

《物理与人工智能》

自学课件1： Python基础

授课教师：马滢青

2024/09/08（第一周）

鸣谢：基于计算机学院《人工智能引论》课程组幻灯片



北京大学



1. Python简介
2. Python语法
3. Numpy库介绍
4. Matplotlib库介绍
5. Python、Numpy、Matplotlib安装
6. Python项目实例

1. Python简介

2. Python语法

3. Numpy库介绍

4. Matplotlib库介绍

5. Python、Numpy、Matplotlib安装

6. Python项目实例

Python简介



- ❖ Python是最**简单**实用的编程语言之一，也是当下最流行的编程语言之一。
- ❖ Python的作者是荷兰人Guido Van Rossum。
- ❖ Python的设想酝酿于19世纪80年代末并于1989年12月开始实现。
- ❖ 1991年-Python0.9.0问世标志着Python正式诞生。
- ❖ 2000年 -Python2.0发布，2020年Python 2.7.8发布后Python2.x停止更新。
- ❖ 2008年-Python3.0发布，Python3.x是目前最流行的版本，本课程编程项目均基于**Python3.11.9**实现。



Python和C++都是常用的编程语言。同C++一样，Python也具有**变量、函数、类等概念**，但Python具有**更简单的编程规则**和**更简洁直观的代码形式**：

Python示例

```
import os
```

```
a=[3,1,4,1,5,9,2,6,5,35]
```

```
b=10
```

```
while b>0:
```

```
    b-=1
```

```
    print(a[b])
```

导入包/库

Python不需要变量声明

通过:和缩进区分代码块
判断语句不需要()

语句结束不需要标识符

通过{}区分代码块
判断语句需要()

语句结束需要加标识符;

C++示例

```
#include <iostream>
```

```
using namespace std
```

```
int a[10];
```

```
a={3,1,4,1,5,9,2,6,5,35};
```

```
int b;
```

```
b=10
```

```
while(b>0)
```

```
{
```

```
    b-=1;
```

```
    cout<<a[b]<<endl;
```

```
}
```

Python和C++在语言类型上分别属于**解释性语言**和**编译性语言**：

Python

- 语言类型-**解释性语言**

执行程序时，通过解释器
逐行解释成机器指令并运行

- 特点

执行效率低，运行慢
跨平台移植容易

C++

- 语言类型-**编译性语言**

先编译**所有**代码生成可执行文件，再运行可执行文件

- 特点

执行效率高，运行快
跨平台移植困难

Python：编程容易，可移植性强

1. Python简介

2. Python语法

3. Numpy库介绍

4. Matplotlib库介绍

5. Python、Numpy, Matplotlib安装

6. Python项目实例



数字类型及运算符

本课程用到的数字类型包含：整型、浮点型、布尔型

①整型(int)表示整数

```
>>> type(1)
```

```
<class 'int'>
```

②浮点型(float)表示带小数点的数

```
>>> type(1.0)
```

```
<class 'float'>
```

③布尔型(bool)数据有True和False两种值，其中True=1， False=0

```
>>> type(True)
```

```
<class 'bool'>
```


数字类型及运算符

本课程主要用到的运算符如下：

运算符	说明	举例
+, -	加法、减法	4+5、4-5
**	幂运算	2**4
*, %, /, //	乘法、取余数、真除法、取整除法	4*5、5%2、5/2、5//2
<, <=, >, >=, ==, !=	比较运算:小于、小于等于、大于、大于等于、相等、不等	4<5、4<=5、4>5、4>=5、4==5、4!=5
not	逻辑非	not True、not 4 <5
and	逻辑与	x > 5 and x <100
or	逻辑或	x <5 or x >100



数字类型及运算符

① +、-: 加、减运算符完成加减法运算

>>> 4+5 (运算符两侧为**相同**数据类型时, 输出类型与输入类型**相同**)

9

>>> 4-5.0 (运算符两侧为**整型**和**浮点型**时, 输出类型为**浮点型**)

