



北京大学
PEKING UNIVERSITY

信息科学技术学院

实用Python程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕！

讲义照片均为郭炜拍摄

➤ 教材

Python程序设计基础及实践（慕课版）

郭炜 编著

人民邮电出版社

➤ 慕课:

中国大学MOOC 实用Python程序设计

<https://www.icourse163.org/course/PKU-1460924165?from=searchPage>

高等院校程序设计新形态精品系列

PYTHON

Python Programming Language

Python 程序设计基础及实践

| 慕课版 |

郭炜 • 编著



零基础，不踩坑

内容从零开始。特别指出初学者常犯的大量典型错误，入门编程不踩坑



大广度，重实用

涵盖大部分常用第三方库。围绕数据的获取、分析、处理和展示展开实践，切实解决工作中的问题



配套多，资源全

提供慕课、重点难点视频、讲义、程序源码等丰富配套资源，配有在线评测网站，便于布置、提交、评判作业



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



北京大学
PEKING UNIVERSITY

信息科学技术学院

数据分析相关库

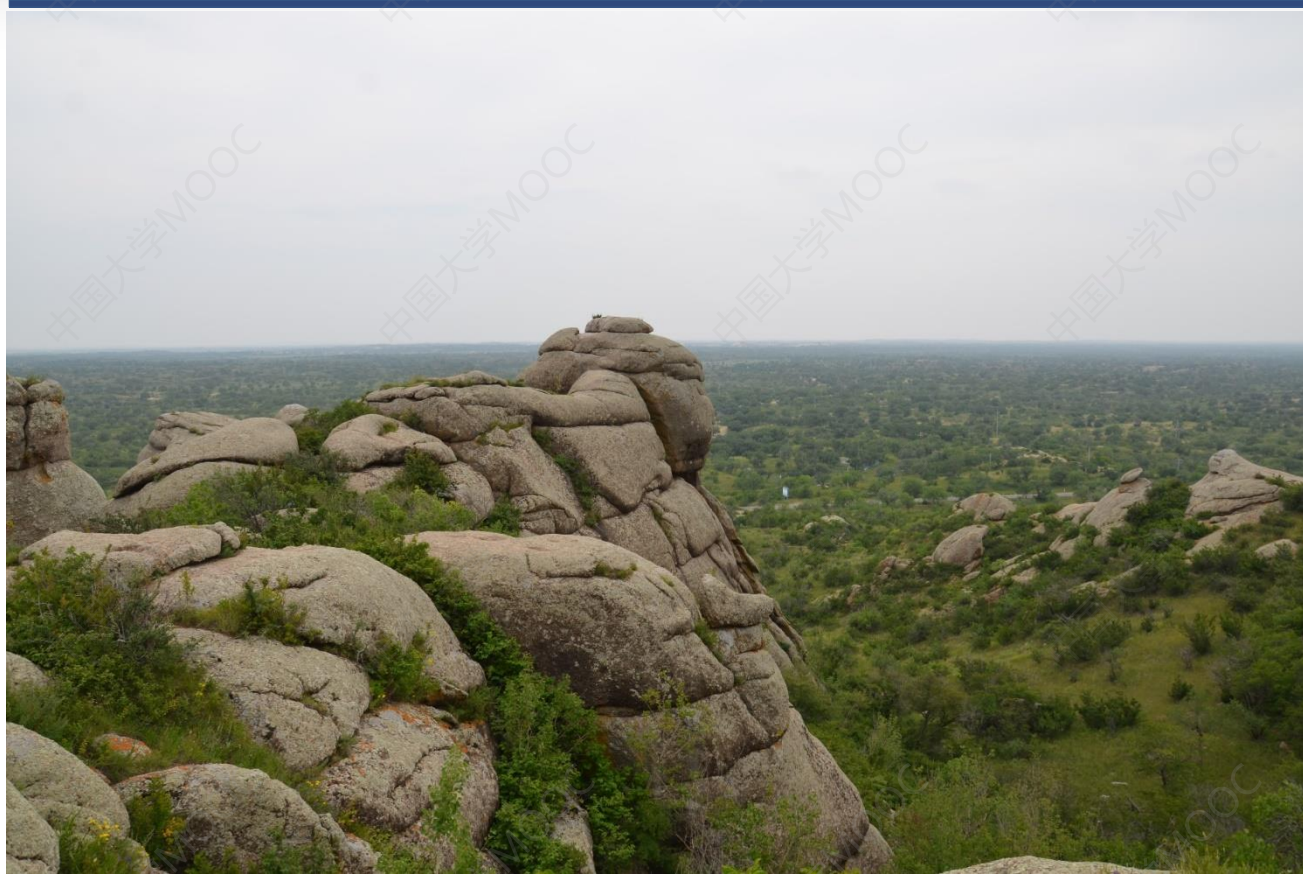
numpy和pandas



北京大学
PEKING UNIVERSITY

信息科学技术学院

多维数组库numpy



内蒙古浑善达克沙地

numpy简介

- 多维数组库，创建多维数组很方便，可以替代多维列表
- 速度比多维列表快
- 支持向量和矩阵的各种数学运算
- 所有元素类型必须相同

pip install numpy 安装

numpy创建数组的函数

| 函数 | 功能 |
|----------------------------------|--|
| <code>array(x)</code> | 根据列表或元组x创建数组 |
| <code>arange(x,y,i)</code> | 创建一维数组，元素等价于 <code>range(x,y,i)</code> |
| <code>linspace(x,y,n)</code> | 创建一个由区间[x,y]的n-1等分点构成的一维数组，包含x和y |
| <code>random.randint(...)</code> | 创建一个元素为随机整数的数组 |
| <code>zeros(n)</code> | 创建一个元素全为0.0的长度为n数组 |
| <code>ones(n)</code> | 创建一个元素全为1.0的长度为n数组 |

numpy创建数组示例

```
import numpy as np          #以后numpy简写为np
print(np.array([1,2,3]))    #>>[1 2 3]
print(np.arange(1,9,2))     #>>[1 3 5 7]
print(np.linspace(1,10,4))  #>>[ 1.  4.  7. 10.]
