

EDU

CSDN学院 IT实战派

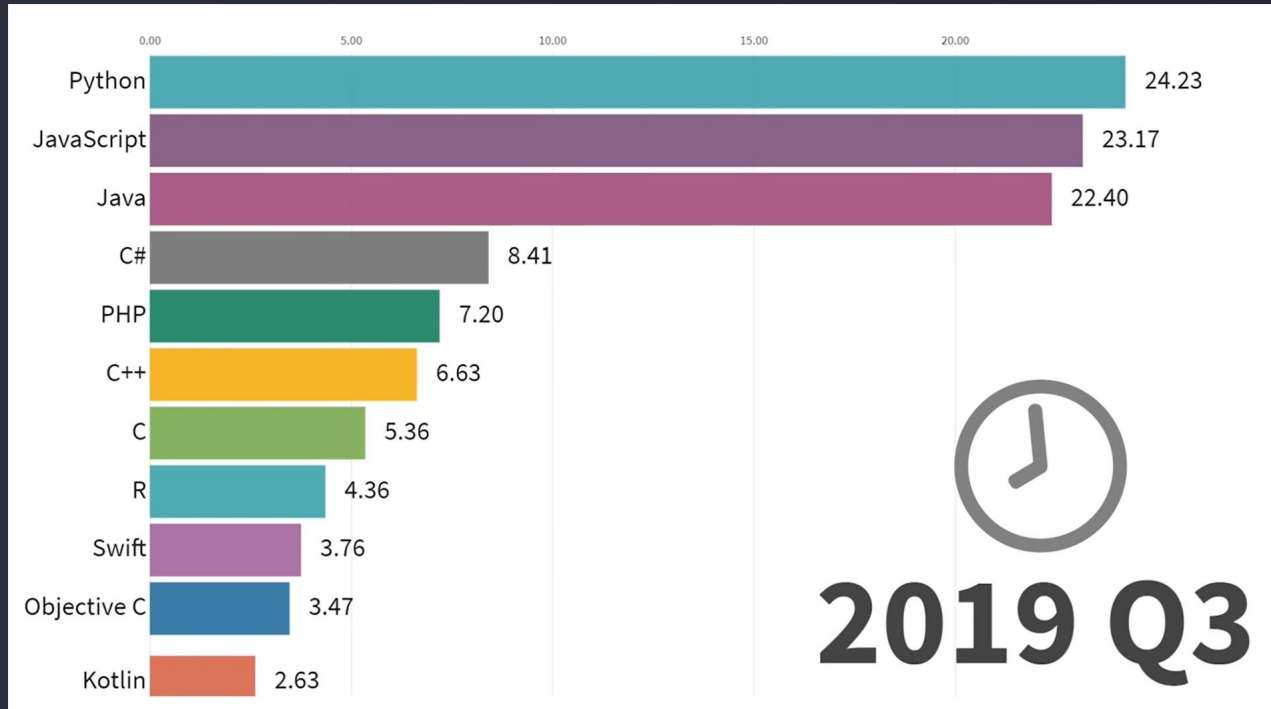
三天零基础入门Python



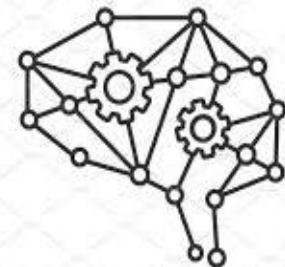
CSDN学院 IT实战派

第一章：课程导学

Python! Python! Python!



python



Python的历史和理念



优雅、明确、简单

Python的创始人
吉多·范罗苏姆 (Guido van Rossum)

