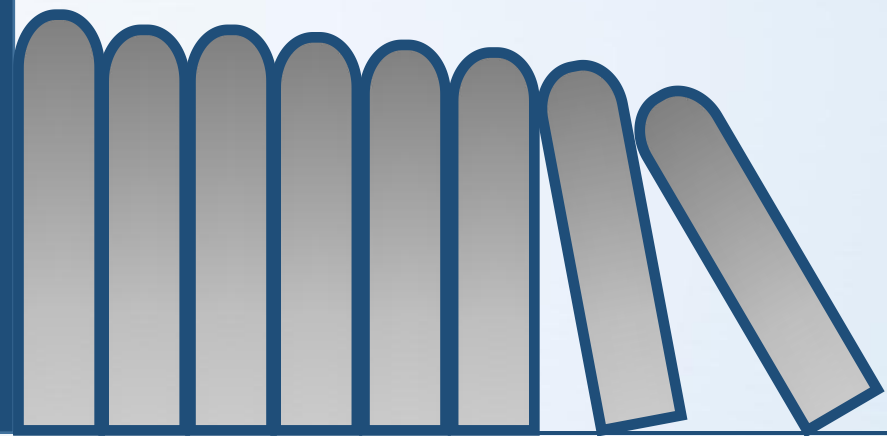




第一章 Python语言简介

苏湘宁

邮箱: suxn@hainanu.edu.cn



Python语言简介






- 为什么学习Python?
- Python语言是什么?
- Python语法简介
- 课后练习



1.1 本课程选择Python作为课程教学编程语言的原因:

Python 语言属于全场景编程语言,在 web 开发、人工智能、数据分析、游戏开发、网络爬虫等领域都有很好的应用。

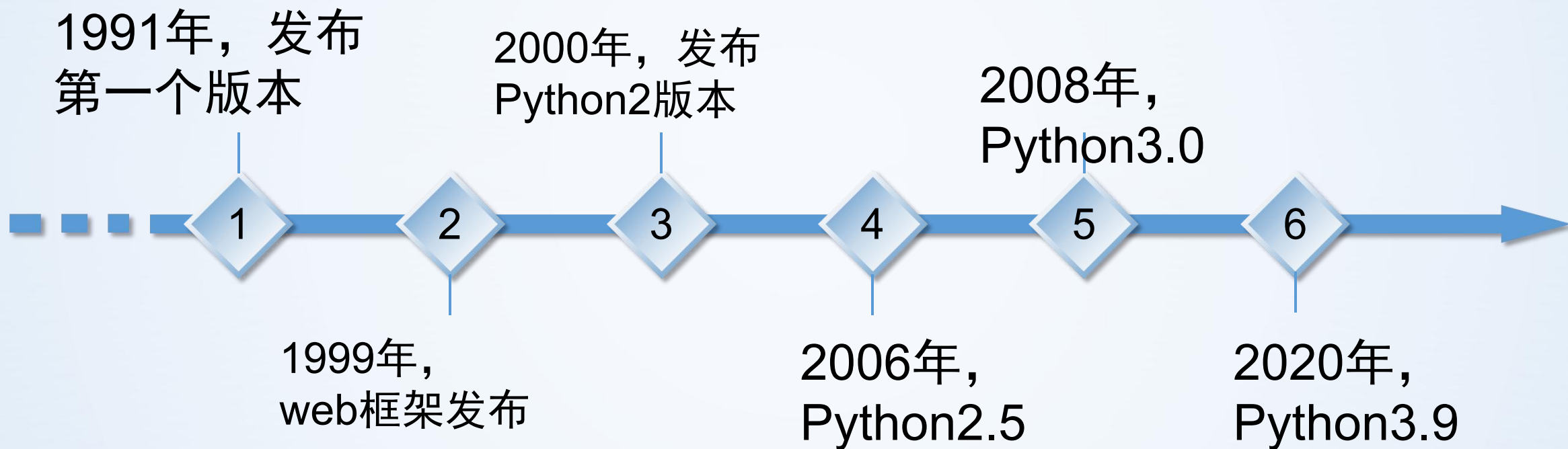
TIOBE 12月Top15编程语言:

