

Lập trình căn bản với Python

1

Nội dung

1. Giới thiệu về ngôn ngữ Python
2. Câu lệnh, chương trình
3. Biến, kiểu dữ liệu và câu lệnh
4. Nhập/xuất dữ liệu

2

2

Ngôn ngữ Python

- Python giới thiệu vào tháng 12/1989 => kế thừa từ ngôn ngữ ABC
- Guido van Rossum (Hà Lan), (1956), đang làm cho Google
- Python 2 được giới thiệu năm 2000
 - Hỗ trợ unicode
- Python 3 được phát hành năm 2008
 - Hiện đã có phiên bản 3.9.5

3

3

Ngôn ngữ Python

- Là ngôn ngữ mã nguồn mở
- Vừa hướng thủ tục, vừa hướng đối tượng
- Hỗ trợ module và hỗ trợ gói (package)
- Xử lý lỗi bằng ngoại lệ (exception)
- Kiểu dữ liệu động ở mức cao
- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface)
- Có khả năng tương tác với các module viết bằng ngôn ngữ lập trình khác

4

4

Ưu điểm Python

- Ngữ pháp đơn giản, dễ đọc
- Viết mã ngắn gọn hơn những chương trình tương đương được viết trong C, C++, C#, Java,...
- Các bộ thư viện chuẩn và các module ngoài, đáp ứng gần như mọi nhu cầu lập trình
- Chạy được trên nhiều nền tảng (Windows, Linux, Unix, OS/2, Mac, Amiga, máy ảo .NET, máy ảo Java,...)
- Cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, mã nguồn chia sẻ nhiều

5

5

Nhược điểm Python

- Chương trình chạy chậm
- Giao tiếp với các thư viện viết bằng các ngôn ngữ khác tương đối khó khăn
- Kém trong hỗ trợ tính toán trên di động
- Gỡ lỗi đòi hỏi kinh nghiệm
- Kém hỗ trợ các CSDL

6

6

Cài đặt



<https://www.python.org/downloads/>

7

7

Khởi chạy

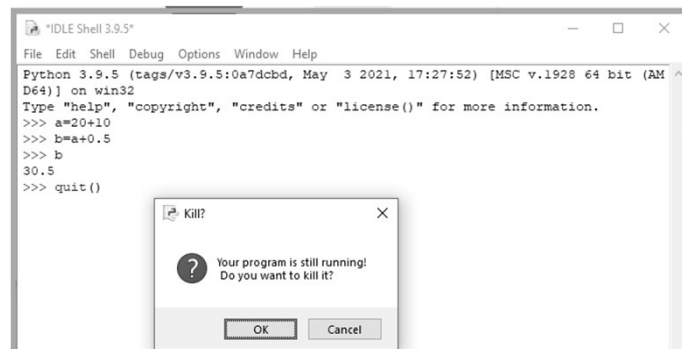
- Python có 2 chế độ thực thi
 - Chế độ thực thi: chỉ ra chương trình cần thực hiện
 - Chế độ dòng lệnh: chạy từng lệnh một
- Chế độ thực thi: **"python abc.py"** chạy file **abc.py**

8

8

Chế độ dòng lệnh

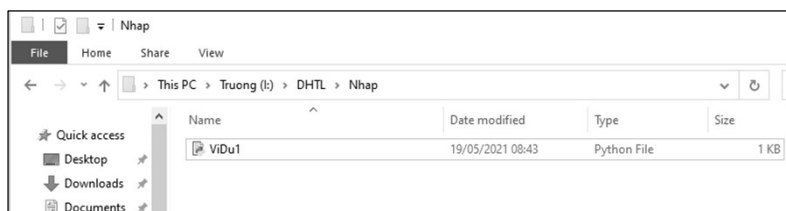
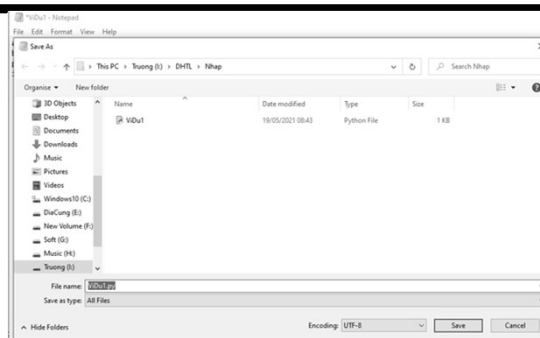
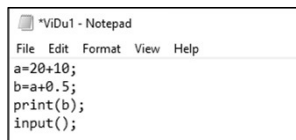
- Lúc này trình thông dịch python sẽ chờ người dùng gõ từng dòng lệnh
- Gõ dòng lệnh nào xong, python chạy liền dòng đó
- Chấm dứt chế độ này bằng cách gõ lệnh: **"quit()"**



