BÀI THỰC HÀNH LAB 1

Giảng viên: TS.Dương Việt Hằng

Trợ giảng: Trần Hà Sơn

Ngày 25 tháng 3 năm 2023

Mục tiêu:

- Nắm vững được Python để viết được các đoạn lệnh.
- Nâng cao kỹ năng lập trình cùng với tư duy toán học.
- Sử dụng được các gói lệnh NumPy, SciPy để thực hiện các tác vụ đơn giản trong xử lý các phép toán trong đại số và giải tích.

1 Cú pháp chính trong lập trình với Python

1.1 Phép gán

Cú pháp: **tendoituong = bieuthuc**.

Ví dụ 1. *Gán giá trị cho hai số nguyên x, y và thực hiện phép toán cộng.*

```
x=2
y=6
x+y
```

Output: *6*

Ví dụ 2. *Gán một danh sách cho biến x và đưa danh sách x ra màn hình.*

```
x=[1,2,3,4,5,6]
print(x)
```

Output: *[1,2,3,4,5,6]*

1.2 Các toán tử cơ bản trong Python

Toán tử	Mô tả	Ví dụ
+	Toán tử cộng các giá trị lại với nhau	x+y=6
-	Toán tử trừ các giá trị lại với nhau	5-6=-1
*	Toán tử nhân các giá trị lại với nhau	2*3=6
/	Toán tử chia các giá trị cho nhau	4/2=2
%	Toán tử chia lấy phần dư	5 % 2 =1
**	Lấy lũy thừa	2**3=8

1.3 Toán tử quan hệ trong Python

Toán tử	Mô tả
==	So sánh bằng
!=	So sánh không bằng
<>	Không bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng

1.4 Cấu trúc rẽ nhánh với if

1.4.1 Mệnh đề lựa chọn if

Cú pháp:

if(<điều kiện>):
[khối lệnh if]

Trong đó:

- <điều kiện> luôn cho kết quả trả về là một trong hai giá trị đúng (True) hoặc sai (False).
- [khối lệnh if]: Bao gồm một hoặc nhiều lệnh được chạy khi biểu thức logic nhận giá trị đúng (True), các câu lệnh trong khối lệnh có lề trái lệch sang bên phải một khoảng trống so với câu lệnh if.

Ví dụ 3. *Kiểm tra một số thực x có là số dương không.*

```
x=9
if (x>0):
     print(x, " la so duong.")
```

Output: *9 la so duong.*

1.4.2 Mệnh đề lựa chọn if...else

Cú pháp:

if(<điều kiện>):
    [khối lệnh 1]
else:
    [khối lệnh 2]

Ví dụ 4. *Kiểm tra một số thực x là số dương hay số âm.*

```
x=-6
if (x>0):
    print(x, " la so duong.")
else:
    print(x,"la so am hoac so khong.")
```

Output: *-6 la so am hoac so khong.*

1.5 Cấu trúc lặp for

Cú pháp:

for <tên biến> **in** <tập hợp>
[Khối lệnh for]

Trong đó

- <tên biến> là một tên do người dùng đặt tên.
- <tập hợp> gồm một tập các phần tử của List, String, Tuple,...

Ví dụ 5. *Tính tổng các số trong một danh sách cho trước.*

```
s=0
for i in [1,2,3,4,5]:
    s=s+i
print("Tong =",s)
```

Output: *Tong = 15*

1.6 Cấu trúc lặp while

Cú pháp:

while <điều kiện>:
[khối lệnh while]

Trong đó

- <điều kiện> là một biểu thức logic chỉ điều kiện cho kết quả là đúng hoặc sai.
- <khối lệnh while> bao gồm một hoặc nhiều câu lệnh được chạy khi biểu thức logic có giá trị đúng, các câu lệnh trong khối lệnh có lề trái dịch sang phải một khoảng so với từ khóa while.