Dec 2022	Dec 2021	Change	Programming Language	Ratings	Change
1	1		 Python	16.66%	+3.76%
2	2		 C	16.56%	+4.77%
3	4	▲	 C++	11.94%	+4.21%
4	3	▼	 Java	11.82%	+1.70%
5	5		 C#	4.92%	-1.48%



1.2 Python语言是什么

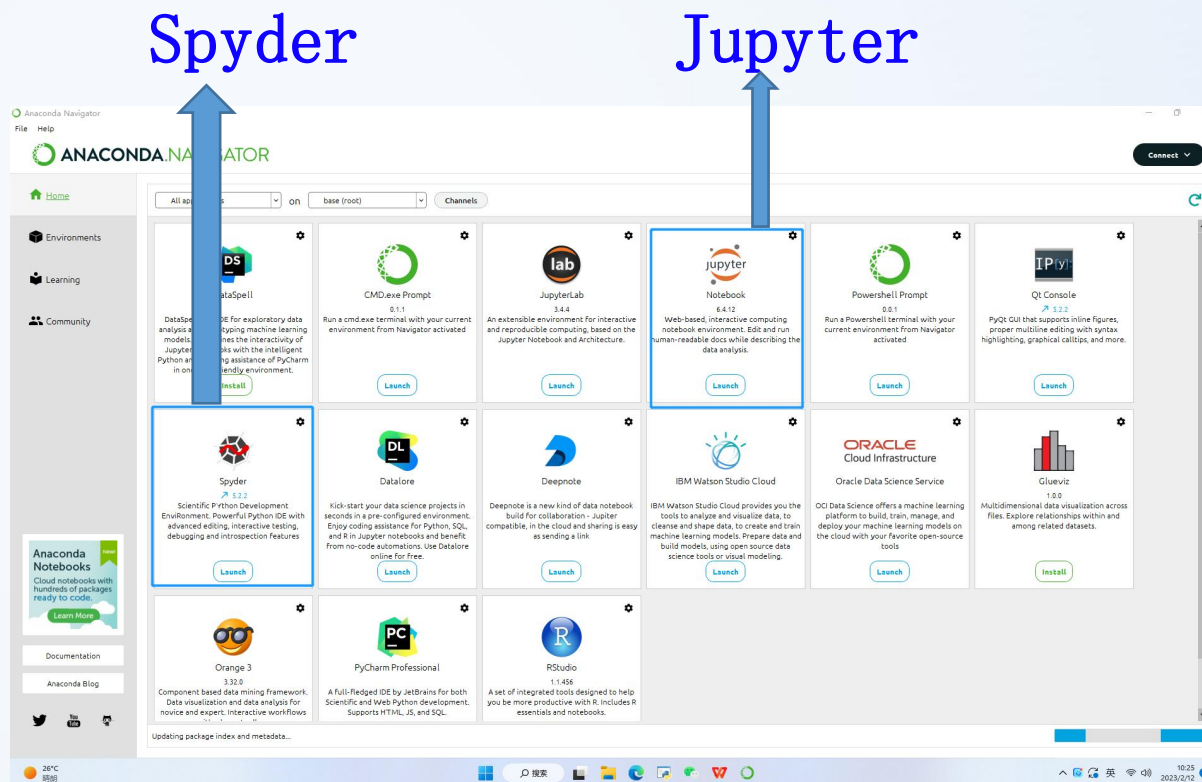
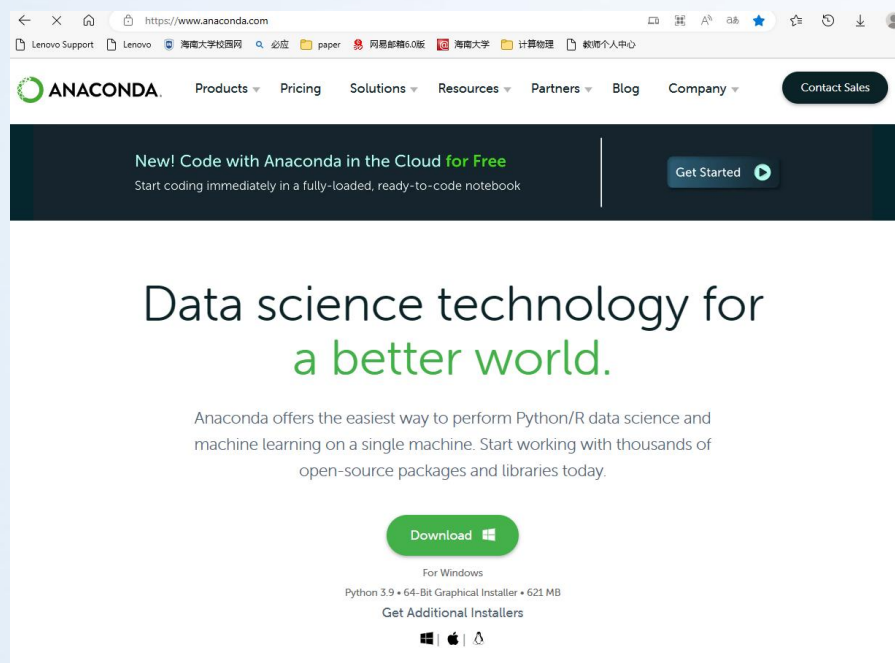
Python是一种解释性脚本语言，由吉姆·范·罗苏姆使用C开发