print(np.random.randint(10,20,[2,3]))
#>>[[12 19 12]
#>> [19 13 10]]
print(np.random.randint(10,20,5)) #>>[12 19 19 10 13]
a = np.zeros(3)
print(a)                    #>>[ 0.  0.  0.]
print(list(a))              #>>[0.0, 0.0, 0.0]
a = np.zeros((2,3),dtype=int) #创建一个2行3列的元素都是整数0的数组
```

numpy数组常用属性和函数

| 属性或函数 | 含义或功能 |
|----------------------|----------|
| dtype | 数组元素的类型 |
| ndim | 数组是几维的 |
| shape | 数组每一维的长度 |
| size | 数组元素个数 |
| argwhere(...) | 查找元素 |
| tolist() | 转换为list |
| min() | 求最小元素 |
| max() | 求最大元素 |
| reshape(...) | 改变数组的形状 |
| flatten() | 转换成一维数组 |

numpy数组常用属性和函数

```
import numpy as np
b = np.array([i for i in range(12)])
#b是[ 0  1  2  3  4  5  6  7  8  9 10 11]
a = b.reshape((3,4))      #转换成3行4列的数组，b不变
print(len(a))             #>>3   a有3行
print(a.size)             #>>12  a的元素个数是12
print(a.ndim)             #>>2   a是2维的
print(a.shape)            #>>(3, 4)   a是3行4列
print(a.dtype)            #>>int32   a的元素类型是32位的整数
L = a.tolist()            #转换成列表，a不变
print(L)
#>>[[0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11]]
b = a.flatten()           #转换成一维数组
print(b)                  #>>[ 0  1  2  3  4  5  6  7  8  9 10 11]
```

numpy数组元素增删

| 函数 | 功能 |
|-------------------------------|---|
| <code>append(x,y)</code> | 若y是数组、列表或元组，就将y的元素添加进数组x得新数组。否则将y本身添加进数组x得新数组 |
| <code>concatenate(...)</code> | 拼接多个数组或列表 |
| <code>delete(...)</code> | 删除数组元素得新数组 |

