

第 0 次课 课程简介和 Python 快速预览

Python 科学计算

周吕文

宁波大学，机械工程与力学学院

2024 年 9 月 1 日



提要

1 课程简介

2 编程简介

3 快速预览

课程简介

课程目标

- 通掌握基本的编程技能，能够使用 Python 编写简单的程序
- 学会利用程序语言去建模，解决一些科学工程中的实际问题

课程内容

- 公式的计算
- 循环与列表
- 函数与分支
- 用户输入和错误管理
- 数组计算和曲线绘图
- 字典和字符串
- 类简介
- 面向对象编程
- 随机数和简单的游戏
- 数列和差分方程
- 微分方程（组）数值解
- 人工神经网络

成绩构成

- 20% 出勤 + 20% 作业 + 60% 期末考试（上机）

参考教材

- 英文：Hans Petter Langtangen. A Primer on Scientific Programming with Python (5th Edition)
- 译本：科学计算基础编程 – Python 版（第五版），清华大学出版社

公众号：数学模型



课程资源：公众号对话框内回复“python”

- 幻灯片：slides/%d.pdf
- 源代码：slides/%d/*.py
- 参考书：ebooks/*.pdf
- 作业：problems/%d.pdf
- 解答：solutions/%d.pdf
solutions/%d/ex%2d_%d.py

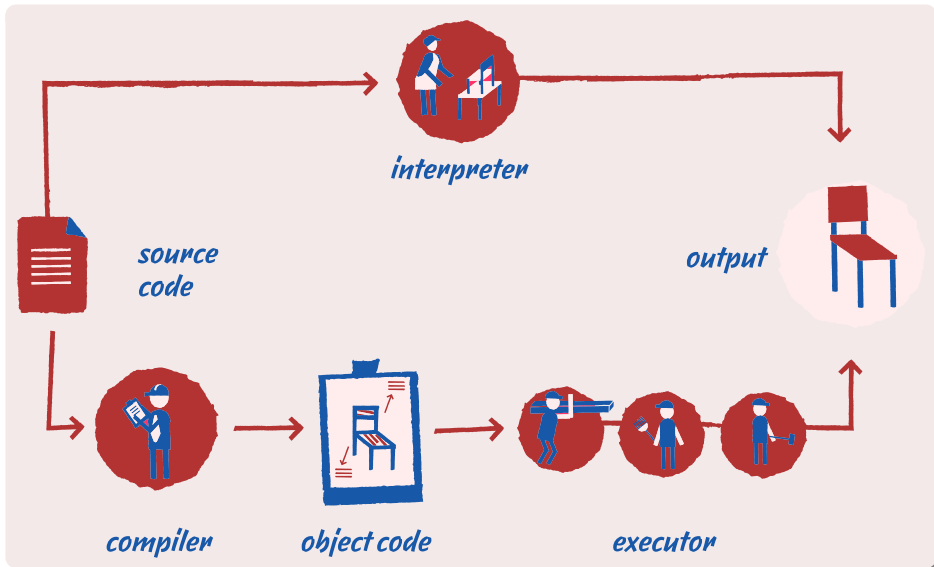
提要

1 课程简介

2 编程简介

3 快速预览

静态语言（编译执行）和脚本语言（解释执行）



Python 语言介绍

起源

- Python 英文单词的意思是“蟒蛇”
- 在 1989 年末，由 Guido 开发，作为 ABC 的继承
- ABC 是由 Guido 参加的为非专业程序员设计的教学语言
- Guido 是英国幽默剧团“Monty Python”的粉丝



优势

- 全能：数据分析，系统运维、网页编程、机器学习、人工智能等
- 简单：语法简洁明了，易于阅读和学习
- 通用：能够把用其他语言制作的各种模块很轻松地联结在一起
- 易用：全模块化，拥有大量用于数学建模的库
- 开源：良好的生态圈，适配各种平台，主流语言

Python 和第三方库的安装和使用

Python 的安装和使用

- 下载安装: [官网](#)、[Anaconda](#)
- 本地运行: IDLE、Visual Studio Code、PyCharm、Spyder、Eclipse
- 在线运行: 已安装了 numpy 和 matplotlib 的 [Python 在线编译器](#)

第三方库的安装和使用

- 安装: 在终端中输入 `pip install 库名`, 例如: `pip install numpy`
- 使用: `import numpy` 或 `import numpy as np`

提要

1 课程简介

2 编程简介

3 快速预览

公式计算：温度转换公式

温度转换公式

$$F = \frac{9}{5}C + 32 \quad \Longleftrightarrow \quad C = \frac{5}{9}(F - 32)$$

如果 $C = -20$, 求 F

```
>>> C = -20
>>> F = 9/5*C + 32
>>> print("C =", C, ", F =", F)
C = -20 , F = -4.0
```

如果 $F = 5$, 求 C

```
>>> F = 5
>>> C = 5/9*(F - 32)
>>> print("F =", F, ", C =", C)
F = 5 , C = -15.0
```