-1.0

② **: 幂运算

>>> 2**4.0 (当指数大于0时, 输出数据类型规则同加、减运算; 当指数小于0时, 输出类型为浮点数)

16.0

③ *: 乘法计算

>>> 4*5 (输出数据类型规则同加、减运算)

20

④ /: 真除法

>>> 5/4 (不论输入是整型还是浮点型, 输出值均为浮点型数据)

1.25



数字类型及运算符

⑤ %: 取余数

>>> 5%4 (输出数据类型规则同加、减运算)

1

⑥ //: 取整除法

>>> 5//2 (输出数据类型规则同加、减运算, 所输出浮点数的小数部分为0)

2

>>> 5.0//2

2.0

⑦ <、<=、>、>=、==、!=: 比较运算符的输出为True或False

>>> 4<5

True

>>> 4<=5

True

>>> 4>5

False

>>> 4>=5

False

>>> 4==5

False

>>> 4!=5

True



数字类型及运算符

⑧ not: 逻辑非, 将输入的True变成False, 输入的False变成True

```
>>> not True
```

```
False
```

```
>>> not False
```

```
True
```

```
>>> not 5>4 (5>4为逻辑值)
```

```
False
```

```
>>> not -0.1 (非0值默认为True)
```

```
False
```

⑨ and: 逻辑与, 当两侧输入均为True时返回True, 否则返回False

```
>>> True and True
```

```
True
```

```
>>> True and False
```

```
False
```

```
>>> False and True
```

```
False
```

```
>>> 1.1 and 5
```

```
True
```

⑩ or: 逻辑或, 当两侧输入均为False时返回False, 否则返回True

```
>>> False or False
```

```
False
```

```
>>> False or True
```

```
True
```

```
>>> True or True
```

```
True
```

```
>>> 0 or 4>5
```

```
False
```

变量与赋值

- ❖ 变量为代表特定对象(如数字)的字母符号，如**a**, **a1**...
- ❖ 赋值符号 “=” 用来将表达式的值赋予一个变量
- ❖ 变量在第一次赋值时被创建，在后续代码中可**直接调用**

```
>>> a=1      # 创建变量a, a的值为1
>>> b=2      # 创建变量b, b的值为2
>>> a+b      # 调用变量a、b, 完成加法计算
```

3

变量命名规则：

- ①必须以下划线或字母开头，后面接任意数量的下划线、字母或数字，如a, a1, _a, aabcd1为合法变量名，1a为非法变量名
- ②变量名区分大小写，例如Abc和abc是两个不同的变量
- ③禁止使用Python保留字(也称关键字)。保留字在Python中具有特殊意义，用保留字作变量名会造成语法错误。Python3.11中的保留字包括：
'False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'。

判断类型是指根据是否满足条件选择性执行语句：

```
a=input('请输入一个整数...\n')
```

```
a=int(a)
```

```
if a>0:
```

```
    print('a 为正数')
```

```
elif a<0:
```

```
    print('a 为负数')
```

```
else:
```

```
    print('a 为0')
```

#input为获取键盘输入的关键字

#数据类型转换

#判断语句

#当满足 $a>0$ 时输出‘a 为正数’

#若前一个if不成立，执行判断

#当满足 $a<0$ 时输出‘a 为负数’

#若前面的if及elif均不成立，执行下方语句

#输出‘a 为0’

Python中，循环类型实现重复调用特定代码块：

①for循环：遍历一个序列，在序列终止前持续调用下方代码块：

```
for i in [1,2,3,4,5]:  
    print(i)
```

②while循环：当满足条件时连续执行语句，直至不满足条件时终止：

```
n=10  
while n>0:  
    print(n)  
    n=n-1
```


