

2018-09-24

Python

1. 基础用法

- Python属性的写法

```
1. class MyClass(object):
2.
3.     @property
4.     def myProperty(self):
5.         return self._privateProperty
6.
7.     @myProperty.setter
8.     def myProperty(self, pro):
9.         self._privatePerperty
```

上面会为类MyClass声明一个属性，使用点语法进行存取；

- 读取某一文件夹下的子文件

```
1. def _filenames(self, filedir):
2.     for root, dirs, files in os.walk(filedir):
3.         print "root: {0}".format(root)
4.         print "dirs: {0}".format(dirs)
5.         print "files: {0}".format(files)
```

- mark一个[for]用法

```
1. index_list = [i+1 in i in np.arange(40000)]
2. # 生成1~40000
3. my_list = [index_list[i] for i in np.random.permutation(np.arange(len(index_list)))]
4. # 生成打乱的1~40000
```

2. pandas库

```
import pandas as pd
```

pandas可看作是一个excel的代码操作库；

```
1.  # 读取csv文件
2.  file = pd.read_csv(filePath, header=0)
3.  # filePath为csv文件的路径, header=0是说不加表头
4.  # 读取后可使用下面的方法更改列的名字
5.  file.columns = ['x', 'y', 'z']
6.
7.  # 连接操作
8.  file2 = pd.read_csv(file2Path)
9.  contact = pd.concat([file, file2], axis=1) #连接操作axis=1时是横向连接
10.
11. # 插入操作
12. data = pd.DataFrame(contact)
13. data.columns = ['x', 'y', 'z', 'b']
14. data.insert(3, 'a', 0) # 在第3列后插入名为'a'的列, 值为0
15.
16. # 交换两列(列名同时交换)
17. data.loc[:, ['a', 'b']] = data.loc[:, ['b', 'a']].values
18. # 保存frame到csv文件中
19. data.to_csv('./file.csv', index=False, header=False)
20. # index, header分别代码行名和列名, False代表不保存名称
21.
22.
23. file = pd.read_csv(filePath, header=0)
24. pdata = pd.DataFrame(np.reshape(data, (-1, 6)), columns=['x', 'y', 'z'
25. ])
26. pdata[['x', 'y', 'z']].astype('float64').to_csv('./transnpy.csv', inde
27. x=None, header=None)
# 这里reshape中的-1是一个占位符, 表示该维度上的维数由数据自行算出
# 上面这段将csv取出的数据转换为ndarray并更改shape, 再重新存到csv文件中
```

3. numpy库

```
import numpy as np
```

假如有一个ndarray类型的变量，data；将data中的数列保存为list，在对data进行访问时要比对list进行访问花费时间要大很多，如下：

```
1. print(np.shape(data)) # (10000, 6)
2. x = [data[i][0] for i in range(len(data[:, 0]))]
3.
4. for i in range(len(x)):
5.     value = x[i]
6.
7. for i in range(len(x)):
8.     value = data[i][0]
```

将ndarray数据保存到npv文件中，loading from npv is 30x faster than loading from pickle；

将一个从cvs文件中读出的数组转化为ndarray并保存为npv文件：

```
1. # 保存和读取npv文件
2. np.save('./data.npv', data)
3. npv = np.load('./data.npv')
```