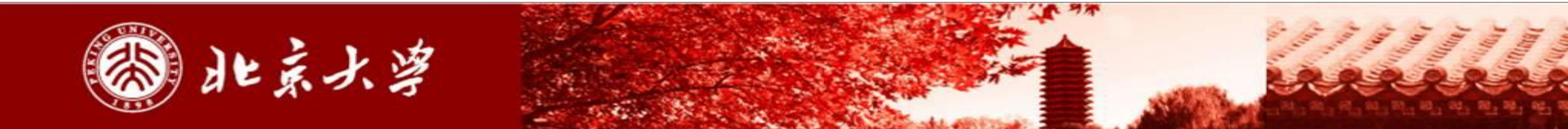
《物理与人工智能》

自学课件1： Python基础

授课教师：马滟青

2026/09/07（第一周）

鸣谢：基于计算机学院《人工智能引论》课程组幻灯片



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简介

北京大学 PEKING UNIVERSITY

❖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。



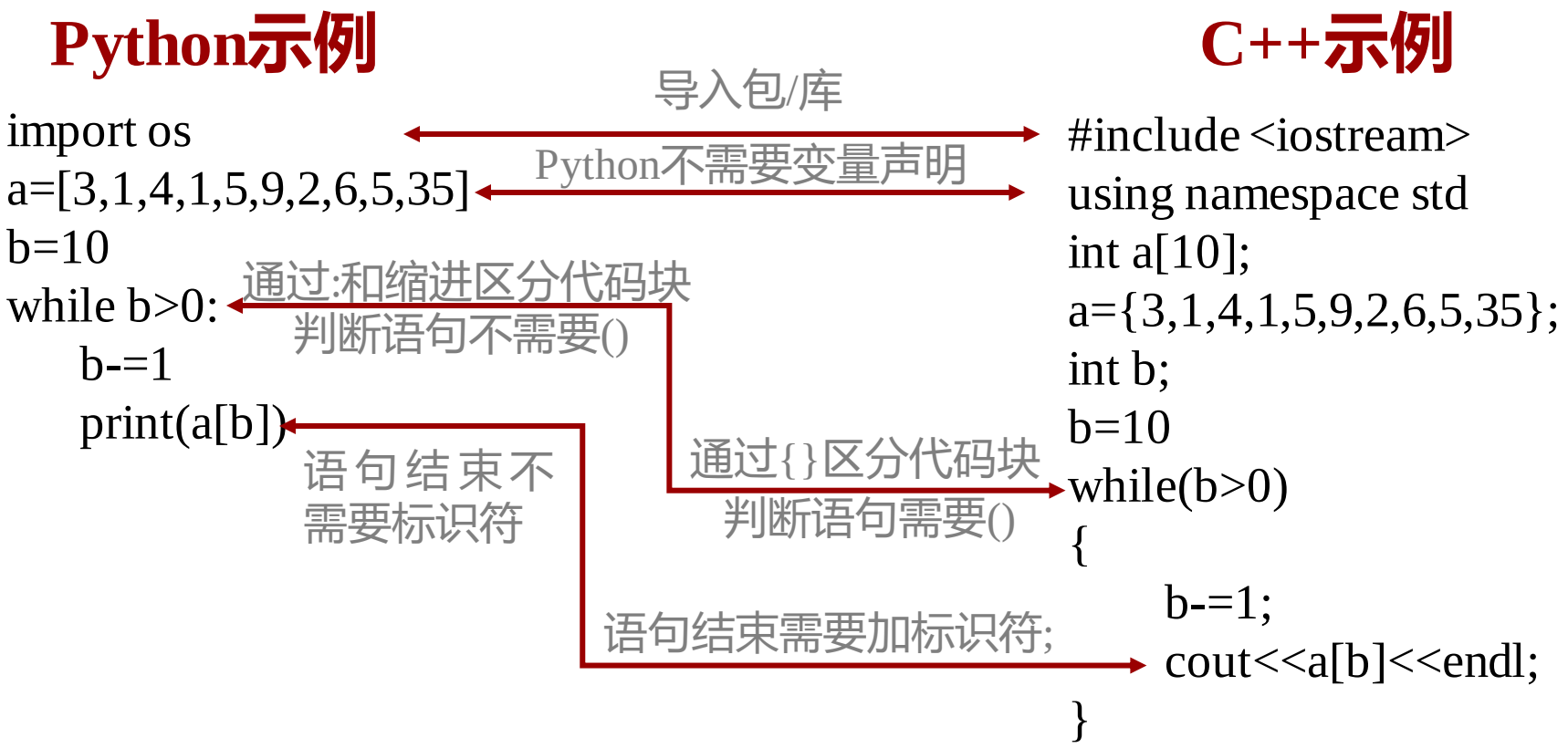
- ❖ Python的作者是荷兰人Guido Van Rossum。
- ❖ Python的设想酝酿于19世纪80年代末并于1989年12月开始实现。
- ❖ 1991年-Python0.9.0问世标志着Python正式诞生。
- ❖ 2000 年 -Python2.0 发 布 ， 2020 年 Python 2.7.8发布后Python2.x停止更新。
- ❖ 2008年-Python3.0发布， Python3.x是目前最流行的版本，本课程编程项目均基于**Python3.11.9**实现。