函数是完成特定处理过程的代码：

`def add(x1, x2):` #def是定义函数的关键字，括号“()”内为输入变量，可为空

`y=x1+x2`

`return y` #return为返回值关键字，return后的内容为函数输出值，运行到return退出函数

`>>>x1=3`

`>>>x2=5`

`>>>add(x1,x2)` #调用add函数

8

缩进用来区分不同的代码块，Python对缩进有严格的要求：

①在def, for, while等特定关键字所在的语句后需要添加缩进：

def add(x1, x2): # : def, for, while所在语句结尾的冒号是代码块开始的标志

y=x1+x2 # 从该行开始为函数执行内容，需要加缩进

return y

for i in [1,2,3,4,5]:

print(i) # 从该行开始为循环执行内容，需要加缩进

print('ending...') # 去掉缩进表示for循环执行的代码块结束，该语句不属于for循环执行内容

②在其他代码中不可改变缩进：

x1=2

x2=3# 错误使用缩进

Python基础

1. Python简介

2. Python语法

3. Numpy库介绍

4. Matplotlib库介绍

5. Python、Numpy、Matplotlib安装

6. Python项目实例

Numpy介绍

Numpy(Numerical Python)是Python语言的扩展程序库，支持多维度的矩阵计算。通常导入使用时 `import numpy as np`

①创建矩阵(数组):

```
>>> np.array([1,2,3,4,5]) #创建数值为设定值的数组
```

```
array([1, 2, 3, 4, 5])
```

```
>>> np.zeros((3,4)) #创建数值全为0的3×4数组
```

```
array([[0., 0., 0., 0.],  
       [0., 0., 0., 0.],  
       [0., 0., 0., 0.]])
```

```
>>> np.random.randn(3,4) #创建数值为随机值的3×4数组
```

```
array([[ 1.76405235,  0.40015721,  0.97873798,  2.2408932 ],  
       [ 1.86755799, -0.97727788,  0.95008842, -0.15135721],  
       [-0.10321885,  0.4105985 ,  0.14404357,  1.45427351]])
```

Numpy介绍

②矩阵(数组)索引(*在计算机中, 首位的序号通常为0):

```
>>> a=np.random.randn(3,4)      #创建数值为随机值的3×4数组
>>> a[0]                        #a为二维数组, a[0]表示第0行数据
array([1.76405235, 0.40015721, 0.97873798, 2.2408932 ])
```

```
>>> a[0][0]                    #索引第0行第0个数据
1.764052345967664
```

③矩阵(数组)分片索引:

```
>>> a=np.random.randn(3,4)
>>> a[0:2]                    #获取第0行到第1行的数据(从0开始, 到2之前, 不包含2)
array([[ 1.76405235,  0.40015721,  0.97873798,  2.2408932 ],
       [ 1.86755799, -0.97727788,  0.95008842, -0.15135721]])
```

```
>>> a[1:3,:2]                #获取在第1行到第2行和第0列到第1列中的数据
array([[ 1.86755799, -0.97727788],
       [-0.10321885,  0.4105985 ]])
```

#注: 切片索引中0可省略, a[1:3,:2]等同于a[1:3][:2]

④矩阵(数组)加减运算:

```
>>> a
array([[0, 1],
       [2, 3]]) #a为2×2数组
>>> b
array([[5, 6],
       [7, 8]]) #b为2×2数组
>>> a+b
array([[ 5,  7],
       [ 9, 11]]) #数组对应位置相加
>>> a-b
array([[ -5, -5],
       [-5, -5]]) #数组对应位置相减
```

```
>>> np.add(a,b) # 同 '+'
array([[ 5,  7],
       [ 9, 11]])
```

```
>>> np.subtract(a,b) # 同 '-'
array([[ -5, -5],
       [-5, -5]])
```

