

Tài liệu hướng dẫn thực hành Buổi 1

Môn : Lập trình Python

Nội dung chính : Giới thiệu ngôn ngữ lập trình Python

1. Giới thiệu

Python là một ngôn ngữ lập trình dạng thông dịch do Guido Van Rossum tạo ra năm 1990. Python được viết từ nhiều ngôn ngữ lập trình khác nhau và tạo ra những bản hiện thực khác nhau. Trong đó bản hiện thực chính là CPython được viết bằng C.

Python có tính tương thích lớn trên nhiều môi trường phát triển và cũng được ứng dụng rộng rãi trên nhiều lĩnh vực.


Điểm khác biệt đặc trưng giữa C và Python là Python sử dụng cơ chế cấp phát bộ nhớ tự động và hoàn toàn tạo kiểu động (khác với C phải khai báo biến và cấp phát bộ nhớ tương ứng với kiểu của biến). Điều này giúp tối thiểu hóa số lần gõ phím để viết mã lệnh và giúp cho cấu trúc có hình thức gọn gàng sáng sủa rất thuận tiện cho người mới học lập trình.

2. Cài đặt Python và trình soạn thảo

Download Python : Truy cập vào trang web : <https://www.python.org/> để download Python phiên bản 3.7

Download trình soạn thảo : Truy cập vào trang web : <https://atom.io/> để download trình soạn thảo Atom

Cách sử dụng Atom để viết một chương trình đơn giản bằng Python :

- Tạo một thư mục trong ổ D và đặt tên là Python
- Mở Atom, từ menu File chọn New File, ta thấy một cửa sổ mới được tạo ra.
- Bấm tổ hợp Ctrl + S để save file đó vào thư mục Python vừa tạo ở ổ đĩa D với tên gọi hello.py
- Trên cửa sổ soạn thảo của Atom viết vào dòng lệnh sau : `print ("Hello Python")` và bấm Ctrl + S để save lại
- Mở cửa sổ terminal (bấm tổ hợp  + R, gõ vào `cmd`).
- Trên cửa sổ terminal, gõ vào dòng lệnh `cd D:\Python` để di chuyển đến thư mục mà ta vừa tạo
- (Nếu báo lỗi thì thực hiện theo 2 bước, ta gõ `D:` để di chuyển qua ổ đĩa D, sau đó ta gõ `cd Python` để di chuyển vào thư mục Python)
- Thực thi file hello.py mà ta vừa tạo bằng cách gõ câu lệnh : `python hello.py`

3. Các cú pháp cơ bản

Python phân biệt các ký tự thường và hoa, đồng thời tăng cường sử dụng các từ khóa tiếng Anh, hạn chế sử dụng các ký hiệu và cấu trúc cú pháp so với các ngôn ngữ khác

Một số từ khóa thông dụng của Python :

- | | | |
|------------|---------|--------|
| • and | exec | not |
| • as | finally | or |
| • assert | for | pass |
| • break | from | print |
| • class | global | raise |
| • continue | if | return |
| • def | import | try |
| • del | in | while |
| • elif | is | with |
| • else | lambda | yield |

- except continue

Các toán tử cơ bản :

+ - * / // (chia làm tròn) % (phần dư) ** (lũy thừa)
 ~ (not) & (and) | (or) ^ (xor)
 << (left shift) >> (right shift)
 == (bằng) <= >= != (khác)

Sử dụng ký tự # để chú thích 1 dòng code

Sử dụng ký tự ''' để chú thích 1 đoạn code

Cấu trúc khối lệnh : sử dụng canh lề để bao các khối lệnh của hàm, lớp, hoặc luồng điều khiển.

Số khoảng trắng dùng để canh lề có thể tùy chọn nhưng các lệnh trong cùng một khối phải được canh lề như nhau.

