

Problem-Based Learning

Python 3

Lecture: 03-for

Online Python

- <https://www.programiz.com/python-programming/online-compiler/>

Lecture Notes

- <https://web.phy.ntnu.edu.tw/~hongyi/?url=notes>



向量 vector

- $a = 0.95$
- 變數只能放一個數字

- 向量有 3 個變數

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

- 把 3 個變數放在一起，需要一個盒子

- `import numpy as np`
- `a = np.array([1,2,3])`

- `print(a)`

盒子(人類可以理解的形式)

盒子(將人類可以理解的形式轉成電腦的形式)

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

任意維度的向量

- Write a program to **repeatedly** prompt for a number between 0 and 10. until the user enters “done”. **Once “done” is entered.** print the vector.

$$\begin{array}{lcl} b = () & & \\ \downarrow & & \\ b = (b_0) & \leftarrow & \text{np.append} \\ \downarrow & & \\ b = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} & \leftarrow & \text{np.append} \\ \downarrow & & \\ b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \end{pmatrix} & \leftarrow & \text{np.append} \end{array}$$

附加元素np.append

- `import numpy as np`
- `b = np.array([])`
- `while True:`
- `C = input("Enter number (or 'done' to finish)= ")`
- `if C.lower() == "done":`
- `break`
- `try:`
- `a = float(C)`
- `if a >= 0 and a <= 10:`
- `b = np.append(b, a)`
- `except:`
- `print("Not a number!")`
- `print(b)`
- 向量**b**共有**N**個元素，編號從**0**開始到**N-1**結束。

$$b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{N-1} \end{pmatrix}$$

迴圈和編號

- Write a program to **repeatedly** prompt for a number between 0 and 10. until the user enters “done”. **Once “done” is entered, find the maximum number.** If the number is out of range, print an error message.
- 向量**b**共有**N**個元素，編號從**0**開始到**N-1**結束。

$$b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{N-1} \end{pmatrix}$$

For迴圈

- 附加底下的程式
 - `max = 0`
 - `for i in range(len(b)):`
 - `if max < b[i]:`
 - `max = b[i]`
 - `print(max)`
 -
- 向量的第*i*個元素

$$b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{N-1} \end{pmatrix} = \begin{pmatrix} b_{i=0} \\ b_{i=1} \\ b_{i=2} \\ \vdots \\ \vdots \\ b_{i=N-1} \end{pmatrix} \left. \vphantom{\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{N-1} \end{pmatrix}} \right\} \text{共有 } N \text{ 個元素}$$

- 說明：
- `for i in range(len(b)):`
 - ↑ 範圍
 - 得到 *N*
- *i* 從 0 開始到 *N*-1 結束, 對應向量個別元素的編號

向量大小 vector norm

- $\vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$
- $\vec{a}^2 = a_0 \times a_0 + a_1 \times a_1 + a_2 \times a_2$
- $|\vec{a}| = \sqrt{\vec{a}^2}$

- `import numpy as np`
- `a = np.array([0.3, 0.4, 0.1])`
- `sum = 0`
- `for i in range(3):`
- `sum = sum + a[i] * a[i]`
- `print(sum)`
- `print(np.sqrt(sum))`

向量內積 dot product

$$\vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

- $\vec{a} \cdot \vec{b} = a_0 \times b_0 + a_1 \times b_1 + a_2 \times b_2$

- `import numpy as np`
- `a = np.array([0.1, 0.2, 0.3])`
- `b = np.array([0.3, 0.5, 0.7])`
- `sum = 0.0`
- `for i in range(3):`
 - `c = a[i] * b[i]`
 - `sum = sum + c`
- `print(sum)`

向量乘以常數 vector multiplication

$$\vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

- $b_0 = a_0 \times 2$
- $b_1 = a_1 \times 2$
- $b_2 = a_2 \times 2$

- `import numpy as np`
- `a = np.array([0.1, 0.2, 0.3])`
- `b = np.zeros(3)`
- `for i in range(3):`
 - `b[i] = a[i] * 2.0`
- `print(b)`

向量相加 vector addition

$$\vec{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

- $c_0 = a_0 + b_0$
- $c_1 = a_1 + b_1$
- $c_2 = a_2 + b_2$

- `import numpy as np`
- `a = np.array([0.1, 0.2, 0.3])`
- `b = np.array([0.3, 0.5, 0.7])`
- `c = np.zeros(3)`
- `for i in range(3):`
 - `c[i] = a[i] + b[i]`
- `print(c)`

術語

電腦：

- 陣列(array)

人：

- 向量(vector)

Problem

- Write a program to use **for loop** to calculate the cross product of two vectors:

$$\vec{a} = \begin{pmatrix} 0.5 \\ 0.3 \\ 0.4 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix}, \quad \vec{c} = \vec{a} \times \vec{b} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$

$$\vec{c} = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \end{vmatrix} = (a_1 b_2 - a_2 b_1) \hat{x} + (a_2 b_0 - a_0 b_2) \hat{y} + (a_0 b_1 - a_1 b_0) \hat{z}$$

$$c[i] = a[j] * b[k] - a[k] * b[j]$$

Solution

- `import numpy as np`
- `a = np.array([0.5,0.3,0.4])`
- `b = np.array([2,1,4])`
- `vec = np.zeros(3)`
- `for i in range(3):`
 - `j = i + 1`
 - `if j > 2:`
 - `j = j - 3`
- `k = i + 2`
- `if k > 2:`
 - `k = k - 3`
- `vec[i] = a[j]*b[k] - a[k]*b[j]`
- `print(vec)`