*图片来源：https://en.wikipedia.org/wiki/Guido_van_Rossum

Python和C++

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和C++

北京大学 PEKING UNIVERSITY

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

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

数字类型及运算符

北京大学 PEKING UNIVERSITY

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

北京大学 PEKING UNIVERSITY

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、小于等子、大于、大于等子、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

北京大学 PEKING UNIVERSITY

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

北京大学 PEKING UNIVERSITY

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

北京大学 PEKING UNIVERSITY

❖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
```

变量与赋值

北京大学 PEKING UNIVERSITY

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'。

判断与循环

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')                #input为获取键盘输入的关键字
a=int(a)                                    #数据类型转换
if a>0:                                     #判断语句
    print(‘a为正数’)                        #当满足a>0时输出‘a为正数’
elif a<0:                                   #若前一个if不成立，执行判断
    print(‘a为负数’)                        #当满足a<0时输出‘a为负数’
else:                                       #若前面的if及elif均不成立，执行下方语句
    print('a为0')                           #输出‘a为0’
```

判断与循环

北京大学 PEKING UNIVERSITY

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
```

函数与缩进

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
```

函数与缩进

北京大学 PEKING UNIVERSITY

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

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

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]])
```

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

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],#注：切片索引中0可省略， a[1:3,:2]等同于a[1:3][:2]
       [-0.10321885,  0.4105985 ]])
```

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

Numpy介绍

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]])
```

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

Numpy介绍

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]])
```

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

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：

dot(a, b)[i,j,k,m] = 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]。

[1]https://www.runoob.com/numpy/numpy-mathematical-functions.html

Numpy介绍

北京大学 PEKING UNIVERSITY

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 '/'
```

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

Numpy介绍

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×n×q…=M，其中M为原始数组所包含的数据元素个数**

**推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html*

Numpy介绍

北京大学 PEKING UNIVERSITY

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*numpy.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)
```