本课程设计的核心思想



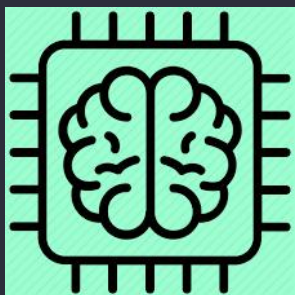
为了学PYTHON而学，目标缺失



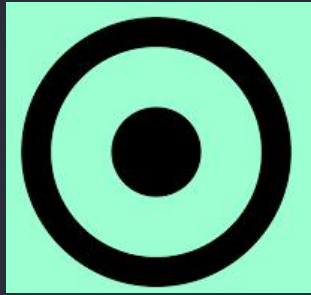
内容繁多复杂，耗时长



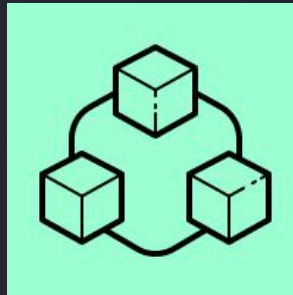
大量语法讲解，实用性不足



针对数据分析和人工智能定制



目标明确的简明课程设计



重点学习实用模块的实战

课程的预期目标



1. 以进入数据分析和人工智能行业为目标，快速掌握最实用的Python知识
2. 学习实际工作中最常用的PYTHON模块
3. 获得PYTHON编码和实战的经验，为未来学习工作打好基础

| 讲师介绍-Charlie老师



- 人工智能算法科学家-曾服务于某世界500强中国AI Lab，后自主创业
- 2019国家培训计划讲师-面向高职院校老师的培训课程
- 深圳市海外高层次人才认定(孔雀人才)
- 美国圣地亚哥国家超算中心博士后
- 加利福尼亚大学圣地亚哥全奖博士
- 参与美国自然科学基金(NSF)及加州能源局 (CEC)资助的392MW IVANPAH等智慧电网项目
- 21篇国际期刊文章(sci收录17篇)，总引用接近1000
- 第一作者发明专利11份

课程纲要

- Python开发环境的准备(Anaconda, Jupyter Notebook, Spyder)
- Python基础知识 (变量类型, 基础数学运算, 基础语法, 常用数据结构, 比较和逻辑运算, 循环语句, Python函数, 文件IO, 时间模块与OS模块, 异常处理)
- Numpy的使用 (ndarray, 向量与矩阵的形变和运算, 向量和矩阵的生成、运算、拼接、和索引, 常用numpy科学计算方法)
- Pandas 和 Matplotlib
- 爬虫实战 (Request, BeautifulSoup)

| Python开发的环境



过了若干时间后。

课程相关资料





欢迎大家扫码或者添加微信好友ai_flare（学习小助手），加入学习群，老师会在群里和大家进行交流和答疑（名额有限、人满即止）

EDU

CSDN学院 IT实战派

第二章：Python基础知识

| 变量的类型

Int

Float

String

| Python基础数学运算

名称	符号
加	+
减	-
乘	*
除	/
模量	%
指数	**

| Python基础语法

- Print和输出
- 行和缩进
- 多行语句
- 注释

| 常用数据结构

- list
- tuple
- dict
- set

| Python比较运算

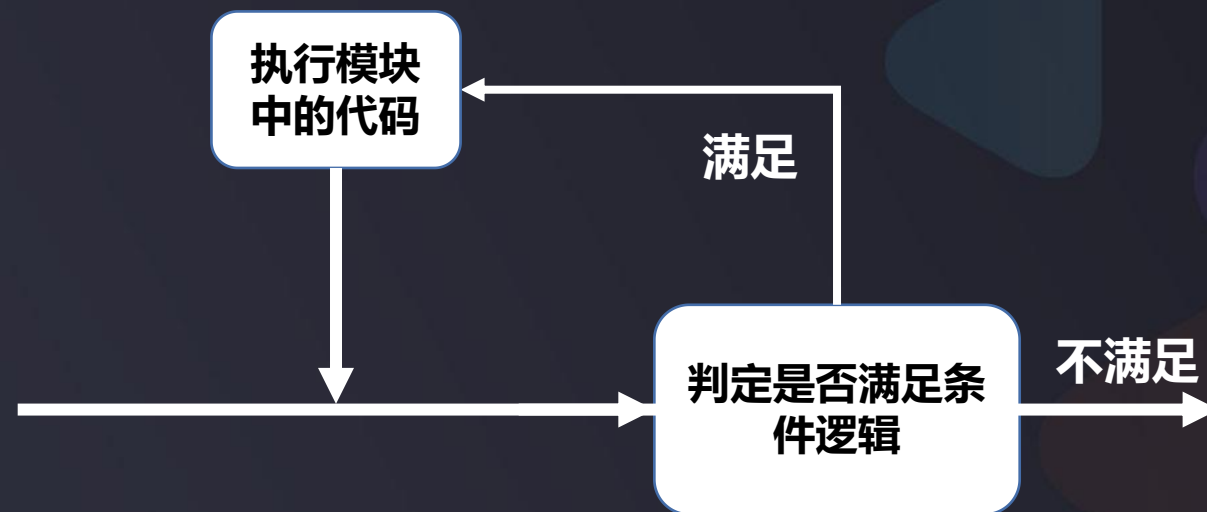
名称	符号
大于	>
小于	<
等于	==
大于等于	>=
小于等于	<=
不等于	!=