1.2 Python语言是什么

Python安装: <https://www.anaconda.com/download#downloads>
<https://www.anaconda.com/>

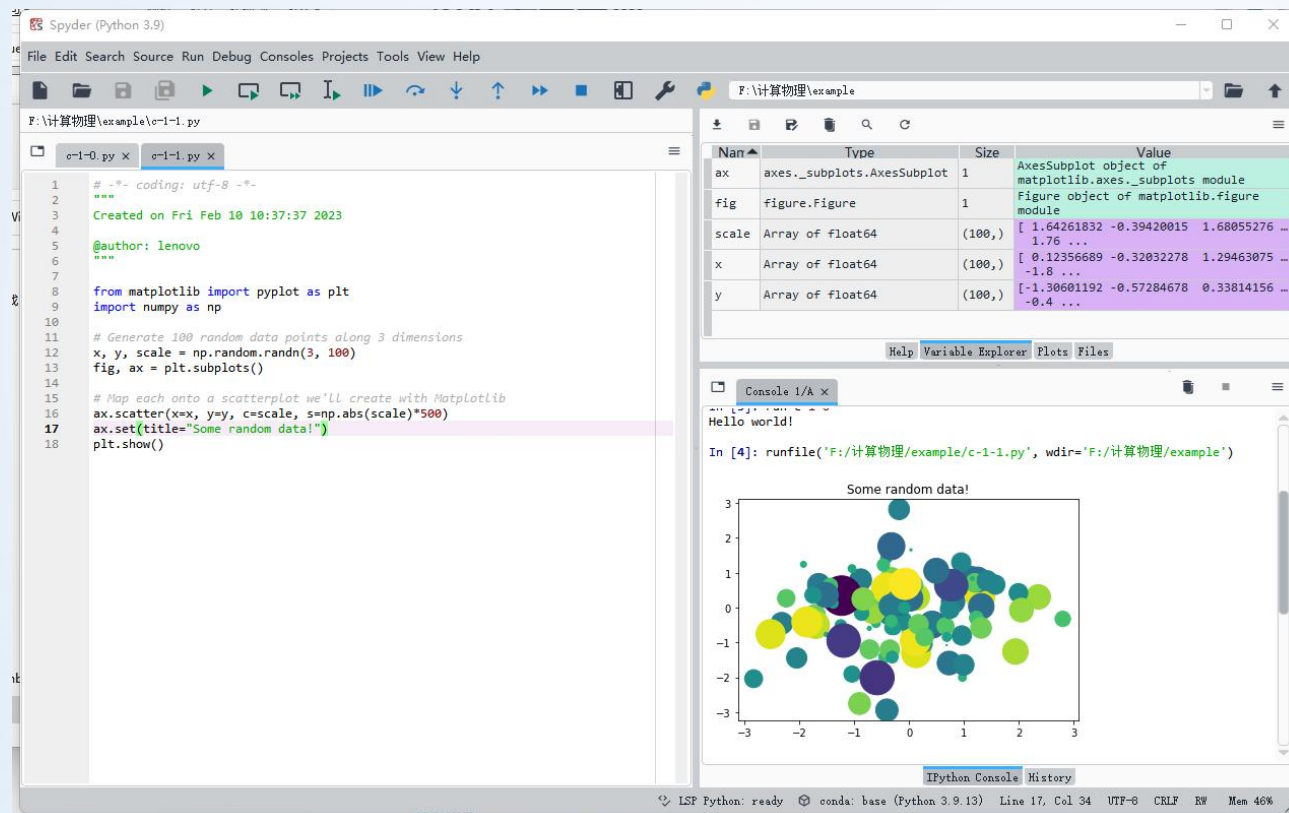





1.3 Python语法简介 Python的执行方式:

- 在Python的IDE工具下运行。

例如：在Spyder环境下

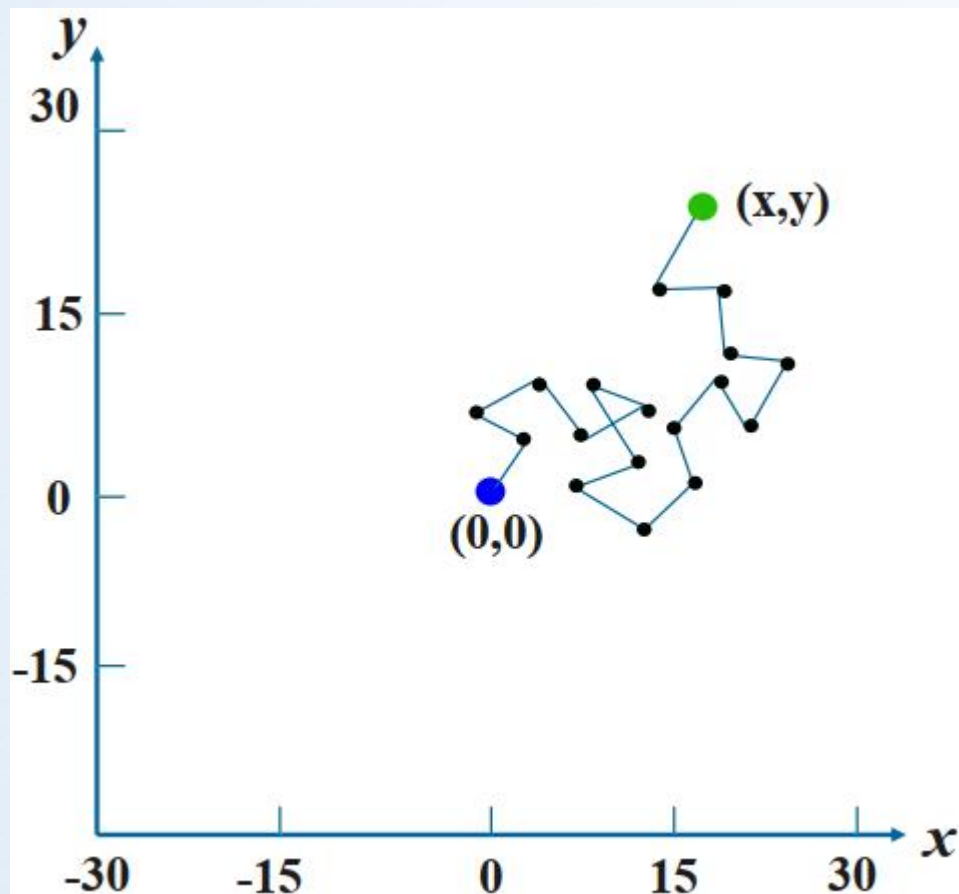


- 左侧编辑窗口可以直接创建、编辑程序。
- 右下角的控制台窗口，可以运行程序，命令为“run code-1-1.py”
- 也可以点击顶部的“run”按钮 ，或者快捷键F5