9

9

Ví dụ

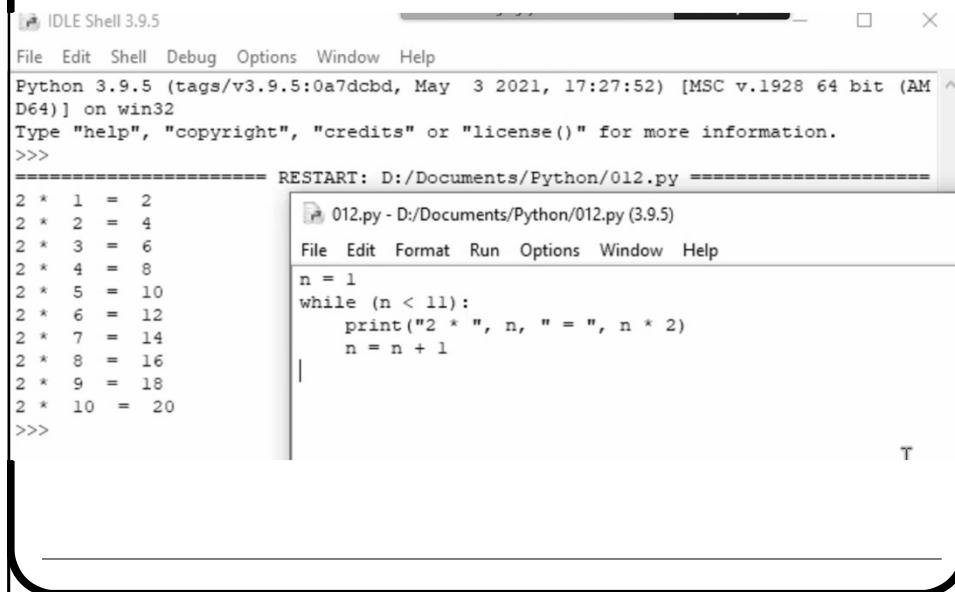


C:\WINDOWS\py.exe

30.5

10

Ví dụ



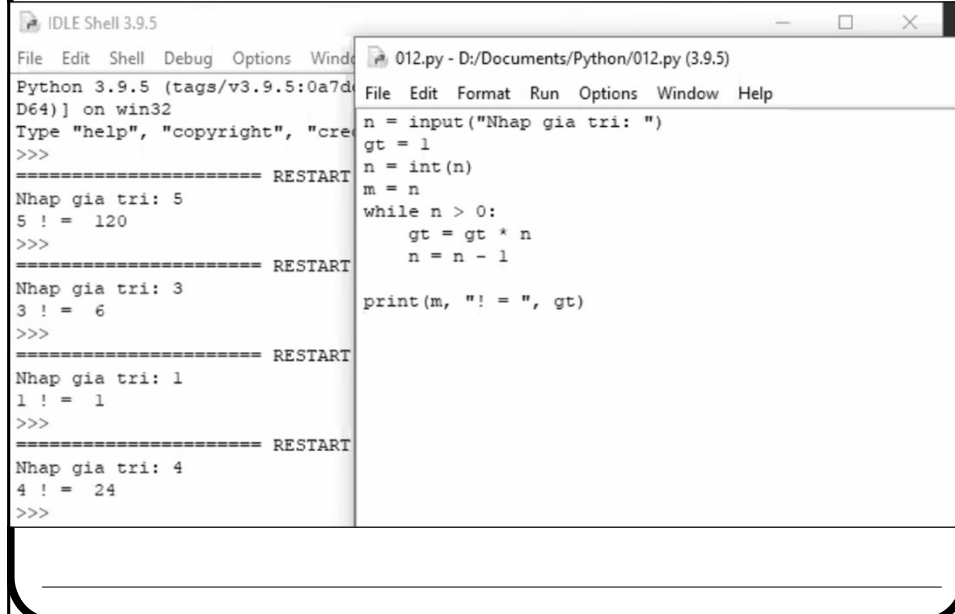
The screenshot shows the Python IDLE Shell 3.9.5 interface. The main window displays a multiplication table for numbers 1 through 10. A secondary window titled '012.py - D:/Documents/Python/012.py (3.9.5)' is open, showing a Python script that generates the same multiplication table using a while loop.