| Python逻辑运算

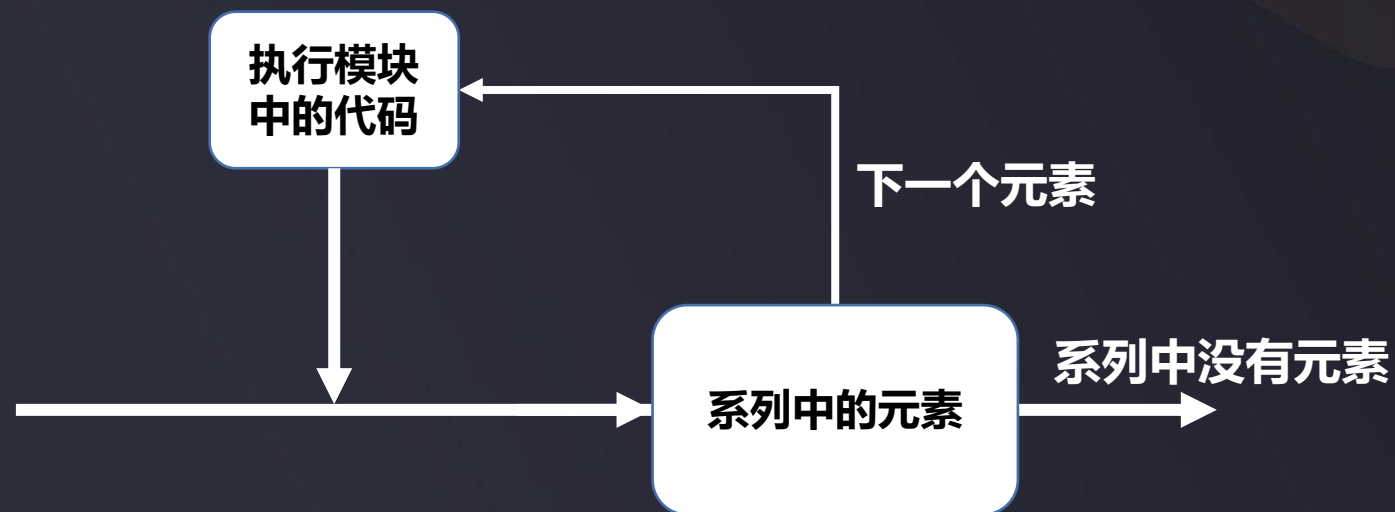
名称	符号
和	and
或	or
非	not

| 循环语句

While



For



本章回顾

- 变量
- 基础数学运算
- 基础语法
- 常用数据结构
- 比较运算
- 逻辑运算
- 循环语句

int, float, string

+ - * ÷ % **

print, indent, == -, -, -

List, Tuple, Dict, Set

> < == >= <= !=

and, or, not

if ~:

else: ~

while
for

I 课后习题

1. 用循环语句实现求和1到200 (1+2+3+... +199+200)

tip

Q

2. 找到13个能被13整除的数字 (比如13,26等, 一共13个)

$$13 \div 13 = 0$$

$$26 \div 13 = 0$$

$$26 \div 13 = 2$$

课程相关资料





欢迎大家扫码或者添加微信好友ai_flare（学习小助手），加入学习群，老师会在群里和大家进行交流和答疑（名额有限、人满即止）



CSDN学院 IT实战派

第三章：Python基础知识2

| 上一章回顾

- 变量
- 基础数学运算
- 基础语法
- 常用数据结构
- 比较运算
- 逻辑运算
- 循环语句

| 习题讲解

1. 用循环语句实现求和1到200 ($1+2+3+\dots+199+200$)
2. 找到13个能被13整除的数字 (比如13,26等, 一共13个)