1.3 Python语法简介 例子：随机游走 (random walk)

- 从坐标原点 $(0, 0)$ 出发的例子，每一步都以固定步长在二维平面上随机运动，用蒙特卡洛方法，模拟粒子随机游走过程，画出游走轨迹。



```
Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\计算物理\example\c-1-2.py

c-1-0.py x c-1-2.py x

7
8
9 #random walk
10 #@author: xns
11
12 import numpy as np
13 import matplotlib.pyplot as plt
14 import random
15
16
17 line = input("input the maximal number of walk steps:")
18 nstep = int(line)
19 print("the maximal number of walk steps is:", nstep)
20
21 if nstep > 10000:
22     print('too many steps')
23     exit()
24 elif nstep < 10:
25     print('too few steps')
26     exit()
27
28 xlist = [] # create empty list
29 ylist = []
30
31 x = 0.0
32 y = 0.0
33 xlist.append(x) # append the initial x value to the xlist.
34 ylist.append(y) # append the initial y value to the ylist.
35
36 random.seed(913) #random number seed.
37 outfile = open('walk-traj.dat', 'w')
38
39
40 for i in range(nstep):
41     dx = 2*random.random()-1 # random number(-1.0,1.0)
42     dy = 2*random.random()-1
43
44     dr = np.sqrt(dx**2+dy**2)
45     dx = dx/dr #normalization
46     dy = dy/dr
47
48     x = x + dx # updating x position
49     y = y + dy
50
51     outfile.write("%d\t%.2f\t%.2f\n" % (i, x, y))
52
53     xlist.append(x)
54     ylist.append(y)
55
56 outfile.close()
57
58 plt.plot(xlist[:], ylist[:], 'r-', lw=1) #visualization
59 plt.xlabel('x-axis')
60 plt.ylabel('y-axis')
61 plt.show()
62
```

Name	Type	Size	Value
a	list	3	[1, 2, 3]
b	list	3	[4, 5, 6]
c	list	7	[1, 2, 3, 4, 5, 6, 10]
d	list	4	[3, 4, 5, 6]
dr	float64	1	0.7654114404754689
dx	float64	1	0.9827755622109328
dy	float64	1	0.18480312313644706
i	int	1	99
line	str	3	100
nstep	int	1	100
outfile	TextIOWrapper	1	TextIOWrapper object of _io module

```
Console 1/A x

x-axis

In [27]: runfile('F:/计算物理/example/c-1-2.py', wdir='F:/计算物理/example')

input the maximal number of walk steps:100
the maximal number of walk steps is: 100

2
0
-2
-4
-6
-8
-2  -1  0  1  2  3  4  5
x-axis

In [28]:

Python Console History
LSP Python: ready conda: base (Python 3.9.13) Line 52, Col 5 UTF-8 CRLF EN Mem 49%
```



1.3 Python语法简介

• 1.3.1 注释语句

多行注释：三个引号

单行注释：在语句前加 # 号

```
"""
random walk
@author: xnS
"""
```

```
#random walk
#@author: xn5
```



1.3 Python语法简介:

- 1.3.2 外部函数库 调用时在程序开头加载相应的函数库

通过`import` 加载函数库`numpy`并命名为`np`

`numpy`库: 对Python数值计算功能的扩展, 可用于处理数组和矩阵等类型的数据

`matplotlib`库: 绘图函数库 (为了节约内存可以只加载函数库的某模块)

`random`库: 随机数生成函数库

```
import numpy as np
import matplotlib.pyplot as plt
import random
```



1.3 Python语法简介:

• 1.3.3 标准输入输出

input ()：输入 默认字符串类型

print () : 输出

类型转换函数: int ()

```
line = input("input the maximal nubmer of walk steps:")
nstep = int(line)
print("the maximal number of walk steps is:", nstep)
```

```
input the maximal nubmer of walk steps:100
the maximal number of walk steps is: 100
```



1.3 Python语法简介:

- 1.3.4 数据类型，赋值语句，数据转换

数字 (number) 类型	int, long, float, complex
字符串 (Strings) 类型	str
列表 (List) 类型	list
元组 (Tuple) 类型	tuple
字典 (dictionary) 类型	dict

```
In [18]: a=[1,2,3]
```

```
In [19]: b=[4,5,6]
```

```
In [20]: c=a+b
```

