

Numpy基础

1.理解矢量化操作

- python之所以流行，是因为写程序确实方便。执行python程序的时候，程序交给python解释器执行。开发人员并不需要处理内存管理和清理等工作。
- python属于高级语言，书写快，但是执行慢；C语言属于低级语言，书写慢，但是执行快。
- Pandas和numpy能实现程序书写快，执行也快。主要的原因就是使用矢量化操作
- 矢量化利用了一个被称为单指令多数据(SIMD)的处理器特性来更快地处理数据

2. 纽约机场出租数据

- pickup_year: 行程年份
- pickup_month: 行程月份
- pickup_day : 行程日
- pickup_location_code: 接驳地点
- dropoff_location_code : 到达地点
- trip_distance: 行程距离
- trip_length: 行程花费时间
- fare_amount: 行程费用
- total_amount: 总费用

[nyc_taxi.csv下载地址](#)

```
import csv
import numpy as np

# import nyc_taxi.csv as a list of lists
f = open('nyc_taxi.csv', 'r')

# 去除首行
taxi_list = list(csv.reader(f))[1:]

# 将里面的数值转化为浮点型
converted_taxi_list = []
for row in taxi_list:
    converted_row = []
    for each in row:
        converted_row.append(float(each))
    converted_taxi_list.append(converted_row)

# 生成ndarray对象
taxi = np.array(converted_taxi_list)
```

3. 理解Nddarray对象

- ndarray对应'N-dimensional array' --N维数组
- 主要处理1维数组和2维数组。
- Narray里面的每个元素不一定是相同的类型
- **ndarray属性**

```
1 # 定义一个元组, (x, y)
2 # 其中x表示数组有多少行, y表示数组有多少列
3 taxi.shape()
```

- **行切片**

```
1 # 行切片
2 taxi[1:]
```

```
1 taxi_ten = taxi[:10]
2 print(taxi_ten)
```

4. Narray元素选择和切片

- 通用公式

```
1 # 通用公式
2 taxi[row, column]
3 # row 和column可以是单个数字, 也可以是一个切片
4 # 比如row-->3:5, column-->4,6, 表示选择3-5行, 4-6列的数据
```

```
1 # 选择第一行
2 row_0 = taxi[0]
3 # 选择391到500行 (包括500行)
4 rows_391_to_500 = taxi[391:501]
5 # 选择21行5列的元素
6 row_21_column_5 = taxi[21,5]
```