numpy数组一旦生成，元素就不能增删。上面函数返回一个新的数组。

numpy添加数组元素

```
import numpy as np
a = np.array((1,2,3))
b = np.append(a,10)
print(b)
print(np.append(a,[10,20]))
c = np.zeros((2,3),dtype=int)
print(np.append(a,c))
print(np.concatenate((a,[10,20],a)))
#>>[ 1  2  3 10 20  1  2  3]
print(np.concatenate((c,np.array([[10,20,30]]))))
#c拼接一行[10,20,30]得新数组
print(np.concatenate((c,np.array([[1,2],[10,20]])),axis=1))
#c的第0行拼接了1, 2两个元素、第1行拼接了10,20两个新元素后得到新数组
```

numpy删除数组元素

```
import numpy as np
a = np.array((1,2,3,4))
b = np.delete(a,1)    #删除a中下标为1的元素,a不会改变
print(b)              #>>[1 3 4]
b = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
print(np.delete(b,1,axis=0))    #删除b的第1行得新数组
#>>[[ 1  2  3  4]
#>> [ 9 10 11 12]]
print(np.delete(b,1,axis=1))    #删除b的第1列得新数组
print(np.delete(b,[1,2],axis=0)) #删除b的第1行和第2行得新数组
print(np.delete(b,[1,3],axis=1)) #删除b的第1列和第3列得新数组
```

在numpy数组中查找元素

```
import numpy as np
a = np.array((1,2,3,5,3,4))
pos = np.argwhere(a==3)      #pos是[[2] [4]]
a = np.array([[1,2,3],[4,5,2]])
print(2 in a)                #>>True
pos = np.argwhere(a==2)      #pos是[[0 1] [1 2]]
b = a[a>2]                  #抽取a中大于2的元素形成一个一维数组
print(b)                    #>>[3 4 5]
a[a > 2] = -1               #a变成[[ 1  2 -1] [-1 -1  2]]
```

numpy数组的数学运算

```
import numpy as np
a = np.array((1,2,3,4))
b = a + 1
print(b)           #>>[2 3 4 5]
print(a*b)         #>>[ 2  6 12 20]    a,b对应元素相乘
print(a+b)         #>>[3 5 7 9]    a,b对应元素相加
c = np.sqrt(a*10)   #a*10是[10 20 30 40]
print(c)           #>>[ 3.16227766  4.47213595  5.47722558  6.32455532]
```

numpy数组的切片

numpy数组的切片是“视图”，
是原数组的一部分，而非一部分的拷贝

```
import numpy as np
a = np.arange(8)      #a是[0 1 2 3 4 5 6 7]
b = a[3:6]           #注意，b是a的一部分
print(b)             #>>[3 4 5]
c = np.copy(a[3:6])  #c是a的一部分的拷贝
b[0] = 100           #会修改a
print(a)             #>>[ 0  1  2 100  4  5  6  7]
print(c)             #>>[3 4 5]    c不受b影响
a = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])
b = a[1:3,1:4]       #b是>>[[ 6  7  8] [10 11 12]]
```



北京大学
PEKING UNIVERSITY

信息科学技术学院

数据分析库pandas

DataFrame的构造和访问



锡林郭勒草原

pandas 简介

- 核心功能是在二维表格上做各种操作，如增删、修改、求一列数据的和、方差、中位数、平均数等
- 需要numpy支持
- 如果有openpyxl或xlrd或xlwt支持，还可以读写excel文档。
- 最关键的类：DataFrame,表示二维表格

pip install pandas 安装

pandas的重要类：Series

➤ Series是一维表格，每个元素带标签且有下标，兼具列表和字典的访问形式

```
import pandas as pd
s = pd.Series(data=[80,90,100],index=['语文','数学','英语'])
for x in s:
    print(x,end=" ")
print("")
print(s['语文'],s[1])
print(s[0:2]['数学'])
print(s['数学':'英语'][1])
for i in range(len(s.index)):
    print(s.index[i],end = " ")
s['体育'] = 110
s.pop('数学')
s2 = s.append(pd.Series(120,index = ['政治'])) #不改变s
print(s2['语文'],s2['政治'])
print(list(s2))
```