| 本章学习内容

- Python函数
- 文件IO
- 模块（时间模块与OS模块）
- 异常处理

Python函数

- 函数是组织好的，可重复使用的，用来实现单一，或相关联功能的代码段。
- 函数能提高应用的模块性，和代码的重复利用率。

Print def

```
functionname( parameters ):  
    function_suite  
    return [expression]
```

| 文件IO

```
file=open("test.txt","r")
```

```
    file.read()
```

```
    file.read(100)
```

```
    file.readline()
```

```
    file.readlines()
```

```
file=open("test.txt","w")
```

```
    file.write("你好")
```

```
file=open("test.txt","a")
```

```
file.close()
```

| Python模块

```
import time
```

```
time.time()
```

```
time.strftime('%Y-%m-%d %H:%M:%S')
```

```
import os
```

```
os.getcwd()
```

```
os.listdir()
```

```
os.mkdir("temp2")
```

```
os.chdir()
```

| 异常处理

try:

运行代码

except:

如果在try部份引发了异常的应对代码

| 本章回顾

- Python函数
- 文件IO
- 模块 (时间模块与OS模块)
- 异常处理

Def xxx(x):

return

try except

I 课后习题

1. 定义一个函数，他能从输入的列表[2,6,7,4,9,5,1,0,3,8]中找到最大值，输出运算的时间，并把结果写成一个文本文件存在运行脚本的同一路径/目录下

[1, 3, ..., 99, ...]



CSDN学院 IT实战派

第四章：Python数据分析之Numpy

I 上一章回顾

- Python函数
- 文件IO
- 模块（时间模块与OS模块）
- 异常处理

| 习题讲解

1. 定义一个函数，他能从输入的列表[2,6,7,4,9,5,1,0,3,8]中找到最大值，输出运算的时间，并把结果写成一个文本文件存在运行脚本的同一路径/目录下

| Numpy简介



NumPy 是一个运行速度非常快的数学库，主要用于数组计算，包含：

- 一个强大的N维数组对象 ndarray
- 广播功能函数
- 整合 C/C++/Fortran 代码的工具
- 线性代数、傅里叶变换、随机数生成等功能

| Numpy简介



NumPy 是一个运行速度非常快的数学库，主要用于数组计算，包含：

- 一个强大的N维数组对象 ndarray
- 广播功能函数
- 整合 C/C++/Fortran 代码的工具
- 线性代数、傅里叶变换、随机数生成等功能

Numerical Python
库

| Spyder介绍



Spyder(Scientific PYthon Development EnviRonment)是一个强大的交互式 Python 语言开发环境, 提供高级的代码编辑、交互测试、调试等特性, 支持包括 Windows、Linux 和 OS X 系统。

- 提供了变量的可视化
- 可以逐行执行代码, 便于DEBUG
- 很便捷的查看路径和执行LOG

$w = [1, 2, 3]$

Matlab

print

Numpy的ndarray

载入numpy

```
import numpy as np
```

定义ndarray

```
a=np.array([1,2,3,4])
```

```
b=np.array([[1,2,3],[3,2,1]])
```

```
c=np.array([[1.,2.5,3.14],[13.1,2.6,1.2]])
```

```
d=np.array([[1,2,3,4]])
```

一些查看的指令

```
a.size
```

```
a.shape
```

```
b.shape
```

```
b.size
```

```
b.ndim
```

1×4
 4×1

1×4

0
1
2
3
4

0	1	2
1	2	3
3	2	1

0	1	2	3
1	2	3	4

Numpy的使用

向量生成

`np.arange(10)`

`np.linspace(0,10,20)`

0, 1, ..., 9

矩阵生成

`temp=np.zeros((5,5))`

`temp=np.ones((5,5))`

2x[0 0]

uniform distribution between 0 and 1

`temp=np.random.rand(5,5)`

#Gaussian distribution with mean 0 and variance 1

`temp=np.random.randn(5,5)`

	0	1	2	3	4
0	-1.38569	2.25316	-0.438421	-0.878835	-0.0026496
1	-0.0846714	-0.196645	1.14361	-0.539697	-1.03301
2	-0.3868	-1.77855	0.0083608	-1.13897	-0.0286621
3	0.903734	0.393386	-0.409159	0.0630359	0.492297
4	-0.0541013	-1.77399	0.54087	0.85008	0.116791