公式计算：使用标准数学库

当 $A = 8$ 时，计算 $r = \sqrt{A/\pi}$

```
>>> import math
>>> A = 8
>>> r = math.sqrt(A/math.pi)
>>> print(r)
1.5957691216057308
```

当 $x = 1.2$ 时，计算 $Q = \sin(\pi/3) \cos x + 4 \ln x$

```
>>> from math import sin, cos, pi, log
>>> x = 1.2
>>> Q = sin(pi/3)*cos(x) + 4*log(x)
>>> print(Q)
1.0430972478109022
```

1 撑杆跳的极限

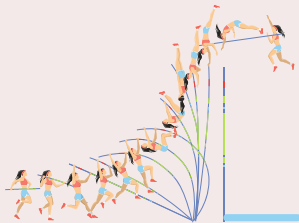
文件名: pole_vault.py

撑杆跳的过程主要是运动员动能转化为重力势能的过程:

$$\frac{1}{2}mv^2 = mgH$$

考虑运动员重心起始高度, 撑杆跳的极限成绩为:

$$H = \frac{v^2}{2g} + r \cdot h$$



其中 v 为助跑速度, h 为身高, r 为重心高占身高比例。根据以上公式估计奥运冠军杜普兰蒂斯撑杆跳高极限成绩:

- 杜普兰蒂斯的助跑速度 $v = 10 \text{ m/s}$, 身高 $h = 1.81 \text{ m}$
- 正常成年人重心占身高的比例 $r = 0.55$; 重力加速度 $g = 9.8 \text{ m/s}^2$

数据类型：数字

加减乘除

```
>>> 2 + 2
```

```
4
```

```
>>> 50 - 5*6
```

```
20
```

```
>>> 8 / 5
```

```
1.6
```

```
>>> (50 - 5*6) / 4
```

```
5.0
```

整数商、余数和乘方

```
>>> 17 // 3
```

```
5
```

```
>>> 17 % 3
```

```
2
```

```
>>> 5 ** 2
```

```
25
```

```
>>> 4 ** 0.5
```

```
2.0
```

数字类型

```
>>> type(3)
```

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

```
>>> type(3.0)
```

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

数据类型：文本

文本由 `str` 类型表示，称为“字符串”

```
>>> 'spam eggs'
'spam eggs'
>>> "Paris rabbit got your back :)! Yay!"
'Paris rabbit got your back :)! Yay!'
>>> type('1975')
<class 'str'>
```

要标示引号本身，需要对其进行“转义”，或使用不同类型的引号

```
>>> 'doesn\'t'
"doesn't"
>>> "doesn't"
"doesn't"
>>> "\"Yes,\" they said."
'"Yes," they said.'
```

数据类型：文本

字符串定义和输出字符串看起来可能不同

```
>>> s = 'First line.\nSecond line.'
>>> s
'First line.\nSecond line.'
>>> print(s)
First line.
Second line.
```

字符串可以用 + 合并（粘到一起）

```
>>> 3 * 'un' + 'ium'
'unununium'
```

数据类型：文本

字符串支持 索引（下标访问），第一个字符的索引是 0

```
>>> word = 'Python'
>>> word[0]
'P'
>>> word[5]
'n'
```

索引还支持负数，用负数索引时，从右边开始计数

```
>>> word[-1]
'n'
>>> word[-2]
'o'
```


数据类型：文本

字符串支持切片，以获取子字符串

```
>>> word = 'Python'
>>> word[0:2]
'Py'
>>> word[2:5]
'tho'
```

省略开始索引，默认为 0，省略结束索引，默认为到字符串结尾

```
>>> word[:2]
'Py'
>>> word[4:]
'on'
>>> word[-2:]
'on'
```

数据类型：列表

列表可将不同值组合，并和字符串一样支持索引和切片

```
>>> squares = [1, 4, 9, 16, 25]
```

```
>>> squares[0]
```

```
1
```

```
>>> squares[-1]
```

```
25
```

```
>>> squares[-3:]
```

```
[9, 16, 25]
```

列表还支持合并操作

```
>>> squares + [36, 49, 64, 81, 100]
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

数据类型：列表

列表的内容可以改变

```
>>> cubes = [1, 8, 27, 65, 125]
>>> cubes[3] = 64
>>> cubes
[1, 8, 27, 64, 125]
```

列表的相关方法

<pre>>>> cubes.append(27) >>> cubes [1, 8, 27, 64, 125, 27] >>> cubes.sort() >>> cubes [1, 8, 27, 27, 64, 125] >>> cubes.index(8) 1</pre>	<pre>>>> cubes.reverse() >>> cubes [125, 64, 27, 27, 8, 1] >>> cubes.count(27) 2 >>> cubes.remove(64) >>> cubes [125, 27, 27, 8, 1]</pre>
--	--

数据类型：列表

列表可以嵌套

```
>>> s = [[1, 2], 3, 'a', ['math', 'models']]
>>> s[0]
[1, 2]
>>> s[1]
3
>>> s[0][0]
1
>>> s[3][1]
'models'
```

2 姓名字符串

文件名: name_str.py

- 将你的姓和名分别定义成字符串，并输出
- 交换这两个字符串的前两个字符，并输出
- 合并这两个字符串并用空格隔开，并输出

3 数字列表操作

文件名: list_nums.py

- 定义列表 A 和 B，分别包含 10 以内所有奇数和偶数
- 将 A 和 B 拼接成一个列表 C
- 并取出 C 中的前六个数并定义为列表 D

结构编程：分支

if 语句