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

Python基础

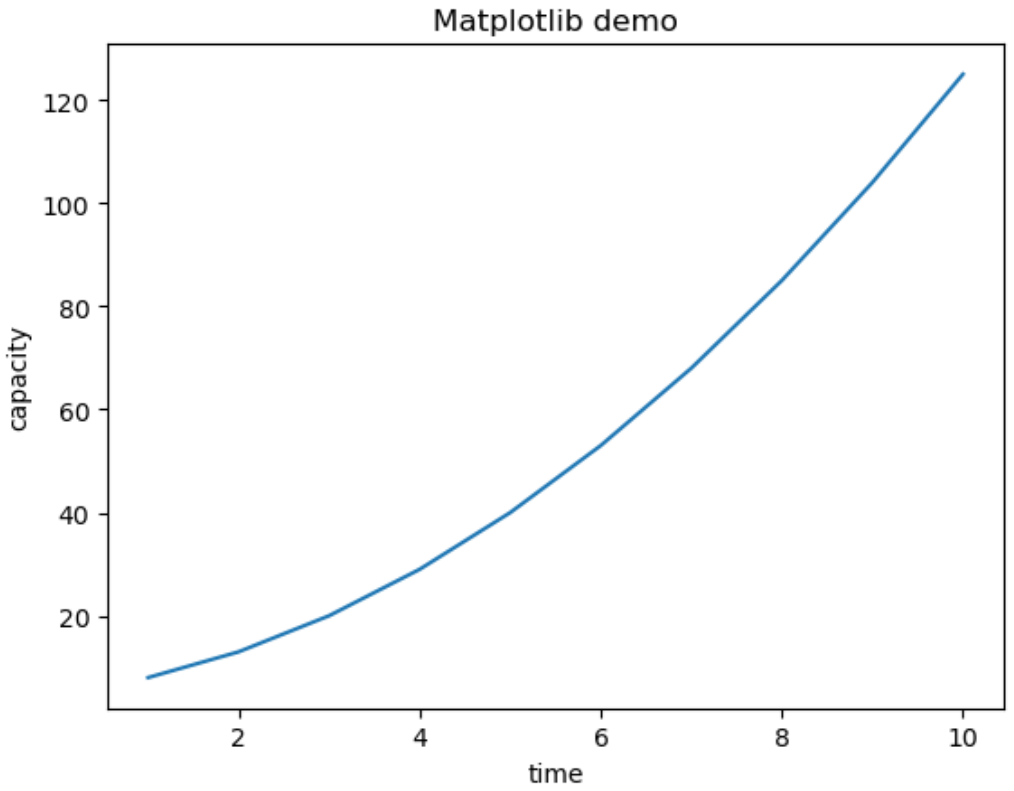
北京大学 PEKING UNIVERSITY

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

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 n
import matplotlib.p
x = np.array([1,2,3,
y = x**2+2 * x + 5
plt.title("Matplotlib
plt.xlabel("time")
plt.ylabel("capacity
plt.plot(x,y)
plt.show()
```



包或模块

入对象重命名

轴变量绘制曲线