In [21]: c

```
Out[21]: [1, 2, 3, 4, 5, 6]
```

列表: `c[M:N]` 引用第m到n-一个元素; `append`: 追加

```
In [21]: c
Out[21]: [1, 2, 3, 4, 5, 6]
```

```
In [22]: d=c[2:6]
```

```
In [23]: d
Out[23]: [3, 4, 5, 6]
```

```
In [24]: c.append(10)
```

```
In [25]: c
Out[25]: [1, 2, 3, 4, 5, 6, 10]
```

In [26]:



1.3 Python语法简介:

- 1.3.5 条件语句

if语句实现选择执行功能

Python通过缩进定义程序区域

```
if nstep > 10000:  
    print('too many steps')  
    exit()  
elif nstep < 10:  
    print('too few steps')  
    exit()
```



1.3 Python语法简介:

- 1.3.6 列表操作

xlist, ylist: 初始值设为空列表,

x, y: 浮点型变量

```
xlist = []    # create empty list
ylist = []

x = 0.0
y = 0.0
xlist.append(x) # append the initial x value to the xlist.
ylist.append(y) # append the initial y value to the ylist.
```




1.3 Python语法简介:

- 1.3.7 随机数产生

设置随机数种子: 如 `random.seed(913)`

`random.random()`: (0, 1) 范围内的均匀分布的随机数

`random.seed(None)`: 计算机时钟确定随机数种子

```
random.seed(913) #random number seed.
```




1.3 Python语法简介:

• 1.3.9 循环语句

对序列内的所有元素循环，每次循环都要执行缩进部分的所有执行语句

```
for i in range(nstep):  
    dx = 2*random.random()-1 # random number(-1.0,1.0)  
    dy = 2*random.random()-1  
  
    dr = np.sqrt(dx**2+dy**2)  
    dx = dx/dr #normalization  
    dy = dy/dr  
  
    x = x + dx # updating x position  
    y = y + dy  
  
    outfile.write('%d\t%.2f\t%.2f\n' %(i, x, y))  
  
    xlist.append(x)  
    ylist.append(y)
```

→ (-1.0, 0, 1.0) 均匀分布的随机数

→ 归一化处理

```
for 变量 in 序列:  
    执行语句 1  
    执行语句 2  
    .....
```



1.3 Python语法简介:

• 1.3.10 Matplotlib作图

`pl.plot()`:调用pylab中的绘图函数

`pl.show()`:将图形在窗口显示出来

访问官网[example](https://www.matplotlib.org.cn/gallery/)部分:

<https://www.matplotlib.org.cn/gallery/>

横坐标

纵坐标

红色实线

宽度

```
pl.plot(xlist[:], ylist[:], 'r-', lw=1) #visualization
pl.xlabel('x-axis')
pl.ylabel('y-axis')
pl.show()
```

```
pl.savefig('fig-1-2.png', bbox_inches='tight')
```



1.3 Python语法简介:

• 1.3.12 绘制3D图形

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
# 创建3d图形的两种方式
# ax = Axes3D(fig)
ax = fig.add_subplot(111, projection='3d')
# X, Y value
X = np.arange(-4,4,0.25)
Y = np.arange(-4,4,0.25)
X,Y = np.meshgrid(X,Y)#X-y 平面的网格
R = np.sqrt(X ** 2 + Y ** 2)
# height value
Z = np.sin(R)
```



1.3 Python语法简介:

• 1.3.11 绘制3D图形

```
X,Y = np.meshgrid(X,Y)#X-y 平面的网格
R = np.sqrt(X ** 2 + Y ** 2)
# height value
Z = np.sin(R)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.get_cmap('rainbow'))
# zdir : 'z'/'x'/'y' 示把等高线投射到哪个面
# offset : 表示等高线岛投射到指定页面的某个刻度
ax.contourf(X,Y,Z,zdir='z',offset=-2)
#设置z轴的显示范围, X、Y轴置方式相同
ax.set_zlim(-2,2)
plt.show()
```



A 3D surface plot of a function, likely $f(x,y) = x^2 - y^2$, showing a saddle point at the origin $(0,0,0)$. The surface is colored with a gradient from purple (low values) to red (high values). The x and y axes range from -4 to 4, and the z-axis ranges from -2 to 2.



1.3 Python语法简介:

• 1.3.11 绘制3D图形