Ví dụ 6. *Tính tổng các số trong một danh sách cho trước.*

```
s=0
i=0
x=[1,2,3,4,5]
n=len(x)
while (i<n):
    s=s+x[i]
    i=i+1
print("Tong =",s)
```

Output: *Tong = 15*

1.7 Cú pháp của hàm trong Python

Cú pháp:

def <tên hàm>(<dsts>):
<khối lệnh>
return[giá trị]

Trong đó

- <tên hàm> là tên của hàm do người dùng đặt theo quy tắc đặt tên, thường tên hàm nên đặt tên gợi nhớ theo nhiệm vụ của hàm.
- <dsts> là danh sách các tham số truyền vào làm giá trị đầu vào cho hàm để giải quyết công việc, <dsts> có thể có hoặc không, nếu có nhiều tham số thì cách nhau bởi dấu phẩy.
- <giá trị> là giá trị trả về của hàm sau khi hoàn thành công việc nếu có lệnh **return**.

Ví dụ 7. *Thực hiện nhân một số nhập từ bán phím với 10.*

```
1        def phepnhan(n):
2              return n*10
3        phepnhan(4)
```

Output: *40*

Hàm range() trả về một danh sách các số Cú pháp 1:

range(stop)

Cú pháp 2:

range(start, stop, step)

Trong đó

- start: số nguyên bắt đầu, chuỗi sẽ bắt đầu với tham số này, giá trị mặc định là 0.
- stop: số nguyên kết thúc, chuỗi sẽ kết thúc với tham số này.
- số nguyên xác định khoảng cách giữa các số bên trong chuỗi. Giá trị mặc định là 1.

2 Làm quen với thư viện Numpy

Các thư viện phổ biến dùng trong tính toán khoa học là NumPy, SciPy, Pandas, Scikit-learn, Mathplotlib,.... Trong phần này, chúng ta sẽ làm quen với thứ viện Numpy.

NumPy là một nền tảng tính toán của Python. Tài liệu của Numpy có thể tìm thấy dễ dàng trênmạng, như: https://docs.scipy.org/doc/numpy/numpy-ref-1.16.1.pdf (khoảng 1372 trang tính đến 01/2019). Chúng ta chỉ quan tâm đến một số lệnh của Numpy liên quan đến các bài toán đại số tuyến tính. Các lệnh còn lại sinh viên có thể chủ động tự tìm hiểu và nghiên cứu để sử dụng.
Để sử dụng Numpy, trước tiên chúng ta phải thực hiện khai báo thư viện:

⋙ **import numpy as np**

Với khai báo như vậy, để sử dụng thư viện NumPy, chúng ta phải sử dụng tiền tố **np**.

2.1 Một số lệnh cơ bản NumPy xử lý vector

2.1.1 Tạo vector

Một bộ gồm n số thực $v = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$ được gọi là một vector thuộc không gian n chiều. Các số $v_1, v_2, \ldots, v_n$ được gọi là các thành phần của vector. Sau đây là một số cách để tạo một vector trong Python:

- Dùng kiểu List đã có sẵn trên Python bằng cú pháp $[v_1, v_2, \ldots, v_n]$.
- Sử dụng hàm **np.array[]** của thư viện NumPy bằng cú pháp:

$$vec1 = np.array([v_1, v_2, \ldots, v_n])$$

- Sử dụng hàm **Matrix([])** của thư viện **sympy**, với vector sẽ được biểu diễn ở dạng cột.

Khi muốn truy cập một phần tử bất kỳ trong vector, ta gọi tên của vector đã đặt và kèm theo chỉ số thứ tự của phần tử cần xuất. Với lưu ý trong Python, số thứ tự của phần tử trong vector được đánh chỉ số từ **0** đến **n-1** hoặc từ **-n** đến **-1**.