⑤矩阵(数组)乘运算:

a. *乘法

```
>>> a
array([[0, 1],
       [2, 3]])          #a为2×2数组
>>> b
array([[5, 6],
       [7, 8]])          #b为2×2数组
>>> a*b
array([[ 0,  6],
       [14, 24]])        #数组对应位置相乘
>>> np.multiply(a,b)    #同 '*'
array([[ 0,  6],
       [14, 24]])
```

b. dot乘法

```
>>> np.dot(a,b)
array([[ 7,  8],
       [31, 36]])
```

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 7 & 8 \\ 31 & 36 \end{bmatrix}$$

numpy.dot() 对于两个一维的数组，计算的是两数组对应下标元素的乘积和(数学上称之为**内积**)；对于二维数组，计算的是两数组的**矩阵乘积**；对于多维数组，它的通用计算公式如下：

$$\text{dot}(a, b)[i,j,k,m] = \text{sum}(a[i,j,:] * b[k,:,m]),$$

即结果数组中的每个元素都是：数组a的最后一维上的所有元素与数组b的倒数第二位上的所有元素的乘积和^[1]。

Numpy介绍



⑥矩阵(数组)除运算:

```
>>> a
array([[0, 1],
       [2, 3]])  #a为2×2数组
>>> b
array([[5, 6],
       [7, 8]])  #b为2×2数组
>>> a/b
array([[ 0. ,  0.16666667],
       [0.28571429,  0.375   ]])  #数组对应位置相除
>>> np.divide(a,b)
array([[ 0. ,  0.16666667],
       [0.28571429,  0.375   ]])  #同 '/'
```


⑦修改数组形状:

```
>>> a
array([[0, 1],
       [2, 3]])
>>> a.reshape(4)      #将a转化成向量
array([0, 1, 2, 3])
>>> a.reshape(1, 4)   #将a转化成1×4矩阵
array([[0, 1, 2, 3]])
```

***通过reshape(m,n,p...)可将数组转化成任意维度，但需要满足 $m \times n \times q \dots = M$ ，其中M为原始数组所包含的数据元素个数**

Numpy介绍



⑧其他数学函数：

```
>>> a = np.array([0,30,45,60,90])           #创建数组
>>> np.sin(a*np.pi/180)                     #计算正弦, numpy.pi=π
array([0.,  0.5,  0.70710678,  0.8660254,  1.])
>>> np.cos(a*np.pi/180)                     #计算余弦
array([1.,  0.8660254,  0.70710678,  0.5,  0.])
>>> np.power(a,2)                           #幂函数
array([ 0, 900, 2025, 3600, 8100], dtype=int32)
>>> a**2                                    #同为幂函数
array([ 0, 900, 2025, 3600, 8100], dtype=int32)
```

⑨强制数据类型转换：

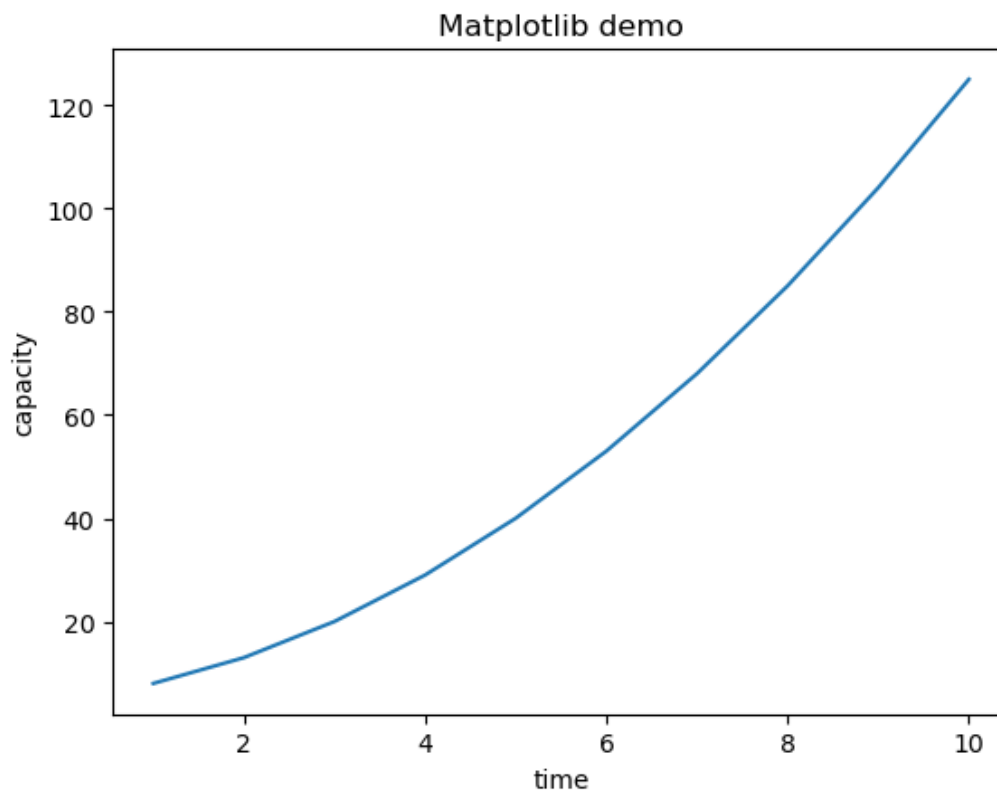
```
>>> a=np.array([1.76405235, 0.40015721, 0.97873798, 2.2408932 ])
>>> a.dtype                                #查看数据类型
dtype('float64')
>>> a.astype('int16')                     #强制类型转换
array([1, 0, 0, 0], dtype=int16)         #从浮点到整型转换出现精度损失(自动取整)
```