#>>80 90 100

#>>80 90 标签和序号都可以作为下标来访问元素

#>>90 s[0:2]是切片

#>>100

#>>语文 数学 英语

#在尾部添加元素，标签为'体育'，值为110

#删除标签为'数学'的元素

#>>80 120

#>>[80, 100, 110, 120]

pandas的重要类：Series

```
print(s.sum(),s.min(),s.mean(),s.median())
```

```
#>>290 80 96.66666666666667 100.0
```

输出和、最小值、平均值、中位数

```
print(s.idxmax(),s.argmax()) #>>体育 2
```

输出最大元素的标签和下标

DataFrame的构造和访问

DataFrame是带行列标签的二维表格，的每一列都是一个Series

```
import pandas as pd
pd.set_option('display.unicode.east_asian_width',True)
#输出对齐方面的设置
scores = [['男',108,115,97],['女',115,87,105],['女',100,60,130],
          ['男',112,80,50]]
names = ['刘一哥','王二姐','张三妹','李四弟']
courses = ['性别','语文','数学','英语']
df = pd.DataFrame(data=scores,index = names,columns = courses)
print(df)
```

| | 性别 | 语文 | 数学 | 英语 |
|-----|----|-----|-----|-----|
| 刘一哥 | 男 | 108 | 115 | 97 |
| 王二姐 | 女 | 115 | 87 | 105 |
| 张三妹 | 女 | 100 | 60 | 130 |
| 李四弟 | 男 | 112 | 80 | 50 |

DataFrame的构造和访问

```
print(df.values[0][1],type(df.values)) #>>108 <class 'numpy.ndarray'>
print(list(df.index))                  #>>['刘一哥', '王二姐', '张三妹', '李四弟']
print(list(df.columns))                #>>['性别', '语文', '数学', '英语']
print(df.index[2],df.columns[2])      #>>张三妹 数学
s1 = df['语文']                        #s1是个Series, 代表'语文'那一列
print(s1['刘一哥'],s1[0])              #>>108 108          刘一哥语文成绩
print(df['语文']['刘一哥'])           #>>108              列索引先写
s2 = df.loc['王二姐']                 #s2也是个Series, 代表“王二姐”那一行
print(s2['性别'],s2['语文'],s2[2])
#>>女 115 87      王二姐的性别、语文和数学分数
```

| | 性别 | 语文 | 数学 | 英语 |
|-----|----|-----|-----|-----|
| 刘一哥 | 男 | 108 | 115 | 97 |
| 王二姐 | 女 | 115 | 87 | 105 |
| 张三妹 | 女 | 100 | 60 | 130 |
| 李四弟 | 男 | 112 | 80 | 50 |



北京大学
PEKING UNIVERSITY

信息科学技术学院

数据分析库pandas

DataFrame的切片和统计



云南石林

DataFrame的切片

#DataFrame的切片:

#iloc[行选择器, 列选择器]

#loc[行选择器, 列选择器]

#DataFrame的切片是视图

用下标做切片

用标签做切片

```
df2 = df.iloc[1:3]
```

```
df2 = df.loc['王二姐':'张三妹']
```

```
print(df2)
```

#行切片, 是视图, 选1,2两行

#和上一行等价

| | 性别 | 语文 | 数学 | 英语 |
|-----|----|-----|----|-----|
| 王二姐 | 女 | 115 | 87 | 105 |
| 张三妹 | 女 | 100 | 60 | 130 |

DataFrame的切片

```
df2 = df.iloc[:,0:3]    #列切片(是视图), 选0、1、2三列  
df2 = df.loc[:, '性别': '数学'] #和上一行等价  
print(df2)
```

| | 性别 | 语文 | 数学 |
|-----|----|-----|-----|
| 刘一哥 | 男 | 108 | 115 |
| 王二姐 | 女 | 115 | 87 |
| 张三妹 | 女 | 100 | 60 |
| 李四弟 | 男 | 112 | 80 |