*推荐链接：https://www.runoob.com/numpy/numpy-mathematical-functions.html

Python基础

北京大学 PEKING UNIVERSITY

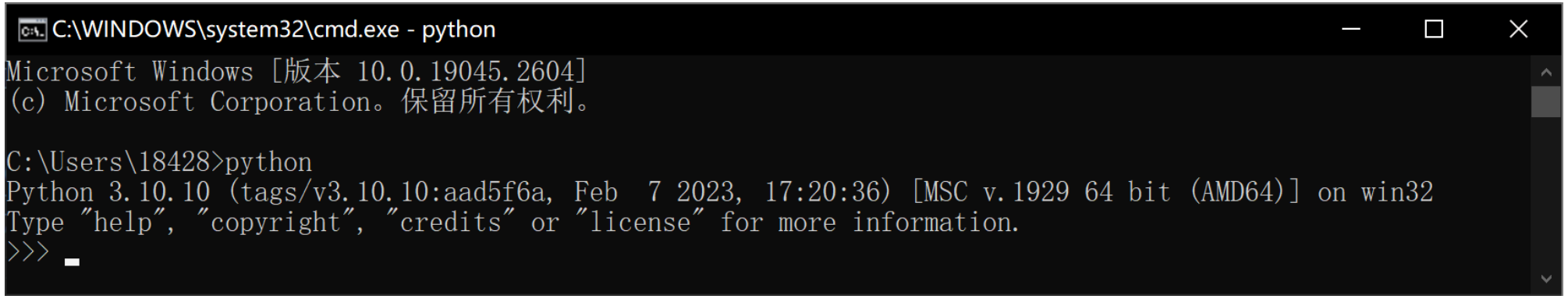
1. Python简介
2. Python语法
3. Numpy库介绍
4. Matplotlib库介绍
5. Python、Numpy、Matplotlib安装
6. Python项目实例

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





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



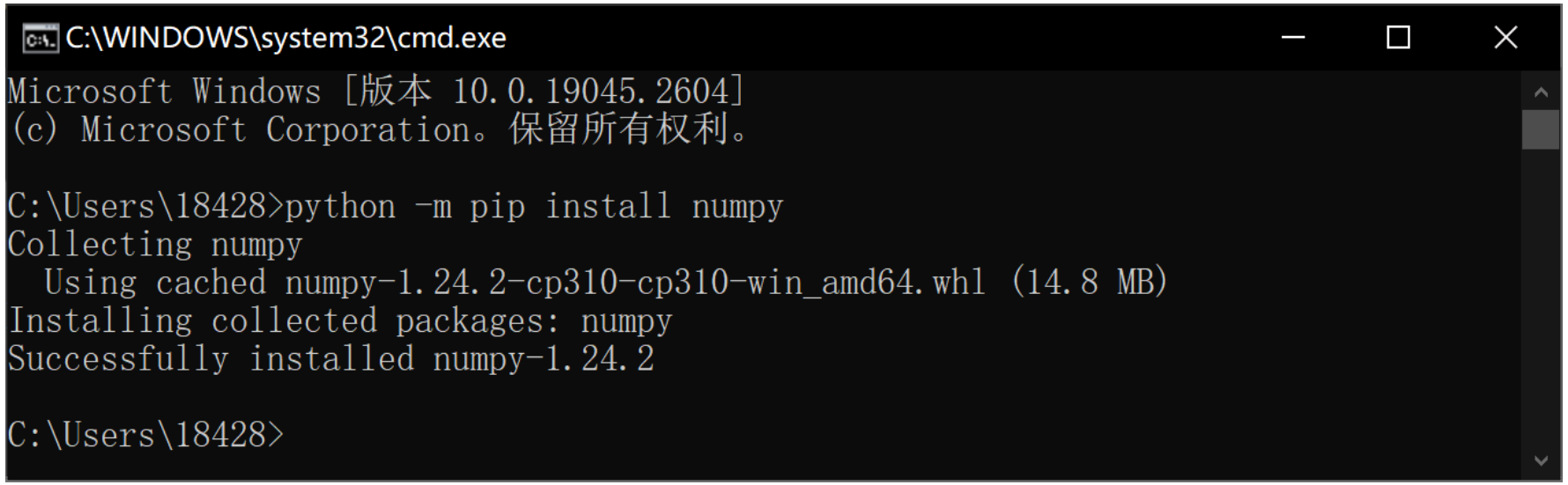


Python安装

③Numpy安装

打开命令行输入：