1. Python简介
2. Python语法
3. Numpy库介绍
- 4. Matplotlib库介绍**
5. Python、Numpy、Matplotlib安装
6. Python项目实例

Matplotlib介绍

Matplotlib是一种Python的绘图工具包，同时兼容Numpy，可实现图像显示、曲线绘制、柱状图绘制等功能。

```
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1,2,3,4,5,6,7,8,9,10])
y = x**2+2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("time")
plt.ylabel("capacity")
plt.plot(x,y)
plt.show()
```



包或模块
导入对象重命名

从轴变量绘制曲线

Python基础



北京大学
PEKING UNIVERSITY

1. Python简介
2. Python语法
3. Numpy库介绍
4. Matplotlib库介绍
5. Python、Numpy、Matplotlib安装
6. Python项目实例

Python安装

①检查系统是否已安装Python

- ❖ 对于Windows系统，Win+R调用系统运行窗口，输出cmd打开命令行；MacOS系统直接打开“终端”
- ❖ 输入python，若出现报错，说明未安装python；若出现 >>>，说明已安装python

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.2604]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\18428>python
'python' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [版本 10.0.19045.2604]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\18428>python
Python 3.10.10 (tags/v3.10.10:aad5f6a, Feb  7 2023, 17:20:36) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Python安装

②Python安装包下载

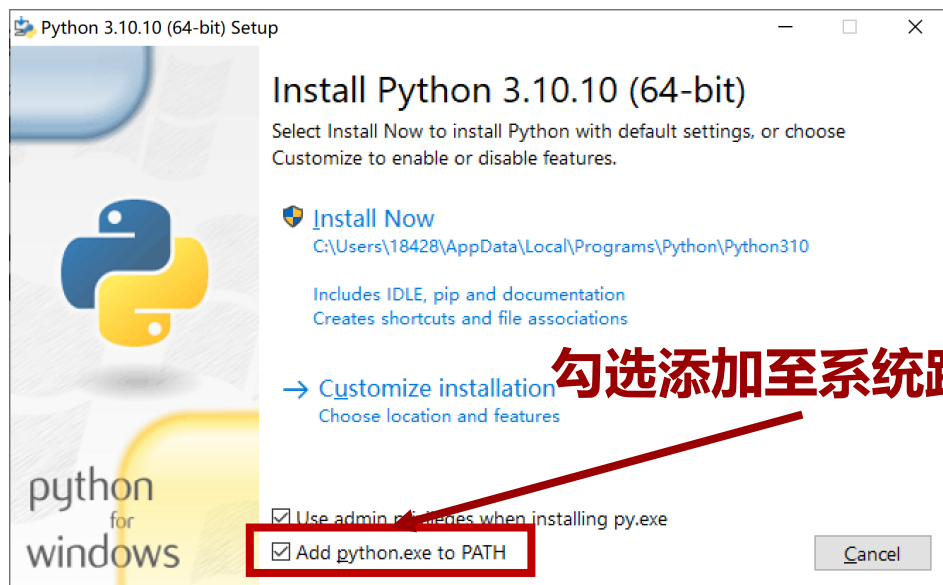
❖ Windows:

❖ 64位: <https://www.python.org/ftp/python/3.10.10/python-3.10.10-amd64.exe>

❖ 32位: <https://www.python.org/ftp/python/3.10.10/python-3.10.10.exe>

❖ macOS:

❖ 64位: <https://www.python.org/ftp/python/3.10.10/python-3.10.10-macos11.pkg>



勾选添加至系统路径



若上一步选择customize installation, 此处选择pip, 其余内容可以任意设置

Python安装

③Numpy安装

打开命令行输入：

```
python -m pip install numpy
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.2604]
(c) Microsoft Corporation。保留所有权利。

C:\Users\18428>python -m pip install numpy
Collecting numpy
  Using cached numpy-1.24.2-cp310-cp310-win_amd64.whl (14.8 MB)
Installing collected packages: numpy
Successfully installed numpy-1.24.2

C:\Users\18428>
```


Python安装

④Matplotlib安装

打开命令行输入：

`python -m pip install matplotlib`

```
C:\WINDOWS\system32\cmd.exe
C:\Users\18428>python -m pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.7.0-cp310-cp310-win_amd64.whl (7.6 MB)
----- 7.6/7.6 MB 4.7 MB/s eta 0:00:00
Collecting pyparsing>=2.3.1
  Using cached pyparsing-3.0.9-py3-none-any.whl (98 kB)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.4-cp310-cp310-win_amd64.whl (55 kB)
----- 55.3/55.3 kB 2.8 MB/s eta 0:00:00
Collecting python-dateutil>=2.7
```

```
C:\WINDOWS\system32\cmd.exe
----- 163.0/163.0 kB 4.9 MB/s eta 0:00:00
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, pyparsing, pillow, packaging, kiwisolver, fonttools, cycler, contourpy, python-dateutil, matplotlib
Successfully installed contourpy-1.0.7 cycler-0.11.0 fonttools-4.38.0 kiwisolver-1.4.4 matplotlib-3.7.0 packaging-23.0 pillow-9.4.0 pyparsing-3.0.9 python-dateutil-2.8.2 six-1.16.0
C:\Users\18428>
```

利用VSCode来安装

详见下发说明文件，包含：安装VSCode等、连接服务器

1. Python简介
2. Python语法
3. Numpy库介绍
4. Matplotlib库介绍
5. Python、Numpy、Matplotlib安装
6. Python项目实例

Python项目实例



```
import numpy as np
import matplotlib.pyplot as plt
x=np.array([0,1,2,3,4,5,6,7,8,9,10])
y_sin=np.sin(x*36*np.pi/180)
y_f=x**2-10*x+25
img=plt.imread('minist.jpg')
fig=plt.figure(figsize=(12,4))
fig.patch.set_facecolor('lightgrey')
plt.subplot(131)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.plot(x,y_sin)
plt.subplot(132)
plt.xlabel('x')
plt.ylabel('f(x)')
plt.bar(x,y_f)
plt.subplot(133)
plt.axis('off')
plt.imshow(img,cmap='gray')
plt.show()
```