Numpy的使用

矩阵运算

$a+1$

$a*b$

$\text{np.dot}(a,b)$

$a.T$

矩阵的拼接

$\text{np.vstack}([a,b])$

$\text{np.hstack}([a,b])$

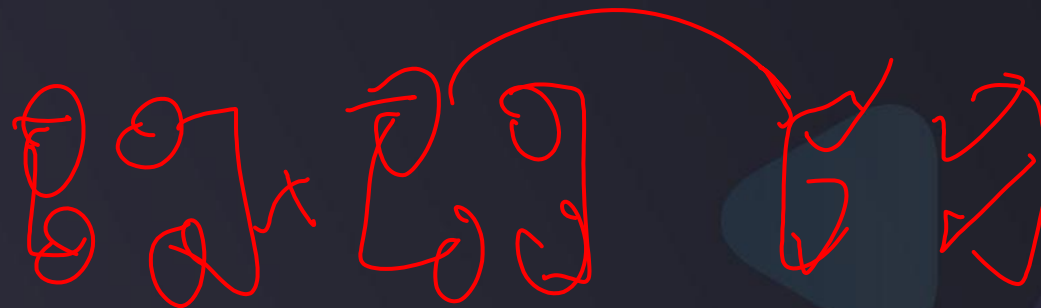
$\text{np.concatenate}((a,b), \text{axis}=0)$

$\text{np.concatenate}((a,b), \text{axis}=1)$

矩阵形变

$\text{.reshape}(2,2)$

3×2
 $\rightarrow 2 \times 3$



	0	1	2
0	1	2	3

X

	0
0	1
1	2
2	3

1×2
 2×2
 $m \times n \cdot n \times k$
 (4×4)
 (2×8)
 $= m \times k$

向量索引

向量索引

```
print(a[0])
```

```
print(a[0:3])
```

```
print(a[1:3])
```

```
print(a[3:])
```

```
print(a[:3])
```

```
print(a[1:-1])
```

矩阵索引

```
print(b[2,0])
```

```
print(b[2][0])
```

```
print(b[3])
```

```
print(b[3,:])
```

```
print(b[:,0])
```

	0	1	2	3
0	1	2	3	4

	0	1	2	3	4
0	-1.38569	2.25316	-0.438421	-0.878835	-0.0026496
1	-0.0846714	-0.196645	1.14361	-0.539697	-1.03301
2	-0.3868	-1.77855	0.0083608	-1.13897	-0.0286621
3	0.903734	0.393386	-0.409159	0.0630359	0.492297
4	-0.0541013	-1.77399	0.54087	0.85008	0.116791

$[2, 0]$

$[2][0]$

$[3, :]$

$b[1:3, 4:5]$

| Numpy常用科学计算

np.log(a)
np.max(a)
np.mean(a)

np.std(a)
np.count(a)

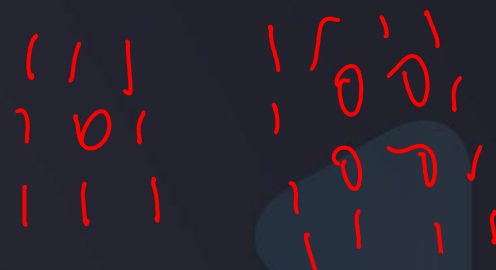
np.sin(a)
np.exp(a)
np.abs(a)

np.argmax(a)
np.argmin(a)

| 本章回顾

- Numpy介绍和使用
- Spyder介绍和使用
- Nddarray
- 向量与矩阵的形变和运算
- 向量和矩阵的拼接和生成
- 向量和矩阵的索引
- 常用numpy科学计算方法