Ví dụ :

```
a = 5
b = 3
if a > b :
    a = a * 2 + 3
    b = b - 6
    c = a/b
    print c
```

Dòng lệnh dài viết trên nhiều dòng sử dụng ký tự \

Ví dụ :

```
c = a + b + \
    10*a - b/4 - \
    5 + 3*a
print c
```

Lệnh nằm trong các cặp dấu ngoặc : [] {} () không cần sử dụng ký tự \ để tiếp tục dòng

Dấu ; để cách nhiều lệnh trên cùng 1 dòng

4. Lệnh và cấu trúc điều khiển

Lệnh if :

```
if biểu_thức_điều_kiện:
    # lệnh...

if biểu_thức_điều_kiện:
    # lệnh...
else:
    # lệnh...
```

Ví dụ :

```
a = 5
b = 3
if a > b :
    print "true"
    print a
else :
    print "false"
    print b
```

Lệnh while :

```
while biểu_thức_đúng:
    # lệnh...
```

Ví dụ :

```
a = 2
b = 6
while a < b :
    a+=1
    print a
```

Lệnh for :

```
for phần_tử in dãy:  
    # lệnh...
```

Ví dụ :

```
for i in range (1,10):  
    print i
```

Khai báo hàm :

```
def tên_hàm (tham_biến_1, tham_biến_2, tham_biến_n):  
    # lệnh...  
    return giá_trị_hàm
```

Ví dụ :

```
def binhphuong(number):  
    return number * number  
  
print binhphuong(3)
```

5. Bài tập

Sinh viên tạo file <<Họ tên>>.py và làm các bài tập sau trên cùng file đó. Ghi chú thích họ tên, mssv và bài làm từng câu trước khi làm.

Câu 1 : Viết chương trình in ra màn hình câu : “Hello! Welcome to Python!”

Câu 2 : Viết chương trình cho phép người dùng nhập tên vào, sau đó in ra màn hình câu : “Hello” + tên, “ Welcome to Python”

Câu 3 : Viết chương trình cho phép người dùng nhập vào 2 số a và b. Nếu b = 0 thì yêu cầu người dùng nhập lại cho đến khi b khác 0. In ra màn hình các kết quả của các phép toán sau : + - * / % **.

Câu 4 : Viết chương trình cho phép người dùng nhập vào bán kính hình tròn. In ra màn hình chu vi và diện tích tương ứng

Câu 5 : Viết chương trình tính giai thừa của số nguyên n được người dùng nhập vào từ bàn phím. Kiểm tra nếu n < 0 thì yêu cầu người dùng nhập lại,

Câu 6 : Viết hàm cho người dùng nhập vào 3 tham số a, b, c của phương trình $ax^2 + bx + c = 0$ và in ra màn hình nghiệm của phương trình đó

Câu 7 : Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không chia hết cho 5, nằm trong đoạn từ 5000 – 7000

Câu 8 : Viết chương trình chuyển số thập phân sang số nhị phân.

Câu 9 : Viết chương trình tìm ước chung lớn nhất và bội chung nhỏ nhất của 2 số nguyên dương a và b được nhập vào từ bàn phím

Câu 10 : Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n, với n được nhập vào từ bàn phím

Câu 11 : Viết chương trình liệt kê tất cả các số nguyên tố có 4 chữ số

Câu 12 : Viết chương trình tính tổng của các chữ số của một số nguyên n. Nếu n nhỏ hơn 0 thì yêu cầu người dùng nhập lại. (ví dụ : n = 123 thì tổng là $1 + 2 + 3 = 6$)

Câu 13 : Viết chương trình tìm tất cả các số trong khoảng từ 1000 – 2000 sao cho tất cả các chữ số trong đó đều là số lẻ (ví dụ : 1157, 1597, 1375,...)

Câu 14 : Viết chương trình tính $\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \dots + \frac{n}{n+1}$ với số n được nhập từ bàn phím. Nếu n < 0 thì yêu cầu người dùng nhập lại.

Câu 15 : Viết chương trình tính dãy fibonacci f(n) với giá trị n được nhập từ bàn phím. Cách tính như sau :

$f(0) = 0, f(1) = 1, f(2) = f(1) + f(0), f(3) = f(2) + f(1), \dots, f(n) = f(n-1) + f(n-2)$