python -m pip install numpy

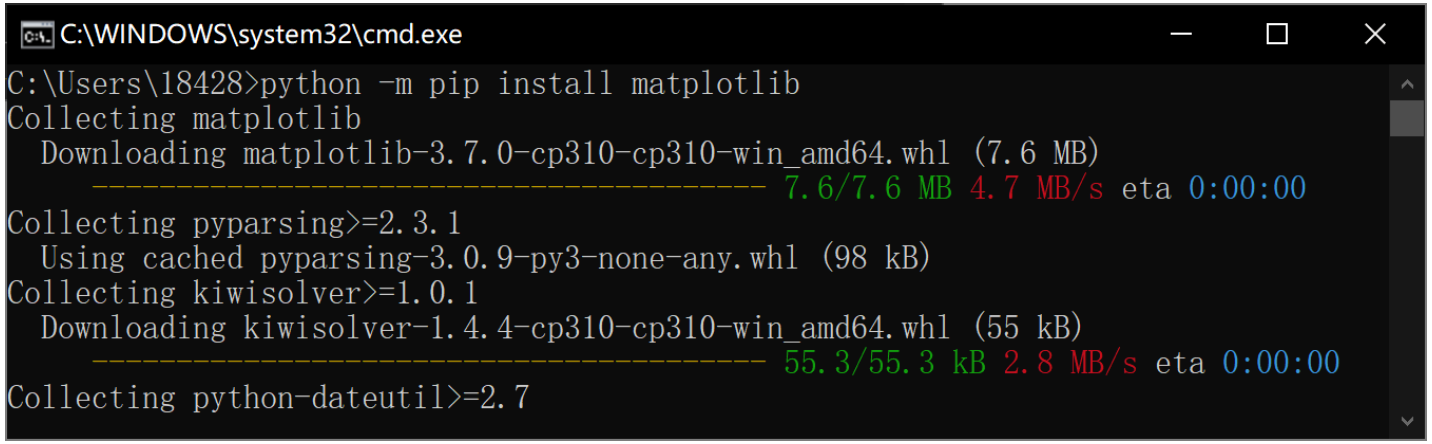


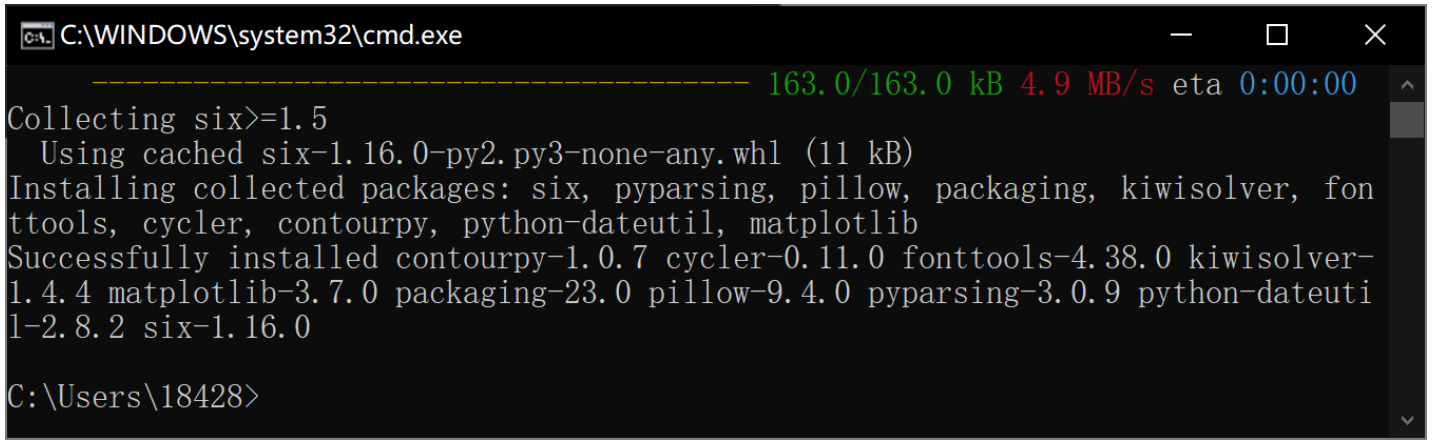
Python安装

④Matplotlib安装

打开命令行输入：

python -m pip install matplotlib





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

Python基础

北京大学 PEKING UNIVERSITY

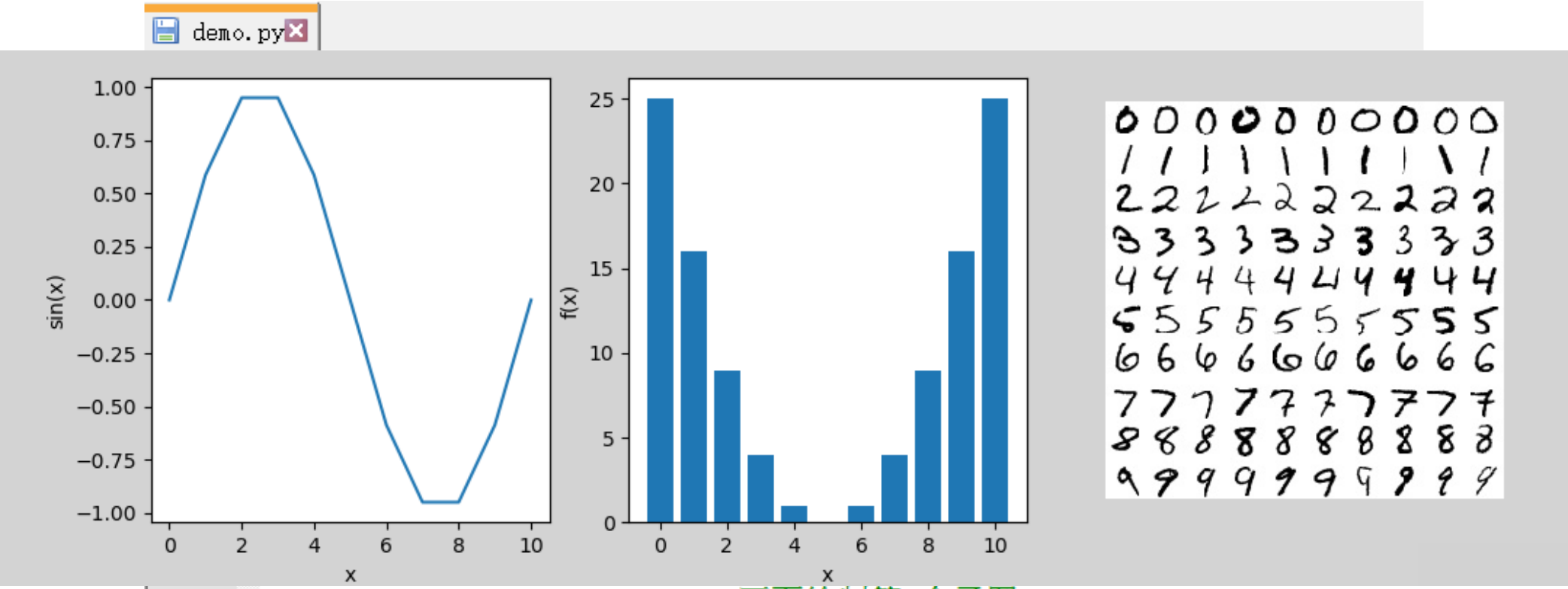
1. Python简介
2. Python语法
3. Numpy库介绍
4. Matplotlib库介绍
5. Python、Numpy、Matplotlib安装
6. Python项目实例

Python项目实例

