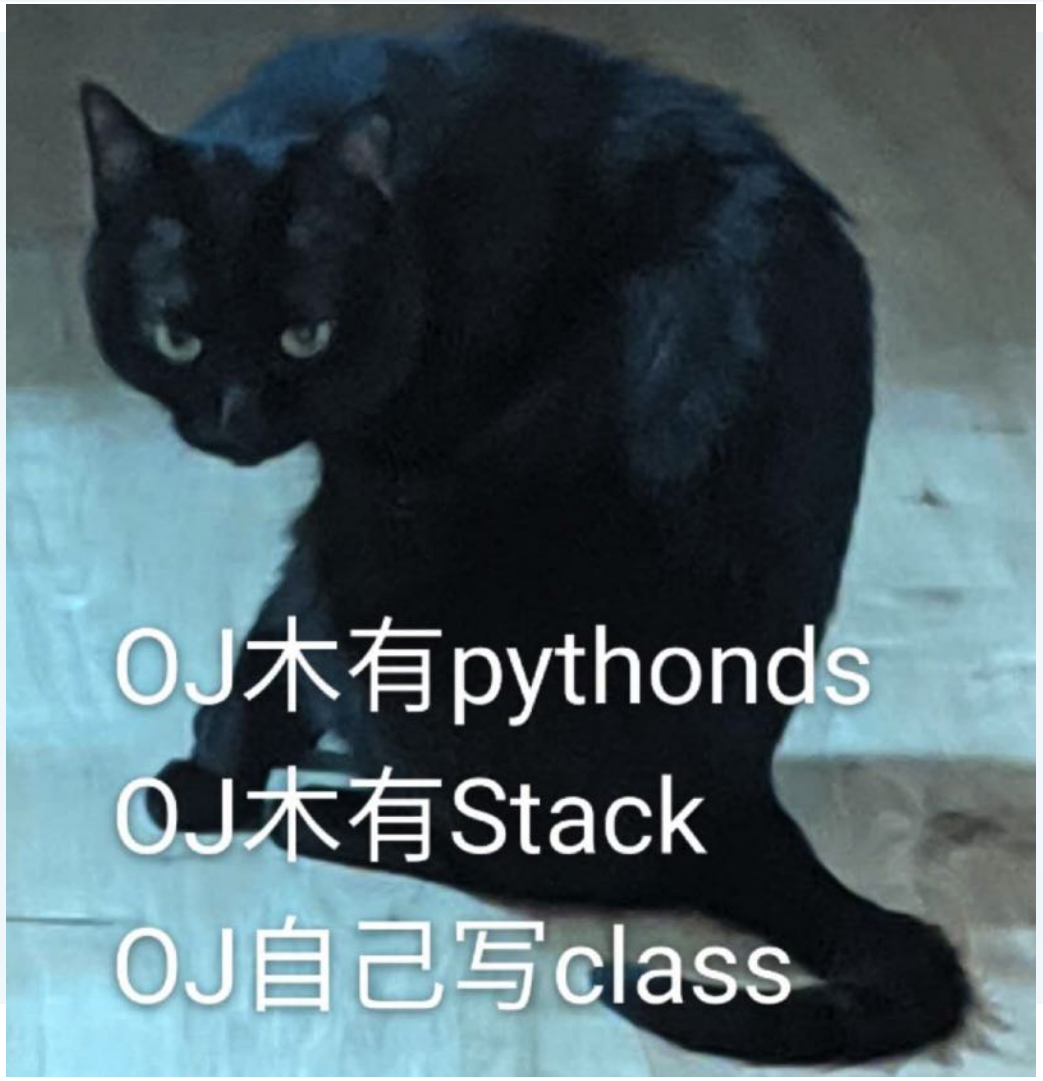
```
1  class Stack(list):
2      def push(self, item):
3          self.append(item)
4      def peek(self):
5          return self[-1]
6      def isEmpty(self):
7          return self==[]
8      def size(self):
9          return len(self)
10     def __repr__(self):
11         l = len(self) * 7
12         s = "|"+ "-" * l + ")\n|"
13         for a in self:
14             s += "| %-5s" % a
15         s += "\n|" + "-" * l + ")"
16         return s
17     __str__ = __repr__
```

```
20 st = Stack()
21 st.push(45)
22 st.push(56)
23 st.push(123)
24 st.push(999)
25 st.push("yes")
26 print(st)
27
```

Shell

```
>>> %Run mystack.py
  |-----------------------------------)
  || 45   | 56   | 123  | 999  | yes
  |-----------------------------------)
>>>
```

慕课OJ作业讲解之前




```python
class Stack(list):
    def push(self, item):
        self.append(item)
    def peek(self):
        return self[-1]
    def isEmpty(self):
        return self==[]
    def size(self):
        return len(self)

st = Stack()
```

极简class Stack
C & P

慕课作业讲解：OJ-W03-1有效的括号


```python
st = Stack()
d = {"(":")", "[":"]", "{":"}"}
s= input()
for a in s:
    if a in "({[":
        st.push(a)
    elif st.isEmpty() or not a == d[st.pop()]: # 右括号多出来，或者不匹配
        print("False")
        break
else:
    if st.isEmpty():
        print("True") # 正好匹配
    else:
        print("False") # 左括号多出来
```

慕课作业讲解：OJ-W03-2一维消消乐


```
st = Stack()
s = input()
for a in s:
    if not st.isEmpty() and a==st.peek(): # 相同的就消
        st.pop()
    else:
        st.push(a)
if st.isEmpty():
    print("None")
else:
    print("".join(st))
```

慕课作业讲解：OJ-W03-3洗碗工


```
st = Stack()
s= input()
n = 0 #正在洗的盘子编号
i = 0 #取盘子的次序，s[i]是取得盘子的编号
while i < 10 and n < 10:
    k = int(s[i])
    # 洗盘子
    # 如果顾客取到了编号k的盘子，那么正在洗的盘子到k之间的所有盘子都洗好叠放
    if n <= k:
        for m in range(n, k+1):
            st.push(m)
            #print("PUSH", m)
        n = k+1 # 正在洗下一个盘子
    # 取盘子
    # 逐个从顶上取盘子，从k开始取，一直取到对不上号，说明要去取的还没洗
    while not st.isEmpty() and st.peek()==int(s[i]):
        m= st.pop()
        #print("POP", m)
        i += 1

# 能取的都取完了，如果盘子堆里还有盘子，说明取的序列不对
if st.isEmpty():
    print("Yes")
else:
    print("No")
```

课堂讨论

- 线性结构的基本特征？
- 栈Stack作为一种特殊的线性结构，有什么进一步的限制？
- 栈Stack主要应用在哪类问题求解？

课堂练习【K03】展示后缀表达式计算过程的栈变化

- 请正确运行后缀表达式求值程序，成功对下列表达式求值，并通过自己定义Stack的__str__和__repr__，来展示计算过程中栈的变化

```
2 3 * 4 +
1 2 + 3 + 4 + 5 +
1 2 3 4 5 * + * +
```

- 请将代码和运行的结果截图做成文档提交。
（pdf，doc，docx，ppt，pptx）