```
from mpl_toolkits.mplot3d import axes3d
from matplotlib import cm
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
fig = plt.figure()
ax = fig.gca(projection='3d')
x, y, z = axes3d.get_test_data(0.05)
ax.plot_surface(x,y, z, rstride=8, cstride=8, alpha=0.5,color='b')
cset = ax.contour(x, y, z, zdir='z', offset=-106, cmap=cm.coolwarm)
cset = ax.contour(x, y, z, zdir='x', offset=-40, cmap=cm.coolwarm)
cset = ax.contour(x, y, z, zdir='y', offset=40, cmap=cm.coolwarm)
ax.set_xlabel('x')
ax.set_xlim(-40,40)
ax.set_ylabel('y')
ax.set_ylim(-40,40)
ax.set_zlabel('z')
ax.set_zlim(-108,188)
plt.show()
```

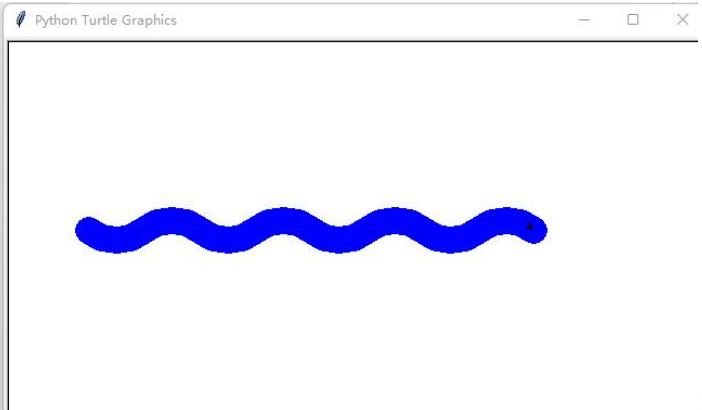
A 3D plot showing a surface $z = f(x, y)$ in a coordinate system with axes x , y , and z . The x and y axes range from -40 to 40, and the z axis ranges from -100 to 150. The surface is colored blue and purple, with a prominent peak near $(0, 0, 150)$. Contour lines are visible on the xy -plane, showing concentric circles around the origin and other features. The plot is rendered with a grid and a perspective view.



1.3 Python语法简介:

- 1.3.12* turtle作图

```
c-1-0.py x c-1-2.py x c-1-3.py x
1  # -*- coding: utf-8 -*-
2  """
3  Created on Sun Feb 12 14:28:16 2023
4
5  @author: lenovo
6  """
7
8  import turtle as tur
9
10 tur.setup(650,350,200,200)
11 tur.penup()
12 tur.fd(-250)
13 tur.pendown()
14 tur.pensize(25)
15 tur.pencolor("blue")
16 tur.seth(-40)
17
18 for i in range(4):
19     tur.circle(40,80)
20     tur.circle(-40,80)
21
22
23 tur.done()
```

The image shows a Python Turtle Graphics window titled "Python Turtle Graphics". Inside the window, a blue wavy line is drawn on a white background. The line starts at the left edge and moves horizontally to the right, with four distinct humps and four distinct valleys, creating a continuous wave pattern. The line is thick and blue.



1.4 课后练习:

振动的合成模拟^[1]:

简谐振动: $x = A\sin(\omega t + \varphi_0)$

题目: 相互垂直的两个方向x, y的振动如下:

$$x = 3.0\sin(3.0t + 0.0)$$

$$y = 3.0\sin(3.0t + 0.3)$$

请编写python程序, 1) 在t-x, t-y, x-y平面画出振动轨迹;

2) 改变 A , ω , φ_0 的取值, 观察振动轨迹如何变化, 思考为什么?

[1]何云存, 任涛. 基于Vpython的简谐振动合成可视化研究[J]. 物理通报, 2022, 1: 30-33



勤动手，多思考！

苏湘宁

邮箱: suxn@hainanu.edu.cn

