python基础教程_02

woniuppp

上节回顾

- 学编程的方法
- 变量和语句
- 数字运算
- 字符串和格式化
- 获取用户输入
- 注释
- 流程控制 if else(嵌套)
- 逻辑 and or not is
- while和for循环
- break和continue

复习一下上节任务，小练习

- 让用户输入数字，并且累加，输入求和
- 存10000块钱，年利率是3.25%，求多少年之后，存款能翻番
- 遗留问题，求平均分

下面问题来了

我想最后把每次输入的用户名和分数，都打印出来，怎么搞？

- 现有的变量目测满足不了需求
- 我们需要新的数据结构XXX
- XXX可以有顺序的存储一堆数量未知的数据
- 每次输入用户名和分数，我都塞到XXX里存起来，最后把整个XXX打印出来，并且计算平均值
- bingo，目测可以搞定需求

先思考一下XXX需要什么能力才能满足我们

XXX实现的能力就像排队买火车票一样

- 数量可以任意多----排多长的队伍，都OK
- 可遍历
- 有序的----可以通过第X个，找到这个人
- 可以修改里面的值
  - 比如我可以和第3个人换个位置
- 切片，获取其中一部分值
  - 第二个到第十个人，去另外一个地方排
- 插入值，就是允许插队
- 得到元素数量----知道有多少人排队

XXX需要的能力

- 获取索引值----通过一个人的名字，能知道他是第几个
- 统计元素项目
  - 可以统计这个队列里，有多少儿童
- 可以追加元素-----新来一个人，去队尾站着
- 两个list合并
  - 窗口A不卖票了，A队伍和B队伍合并一起排队
- 删除元素----一个人不想排了，直接走了
- 其他

List 隆重出场

定义一个list

```
>>> arr = ['C','python','js','css','html','node']
>>> arr
['C','python','js','css','html','node']
>>> arr[0]
'C'
>>> arr[1]
'python'
>>> arr[-1]
'node'
>>> arr[3]
'css'
```

list函数可以把字符串变成list

所以不要用list当作变量名

```
>>> arr = list('python')
>>> arr
```

可遍历

```
for o in ['wd','pc','me']:
    print o
```

成员是否存在

```
>>>'wd' in ['wd','pc']
>>>
>>>'me' in ['wd','pc']
```

长度 最大值 最小值

```
>>> arr = [,2,3,6,123,345,-2,-8]
>>> len(arr)
>>>
>>> max(arr)
>>>
>>> min(arr)
```

任务：用之前的知识，实现in和这三个功能

del删除list中的元素，

```
>>> arr = ['C','python','js','css','html','node']
>>> del arr[2]
>>> arr
```

list可以相加和做乘法

```
>>> arr = ['wd','pc']
>>> arr*2
>>> arr1 = ['me',1,2,3]
>>> arr + arr1
```

修改值和切片

```
>>> arr = [C','python','js','css','html','node']
>>> arr[0] = 'java'
>>> arr
['java','python','js','css','html','node']
>>> arr[-1] = ruby
>>> arr
['java','python','js','css','html','ruby']
>>> arr[1]
'python'
>>> arr[-1]
'node'
>>> arr[-3]
'css'
```

切片---获取list中一部分连续的数据

```
>>> arr = ['C','python','js','css','html','node']
>>> arr[1:3]
>>>
>>> arr[0:1]
>>>
>>> arr[3:]
>>>
>>> arr[-3:]
>>>
>>> arr[3]
>>>
>>> arr[:]
```

切片可以赋值

```
>>> arr = ['C','python','js','css','html','node']
>>> arr[2:4] = ['pc','wd']
>>> arr
```

切片可以插入和删除元素

```
>>> arr = ['C','python','js','css','html','node']
>>> arr[1:1] = ['pc','wd']
>>> arr
>>> arr[2:3] = []
```

列表方法

- append 追加元素
- count 统计次数
- extend 扩展原列表
- index 获取索引
- insert 插入元素
- pop remove 移出一个元素
- reverse 反向存储

append 向list最后追加元素

没有返回值，修改原数组

```
>>> arr = [1,2,3]
>>> arr.append(4)
>>> arr
```

count 统计list中某个元素出现的次数

count返回值

```
>>> arr = [1,2,3,4,43,3,1,2,3]
>>> arr.count(2)
```

任务:用之前的list知识，实现append和count的功能

extend 扩展原列表

```
>>> a = [1,2,3,4]
>>> b = [5,6,7,8]
>>> a.extend(b)
>>> a
#extend和append的区别
>>> a.append(b)
>>> a
```

extend和直接相加的区别

- extend没有返回值，是修改原数组

任务:用之前的list知识，实现extend的功能

index 从列表中找出某个值，返回第一个匹配项的索引位置

不存在的话，会报错，可以先用in检测

```
>>> arr = [,2,'a',3,5,1,56,45,234,6,7,234]
>>> arr.index('a')
```

insert 插队

arr.insert(位置，插入内容)

```
>>> arr=[,2,3,4,5,6,7]
>>> arr.insert(3,'four')
insert和下面效果一样
```

任务:用之前的list知识，实现index和insert的功能

pop 根据索引移除list中的元素，并且返回，默认删除最后一个

不传参数的话，和append正好相反

```
>>> arr = [,2,3,4]
>>> arr.pop()
>>>
>>> arr
>>> arr1 = [,2,3,4]
>>> arr.pop(2)
>>>
>>> arr
```

arr.append(arr.pop())，arr没有变化~

任务：实现队列和栈

- 先入先出
- 先入后出

remove根据值来删除元素，删除第一个匹配项

如果没有匹配项，报错

```
>>> arr = [a','b','c']
>>> arr.remove'(a')
>>> arr
```

reverse 数组反向 修改列表，不返回值

```
>>> arr = [,2,8,5,4]
>>> arr.reverse()
>>>
>>> arr
```

任务:用之前的list知识，实现remove和reverse的功能

引导复习一下list的知识

- 怎么定义一个list
- in，len，max，min，del
- 切片
- append，count，extend，index，insert，pop，remove，reverse

	方法	原数组　返回值
append	修改	无
count	无	返回
extend	修改	无
index	修改	无
insert	修改	无
pop	修改	返回删除的值
remove	修改	无
reverse	修改	无

实践时间，操练起来

- 对一个list，求出最大的三个值
- 给一个字符串，反向打印出来
- 给定两个数组，判断两个数组里面，是不是有相同的元素
- 用户密码登陆系统，
  - 密码错误三次，锁定用户，不能登陆
- 购物车
  - 用户登陆之后，才能看到商品列表
  - 可以输入商品名，把商品加入购物车
  - 打印购物车列表

Q & A

〈Thank You!〉

Powered By nodePPT v0.9.8-3