DataFrame的切片

```
df2 = df.iloc[:2, [1,3]]  
df2 = df.loc[:'王二姐', ['语文', '英语']]  
print(df2)
```

#行列切片
#和上一行等价

| | 语文 | 英语 |
|-----|-----|-----|
| 刘一哥 | 108 | 97 |
| 王二姐 | 115 | 105 |

DataFrame的切片

```
df2 = df.iloc[[1,3],2:4]          #取第1、3行，第2、3列
df2 = df.loc[['王二姐','李四弟'],'数学':'英语'] #和上一行等价
print(df2)
```

| | 数学 | 英语 |
|-----|----|-----|
| 王二姐 | 87 | 105 |
| 李四弟 | 80 | 50 |

DataFrame的分析统计

```
print("---下面是DataFrame的分析和统计---")
print(df.T) #df.T是df的转置矩阵,即行列互换的矩阵
print(df.sort_values('语文',ascending=False)) #按语文成绩降序排列
print(df.sum()['语文'],df.mean()['数学'],df.median()['英语'])
#>>435 85.5 101.0 语文分数之和、数学平均分、英语中位数
print(df.min()['语文'],df.max()['数学'])
#>>100 115 语文最低分,数学最高分
print(df.max(axis = 1)['王二姐']) #>>115 王二姐的最高分科目的分数
print(df['语文'].idxmax()) #>>王二姐 语文最高分所在行的标签
print(df['数学'].argmin()) #>>2 数学最低分所在行的行号
print(df.loc[(df['语文'] > 100) & (df['数学'] >= 85)])
```

| | 性别 | 语文 | 数学 | 英语 |
|-----|----|-----|-----|-----|
| 刘一哥 | 男 | 108 | 115 | 97 |
| 王二姐 | 女 | 115 | 87 | 105 |

`sort_values(...inplace=True,axis=1....)`
则原地排序, 将各列排序

DataFrame的修改和增删

```
print("---下面是DataFrame的增删和修改---")
df.loc['王二姐','英语'] = df.iloc[0,1] = 150 #修改王二姐英语和刘一哥语文成绩
df['物理'] = [80,70,90,100] #为所有人添加物理成绩这一列
df.insert(1,"体育",[89,77,76,45]) #为所有人插入体育成绩到第1列
df.loc['李四弟'] = ['男',100,100,100,100,100] #修改李四弟全部信息
df.loc[:, '语文'] = [20,20,20,20] #修改所有人语文成绩
df.loc['钱五叔'] = ['男',100,100,100,100,100] #加一行
df.loc[:, '英语'] += 10 #>>所有人英语加10分
df.columns = ['性别','体育','语文','数学','English','物理'] #改列标签
print(df)
```

| | 性别 | 体育 | 语文 | 数学 | English | 物理 |
|-----|----|-----|-----|-----|---------|-----|
| 刘一哥 | 男 | 89 | 20 | 115 | 107 | 80 |
| 王二姐 | 女 | 77 | 20 | 87 | 160 | 70 |
| 张三妹 | 女 | 76 | 20 | 60 | 140 | 90 |
| 李四弟 | 男 | 100 | 20 | 100 | 110 | 100 |
| 钱五叔 | 男 | 100 | 100 | 100 | 110 | 100 |

初始的df:

| | 性别 | 语文 | 数学 | 英语 |
|-----|----|-----|-----|-----|
| 刘一哥 | 男 | 108 | 115 | 97 |
| 王二姐 | 女 | 115 | 87 | 105 |
| 张三妹 | 女 | 100 | 60 | 130 |
| 李四弟 | 男 | 112 | 80 | 50 |