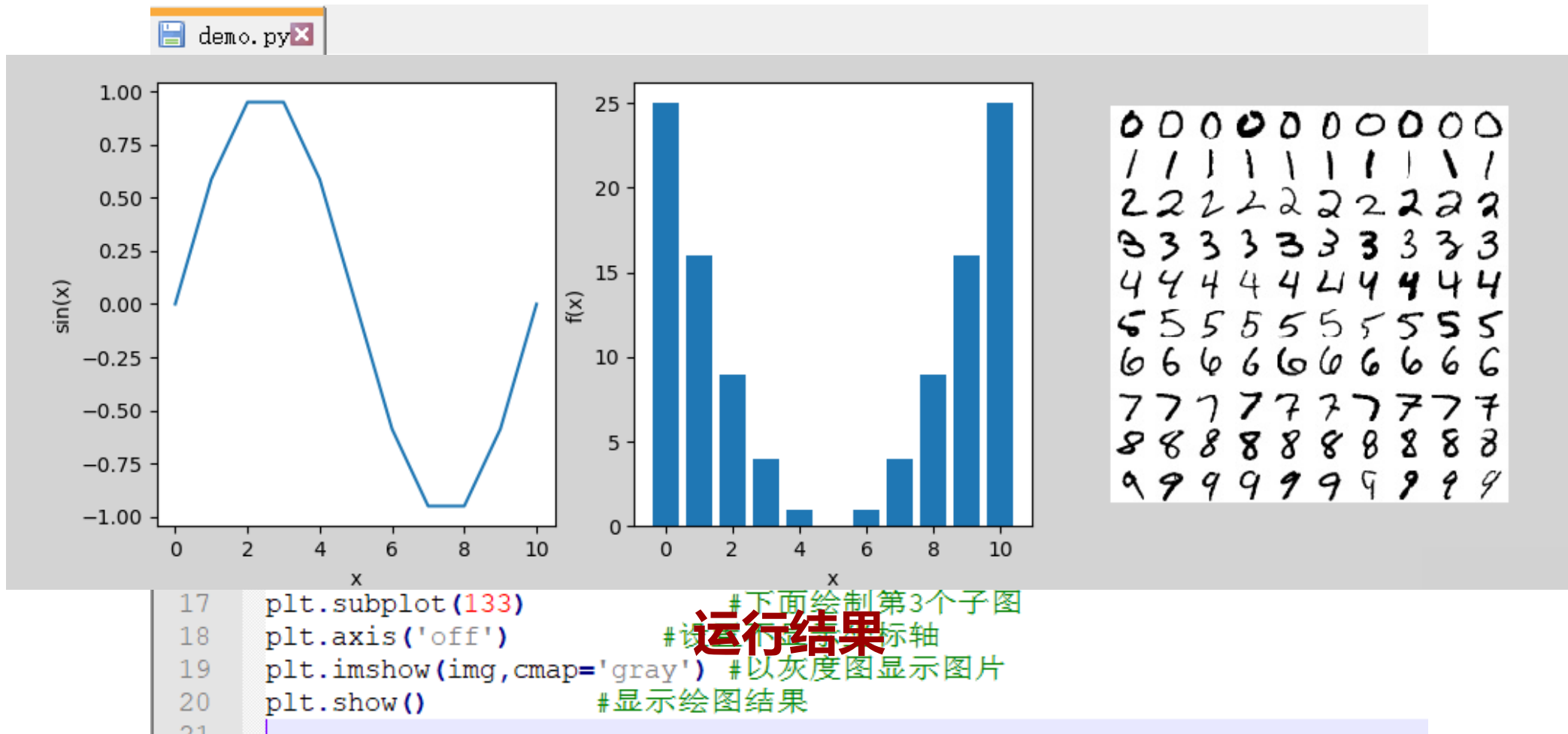
```
#导入numpy
#导入matplotlib
#创建数组
#计算正弦值
#计算二次函数值
#读取图像
#设置绘图大小
#设置绘图背景
#绘图区域分成1行3列，下面绘制第1个子图
#设置横坐标标题
#设置纵坐标标题
#绘制正弦曲线
#下面绘制第2个子图
#设置横坐标标题
#设置纵坐标标题
#绘制柱状图
#下面绘制第3个子图
#设置不显示坐标轴
#以灰度图显示图片
#显示绘图结果
```

Python项目实例



北京大学
PEKING UNIVERSITY

创建demo.py文件，文件内写入以上代码，命令行运行：
Python demo.py



运行结果

谢谢



北京大学
PEKING UNIVERSITY