课后习题



1. 创建一个10*10的ndarray对象，且矩阵边界全为1，里面全为0
2. 创建一个5X3随机矩阵和一个3X2随机矩阵，求矩阵积以及其最大值，最小值，平均值和标准差
3. $y = x^{e^{x/3}}$ ，如果 $y=10$ ，求解 x ，(用数值法求解)

$$x = 0.000102 \dots 10$$

$$x \rightarrow y = 10$$

课程相关资料





欢迎大家扫码或者添加微信好友ai_flare（学习小助手），加入学习群，老师会在群里和大家进行交流和答疑（名额有限、人满即止）



第五章：Pandas与Matplotlib

| 上一章回顾

Numpy介绍和使用

Spyder介绍和使用

Ndarray

向量与矩阵的生成

向量和矩阵的运算，拼接和形变

向量和矩阵的索引

常用numpy科学计算方法

I 习题讲解

1. 创建一个10*10的ndarray对象，且矩阵边界全为1，里面全为0
2. 创建一个5X3随机矩阵和一个3X2随机矩阵，求矩阵积以及其最大值，最小值，平均值和标准差
3. $y = x^{e^{x/3}}$ ，如果 $y=10$ ，求解 x ，(用数值法求解)

| Pandas简介



Pandas是为了解决数据分析任务而创建的。Pandas纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。是使Python成为强大而高效的数据分析环境的重要因素之一

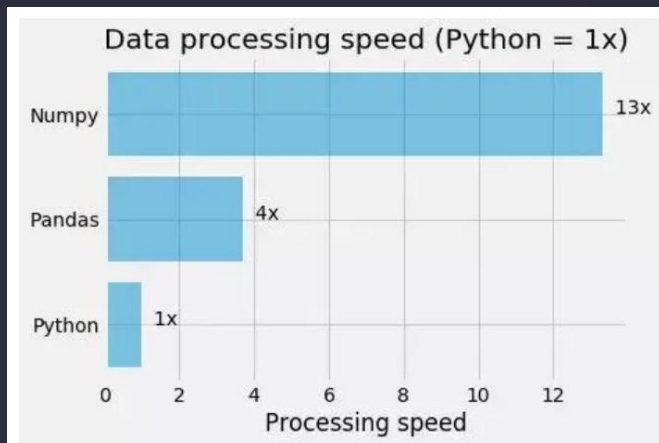
Panel data



Pandas 与 Numpy 对比



VS



- 可以轻易的处理浮点及非浮点数据类型的缺失值(NaN)
- 大小可变: DataFrame和Panel都可以删除或插入列
- 数据自动对齐
- 灵活强大的分组功能, 可对数据集进行拆分组合操作
- 基于智能标签的切片, 花式索引, 轻易从大数据集中取出子集
- 直观的合并, 连接数据集
- 轻易的重新定义数据集形状和转置
- 轴(axes)的分层标签 (使每个元组有多个标签成为可能)

| Matplotlib介绍



matplotlib是基于Python语言的开源项目，旨在为Python提供一个数据绘图包，实现专业的绘图功能。

Pandas

数据导入和储存

```
data=pd.read_csv("data.csv")  
data.to_csv("data1.csv")
```

数据查看

```
data.index  
data.columns  
data.head()  
data["Job"]  
data.loc[3,"Job"]  
data.iloc[3:5,0:1]
```

data - DataFrame			
Index	Name	Age	Job
0	Alice	20	Analysis
1	Charlie	30	Teacher
2	Bob	40	Doctor
3	David	25	Police
4	Egg	34	nan
5	Fish	21	farmer
6	Grubby	54	worker
7	Harry	22	farmer
8	Ice	45	lawyer

| Pandas数据预处理

数据连接

```
pd.read_csv("data.csv")
```

```
pd.read_csv("data.csv")
```

```
pd.concat([data1,data2])
```

```
data.append([data1,data2])
```