```
import numpy as np                          #导入numpy
import matplotlib.pyplot as plt             #导入matplotlib
x=np.array([0,1,2,3,4,5,6,7,8,9,10])        #创建数组
y_sin=np.sin(x*36*np.pi/180)                #计算正弦值
y_f=x**2-10*x+25                            #计算二次函数值
img=plt.imread('minist.jpg')                #读取图像
fig=plt.figure(figsize=(12,4))              #设置绘图大小
fig.patch.set_facecolor('lightgrey')        #设置绘图背景
plt.subplot(131)                            #绘图区域分成1行3列，下面绘制第1个子图
plt.xlabel('x')                             #设置横坐标标题
plt.ylabel('sin(x)')                        #设置纵坐标标题
plt.plot(x,y_sin)                           #绘制正弦曲线
plt.subplot(132)                            #下面绘制第2个子图
plt.xlabel('x')                             #设置横坐标标题
plt.ylabel('f(x)')                          #设置纵坐标标题
plt.bar(x,y_f)                              #绘制柱状图
plt.subplot(133)                            #下面绘制第3个子图
plt.axis('off')                             #设置不显示坐标轴
plt.imshow(img,cmap='gray’)                 #以灰度图显示图片
plt.show()                                  #显示绘图结果
```

Python项目实例

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




```
17  plt.subplot(133)                #下面绘制第3个子图
18  plt.axis('off')                 #设置不显示坐标轴
19  plt.imshow(img,cmap='gray')  #以灰度图显示图片
20  plt.show()                  #显示绘图结果
21
```

运行结果

谢谢

北京大学
PEKING UNIVERSITY