DataFrame的修改和增删

```
df.drop(['体育','物理'],axis=1, inplace=True) #删除体育和物理成绩
df.drop('王二姐',axis = 0, inplace=True) #删除 王二姐那一行
print(df)
```

| | 性别 | 语文 | 数学 | English |
|-----|----|-----|-----|---------|
| 刘一哥 | 男 | 20 | 115 | 107 |
| 张三妹 | 女 | 20 | 60 | 140 |
| 李四弟 | 男 | 20 | 100 | 110 |
| 钱五叔 | 男 | 100 | 100 | 110 |

```
df.drop([df.index[i] for i in range(1,3)],axis=0,inplace = True)
#删除第1,2行
df.drop([df.columns[i] for i in range(3)],axis = 1,inplace =
True) #删除第0到2列
```



北京大学
PEKING UNIVERSITY

信息科学技术学院

数据分析库pandas

读写excel和csv文档



北京房山红井路

用pandas读excel文档

- 需要openpyxl(对.xlsx文件)或xlrd或xlwt支持(老的.xls文件)
- 读取的每张工作表都是一个DataFrame

| | A | B | C | D | E | F |
|---|------|------|-------------|------------|-------------|---|
| 1 | 产品类别 | 数量 | 销售额 | 成本 | 利润 | |
| 2 | 睡袋 | 4080 | 224,192.97 | 180,501.27 | 43,691.70 | |
| 3 | 彩盒 | 502 | | 62,452.41 | -62,452.41 | |
| 4 | 宠物用品 | 437 | 51,558.43 | | 51,558.43 | |
| 5 | 警告标 | 382 | 36,796.62 | 32,100.23 | 4,696.40 | |
| 6 | 总计 | 5401 | 312548.0199 | 275053.904 | 37494.11589 | |
| 7 | | | | | | |

◀ ▶

销售情况

CVOID | odd | ⊕

⋮

◀ |

用pandas读excel文档

```
import pandas as pd
pd.set_option('display.unicode.east_asian_width', True)
dt = pd.read_excel("excel_sample.xlsx", sheet_name=['销售情况', 1],
                  index_col=0) #读取第0和第1张工作表
df = dt['销售情况']          #dt是字典, df是DataFrame
print(df.iloc[0, 0], df.loc['睡袋', '数量']) #>>4080 4080
print(df)
```

| | 数量 | 销售额 | 成本 | 利润 |
|------|------|---------------|---------------|---------------|
| 产品类别 | | | | |
| 睡袋 | 4080 | 224192.969785 | 180501.266580 | 43691.703206 |
| 彩盒 | 502 | NaN | 62452.410032 | -62452.410032 |
| 宠物用品 | 437 | 51558.425403 | NaN | 51558.425403 |
| 警告标 | 382 | 36796.624662 | 32100.227353 | 4696.397309 |
| 总计 | 5401 | 312548.019850 | 275053.903964 | 37494.115886 |

```
print(pd.isnull(df.loc['彩盒', '销售额'])) #>>True
df.fillna(0, inplace=True)                 #将所有NaN用0替换
print(df.loc['彩盒', '销售额'], df.iloc[2, 2]) #>>0.0 0.0
```


用pandas写excel文档

```
df.to_excel(filename, sheet_name="Sheet1", na_rep='', .....
```

- 将DataFrame对象df中的数据写入excel文档filename中的"Sheet1"工作表, NaN用''代替。
- 会覆盖原有的filename文件
- 如果要在一个excel文档中写入多个工作表, 需要用 `ExcelWriter`

用pandas写excel文档

(接上面程序)

```
writer = pd.ExcelWriter("new.xlsx")
```

#创建ExcelWriter对象

```
df.to_excel(writer, sheet_name="S1")
```

```
df.T.to_excel(writer, sheet_name="S2")
```

#转置矩阵写入

```
df.sort_values('销售额', ascending= False).to_excel(writer,  
                                                    sheet_name="S3")
```

#按销售额排序的新DataFrame写入工作表S3

```
df['销售额'].to_excel(writer, sheet_name="S4")
```

#只写入一列

```
writer.save()
```

用pandas读写csv文件

```
df.to_csv("result.csv", sep=",", na_rep='NA',  
          float_format="%.2f", encoding="gbk")
```

```
df = pd.read_csv("result.csv")
```