数据清洗

```
data.isnull()
```

```
data.dropna()
```

```
data.fillna(0)
```

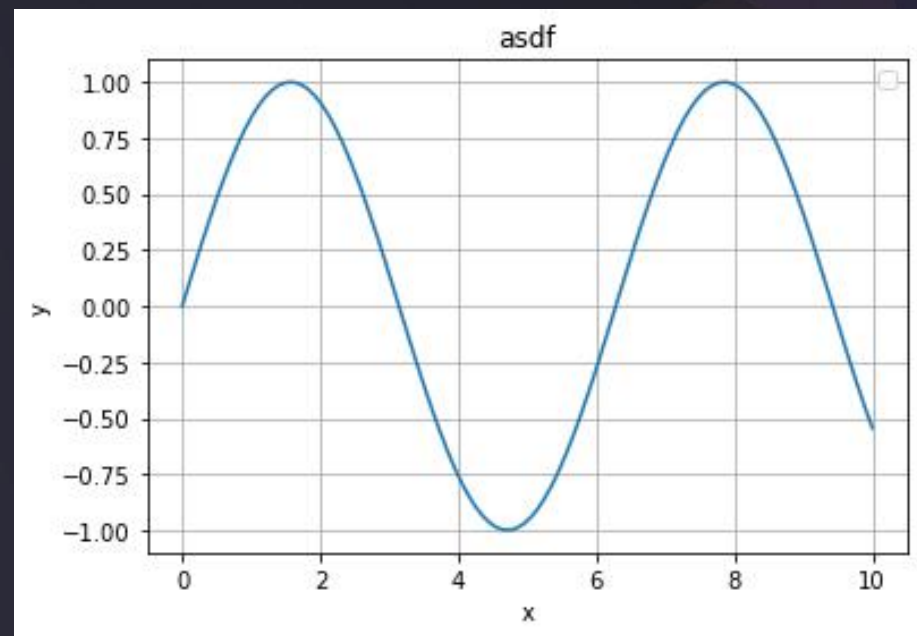
| Pandas的函数应用

```
data.drop(2)
data["Age"].sum()
data["Age"].mean()
data["Age"].std()
data["Age"].argmax()
data["Age"].argmin()
data.sort_values(by="Age")
```

```
data["Job"].unique()
data["Age"].values
data["Job"].describe()
```

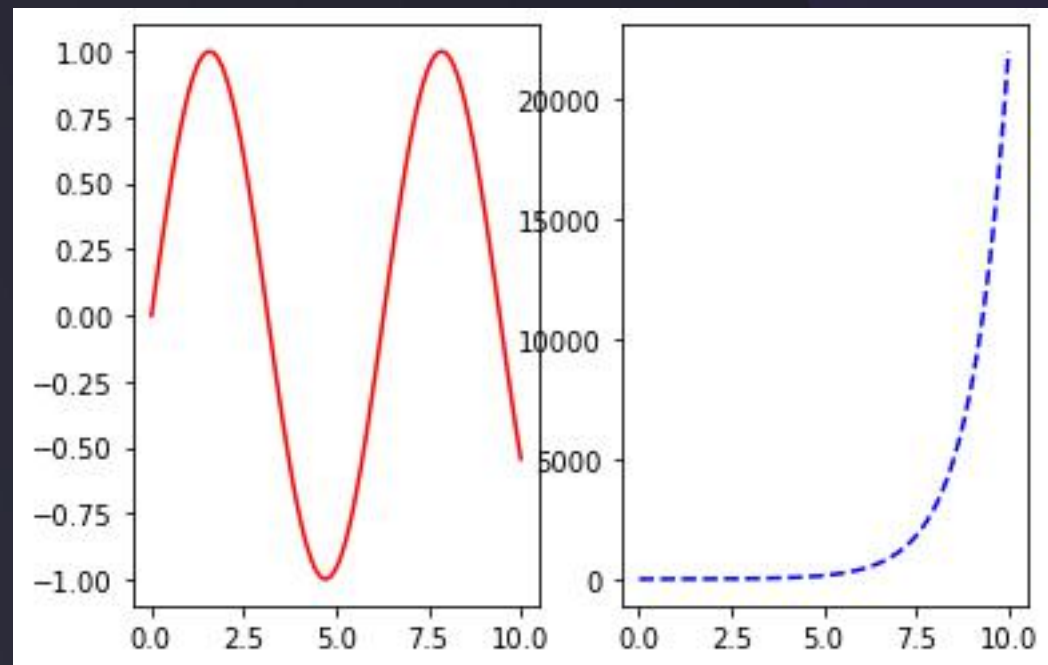
Matplotlib代码作图的实现1

```
import matplotlib.pyplot as plt  
plt.plot(x,y)  
plt.legend()  
plt.xlabel("x")  
plt.ylabel("y")  
plt.xlim(0,5)  
plt.ylim(0,2)  
plt.title("asdf")  
plt.grid(True)  
plt.show()
```