```
>>> x = 42
>>> if x < 0:
...     x = 0
...     print('Negative changed to zero')
... elif x == 0:
...     print('Zero')
... elif x == 1:
...     print('Single')
... else:
...     print('More')
...
More
```

结构编程：循环

for 语句

```
>>> words = ['cat', 'window', 'defenestrate']
>>> for w in words:
...     print(w, len(w))
...
cat 3
window 6
defenestrate 12
```

range() 函数

```
>>> for i in range(3):
...     print(i)
...
0
1
2

>>> range(4)
range(0, 4)
>>> list(range(6))
[0, 1, 2, 3, 4, 5]
>>> sum(range(6))
15
```

两圆关系：定义函数 CircCirc

- 输入：圆 C1 的心 (x_1, y_1) 和半径 r_1 ；圆 C2 的心 (x_2, y_2) 和半径 r_2
- 返回：如果 C2 在 C1 中，则输出“C2 在 C1 中”；
如果 C1 在 C2 中，则输出“C1 在 C2 中”；
如果 C1 与 C2 相交，则输出“C1 与 C2 相交”；
如果 C1 与 C2 分离，则输出“C1 与 C2 相离”
- 实例：输入 $(0,0), 1, (1,1), 1$ ，输出“C1 与 C2 相交”

求偶数和：定义函数 SumOdd

- 输入：一个列表或元组
- 返回：列表中所有偶数构成的列表，以及该这些偶数的和
- 实例：输入 $[1, 2, 3, 4, 5]$ ，返回 $[2, 4]$ 和 6

数组计算：数组定义

一维数组

```
>>> import numpy as np    # 需要安装 numpy 库
>>> a = np.arange(6)
>>> a
array([0, 1, 2, 3, 4, 5])
>>> type(a)
<class 'numpy.ndarray'>
>>> b = np.array([1, 2, 3, 4, 5, 6])
>>> b
array([1, 2, 3, 4, 5, 6])
>>> b[1]
2
>>> np.zeros(2)
array([0., 0.])
>>> np.ones(2)
array([1., 1.])
```

数组计算：数组定义

二维数组

```
>>> c = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
>>> c
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
>>> c[0]
array([1, 2, 3, 4])
>>> c[1][0]
5
>>> c[1,0]
5
>>> c[1,0:3]
array([5, 6, 7])
```

数组计算：数组的简单运算

加减乘除

```
>>> a = np.array([1, 2, 3, 4])
>>> b = np.array([5, 6, 7, 8])
>>> c = a + b
>>> c
array([ 6,  8, 10, 12])
>>> d = b - a
>>> d
array([4, 4, 4, 4])
>>> e = a * b
>>> e
array([ 5, 12, 21, 32])
>>> f = a / b
>>> f
array([0.2, 0.33333333, 0.42857143, 0.5])
```

数组计算：数组的统计函数

求和、求平均、求极值、求最小值的索引

```
>>> a = np.array([3, 2, 1, 6, 8])
```

```
>>> s = np.sum(a)
```

```
>>> s
```

```
20
```

```
>>> m = np.mean(a)
```

```
>>> m
```

```
4.0
```

```
>>> maxval = np.max(a)
```

```
>>> maxval
```

```
8
```

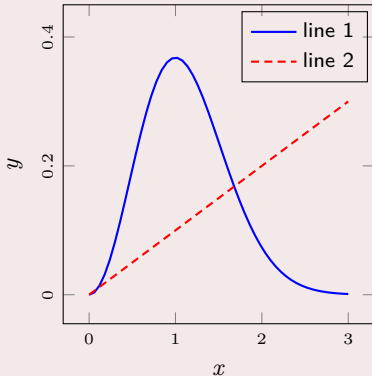
```
>>> minind = np.argmin(a)
```

```
>>> minind
```

```
2
```

绘图：曲线图

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 3, 51)
y1 = x**2*np.exp(-x**2)
y2 = x/10
plt.plot(x, y1)
plt.plot(x, y2, 'r--')
plt.legend(["line 1", "line 2"])
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



4 心心相印

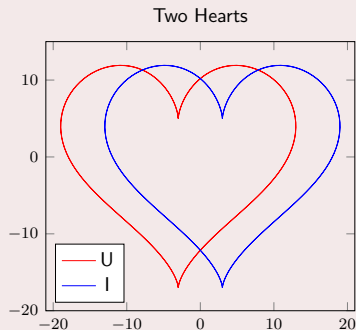
文件名: two_hearts.py

以下两式是心形曲线的参数方程:

$$x = 16 \sin^3 \theta$$

$$y = 13 \cos \theta - 5 \cos(2\theta) - 2 \cos(3\theta) - \cos(4\theta)$$

其中 $\theta \in [0, 4\pi]$, 请绘制右图所示的两颗心。



The End!