Ví dụ 8. *Tạo và truy cập vào một số phần tử của vector.*

```
v=[12,23,25,34,50,49,27] # cach 1 de tao vector
w=np.array([12,23,25,34,50,49,27])# dung ham array trong thu vien NumPy
print("Kich thuoc cua vector w la:",len(w))
print(v)
print(w)
print(w[0])
# Truy cap entry, subsector theo thu tu cua mang duoc danh so tu 0 den n-1
print(w[-1])
```

Ví dụ 9. *Tạo và truy cập vào một số phần tử của vector.*

```
v=[12,23,25,34,50,49,27] # cach 1 de tao vector
w=np.array([12,23,25,34,50,49,27])# dung ham array trong thu vien NumPy
print("Kich thuoc cua vector w la:",len(w))
print(v)
print(w)
print(w[0])
# Truy cap entry  theo thu tu cua mang duoc danh so tu 0 den n-1
print(w[-1])
```

2.1.2 Tích vô hướng giữa hai vector

Để tính tích vô hướng giữa hai vector trong thư viện NumPy, ta dùng các hàm **np.iner(u,v)** hoặc phép tính **u@v**.

Ví dụ 10. *Tính tích vô hướng của hai vector.*

```
u=[1,2,-2]
v=[0,7,2]
np.iner(u,v)
```

Output: *10*

Ví dụ 11. *Tính tích vô hướng của hai vector.*

```
u1=np.array([1,2,-2])
v1=np.array([0,7,2])
u1 @ v1
```

Output: *10*

2.2 Giới thiệu ma trận

Để nhập ma trận, ta có thể sử dụng một trong các cách sau:

- Dùng danh sách lồng (nested list) các số

  [[dòng 1], [dòng 2],. . . , [dòng m]]

- Sử dụng mảng hai chiều trong thư viện NumPy:

  np.array([[dòng 1], [dòng 2],. . . , [dòng m]])

Ví dụ 12. *Tính tích vô hướng của hai vector.*

```
A=[[0,1,2],[1,4,6]]
print(A)
```

Output: *[[0,1,2],[1,4,6]]*

Ví dụ 13. *Tính tích vô hướng của hai vector.*

```
A=np.array([[0,2,1],[1,4,6]])
print(A, A.shape)
```

Output: *[[0,2,1],[1,4,6]] (2,3)*

3 Đồ án

Bài 1. Sinh viên viết hàm **Gauss_elimination(A)**, trong đó:

- Input: A là ma trận mở rộng của hệ phương trình.
- Output: Ma trận có dạng bậc thang có được từ ma trận A.

Lưu ý: Sinh viên không được dùng các hàm có sẵn trong thư viện để tìm ma trận bậc thang.

Bài 2. Sinh viên viết hàm **back_substitution(A)**, trong đó:

- Input: A là ma trận có dạng bậc thang từ ma trận mở rộng của hệ phương trình $Ax = b$.
- Output: Nghiệm của hệ phương trình (trường hợp nghiệm duy nhất/vô số nghiệm) hoặc thông báo hệ phương trình vô nghiệm.

Lưu ý: Sinh viên không được dùng các hàm có sẵn của thư viện để tìm ma trận bậc thang.

3.1 Quy định bài nộp

- Thực hiện toàn bộ bài làm trên một tập tin Python (.py).
- Nộp tập tin **MSSV.zip** được nén từ thư mục **MSSV** chứa các tập tin sau:
  1. Báo cáo toàn bộ bài làm: MSSV.pdf.
     Nội dung báo cáo gồm có: Thông tin cá nhân (họ và tên, mã số sinh viên); ý tưởng thực hiện, mô tả các hàm.
  2. Mã nguồn: MSSV.py

3.2 Quy định chấm bài

Những trường hợp sau sẽ bị **0** điểm toàn bộ đồ án:

- Nộp sai quy định.
- Không có báo cáo.
- Thực thi mã nguồn báo lỗi.

LƯU Ý: SAO CHÉP BÀI LÀM CỦA NHAU SẼ BỊ 0 ĐIỂM TOÀN BỘ PHẦN THỰC HÀNH.