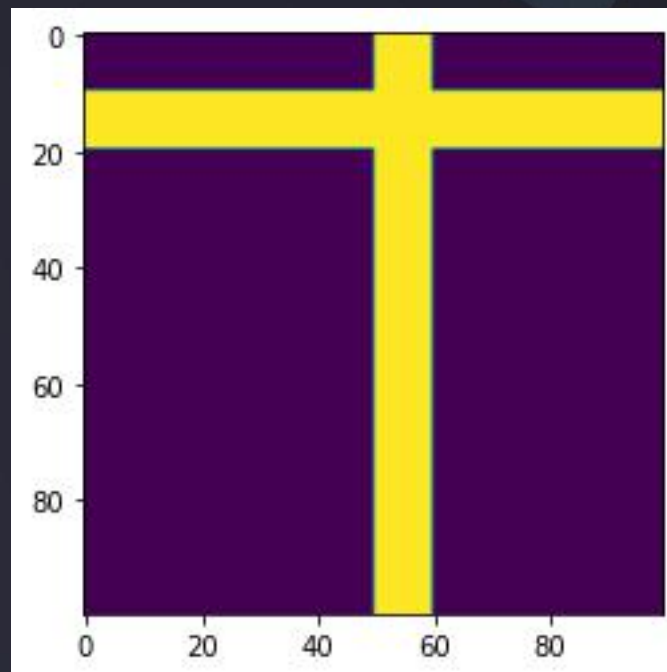
Matplotlib代码作图的实现2

```
plt.figure()  
plt.subplot(1,2,1)  
plt.plot(x,y1,"r-")  
plt.subplot(1,2,2)  
plt.plot(x,y2,"b--")
```



Matplotlib代码作图的实现3

`plt.imshow()`



| 本章回顾



课程相关资料





欢迎大家扫码或者添加微信好友ai_flare（学习小助手），加入学习群，老师会在群里和大家进行交流和答疑（名额有限、人满即止）



CSDN学院 IT实战派

第六章：简明爬虫实战

| 上一章回顾



I 爬虫简介

web crawler



- 自动获取网页内容的程序
- 高效获得网上的海量数据

I 爬虫简介



- 通过网址和DNS服务器找到服务器主机
- 发送请求获得浏览器结果
- 解析获得浏览器呈现的结果

I 几种常用爬虫工具

- Urllib
- Requests
- Scrapy
- Selenium

`pip install requests`

| Requests的使用

```
import requests
```

```
data=requests.get("https://www.csdn.net")
```

```
data.encoding
```

```
data.text
```

```
data.status_code
```

```
header = {'user-agent':'Mozilla/5.0'}
```

```
requests.get(url=url,headers=header, timeout=3)
```

```
requests.get(url="www.csdn.net",proxies=prox)
```

| status_code

200

2字头：代表请求已经被服务器成功接收和理解

3字头：重新定向

4字头：客户端发生错误

404

5,6字头：服务器发生错误或异常

| BeautifulSoup

```
pip install beautifulsoup4  
from bs4 import BeautifulSoup
```

```
BeautifulSoup(sample_text, 'lxml')  
soup.find(name="ul")  
soup.find_all(name="li")
```

```
sample_text='''  
<html>  
  <div>  
    <ul>  
      <li class="123"><a href="abc.html">hello</a></li>  
      <li class="123"><a>world</a></li>  
      <li class="abc"><a href="123.html">asdf</a></li>  
    </ul>  
  </div>  
</html>  
'''
```


| 网页搜索内容爬取实战

| 本章回顾



- Requests
- BeautifulSoup
- 网页爬取实战

| 课程回顾

- 环境准备
- Python基础知识和使用
- time和os
- Numpy
- Pandas
- Matplotlib
- 爬虫实战

课程相关资料





欢迎大家扫码或者添加微信好友ai_flare（学习小助手），加入学习群，老师会在群里和大家进行交流和答疑（名额有限、人满即止）