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/Documents/Python/012.py =====
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
>>>
```

```
012.py - D:/Documents/Python/012.py (3.9.5)
File Edit Format Run Options Window Help
n = 1
while (n < 11):
    print("2 * ", n, " = ", n * 2)
    n = n + 1
```

11

Ví dụ: Tính $n!$, với n nhập từ bàn phím



The screenshot shows the Python IDLE Shell 3.9.5 interface. The main window displays several factorial calculations performed by the user. A secondary window titled '012.py - D:/Documents/Python/012.py (3.9.5)' is open, showing a Python script that calculates the factorial of a user-input number using a while loop.

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART =====
Nhap gia tri: 5
5 ! = 120
>>>
===== RESTART =====
Nhap gia tri: 3
3 ! = 6
>>>
===== RESTART =====
Nhap gia tri: 1
1 ! = 1
>>>
===== RESTART =====
Nhap gia tri: 4
4 ! = 24
>>>
```

```
012.py - D:/Documents/Python/012.py (3.9.5)
File Edit Format Run Options Window Help
n = input("Nhap gia tri: ")
gt = 1
n = int(n)
m = n
while n > 0:
    gt = gt * n
    n = n - 1
print(m, "! = ", gt)
```

12

Chế độ thực thi (soạn thảo)

- Viết chương trình python(.py) => dùng phần mềm soạn thảo văn bản thô (txt) bất kỳ để soạn, lưu file ở dạng .py => dịch bằng python
- Ví dụ:
 - IDLE
 - Sublime Text
 - Notepad++
 - PyCharm,....

13

Thực hiện câu lệnh

Chế độ thực thi (biên dịch mã python)

- Mã python có thể được biên dịch, kết quả dịch => chương trình dạng bytecode cho máy ảo python
=> Tương tự như ngôn ngữ java
- Mã lệnh dịch được lưu vào file với đuôi **.pyc**

14

14

Biến

- Biến = vùng bộ nhớ được đặt tên (để dễ thao tác)
- Cách đặt tên biến:
 - Có tên, phân biệt chữ hoa/thường
 - Không cần khai báo trước
 - Không cần chỉ ra kiểu dữ liệu
 - Có thể thay đổi sang kiểu dữ liệu khác
 - Nên gán giá trị ngay khi bắt đầu xuất hiện
- Ví dụ:

```
a = 4          # biến a là kiểu số nguyên  
a = a + 0.6    # biến a chuyển sang kiểu số thực
```

15

15

Biến

- Tên biến có thể chứa chữ cái hoặc chữ số hoặc gạch dưới (), kí tự bắt đầu không được dùng chữ số
 - Không được trùng với từ khóa.
 - Từ python 3 được dùng chữ cái unicode

16

16

Kiểu int (số nguyên)

- Kiểu **int** biểu diễn số nguyên (Integer)
 - Các toán tử: +, −, *, //, **, toán tử 1 ngôi −
- Các toán tử trên các giá trị **int** phải mang kiểu **int**
 - Ví dụ: 2 // 3, kết quả trả về phần nguyên 0;
 - Toán hạng thường sử dụng cùng: % (lấy phần dư)
 - Toán hạng / KHÔNG phải là phép chia trong Python3
- Trong Python 3.X, kiểu dữ liệu số nguyên là vô hạn => đa số các ngôn ngữ lập trình khác KHÔNG THỂ.

17

17

Kiểu float (số thực)

- Kiểu float biểu diễn số thực: tập hợp các số nguyên và số thập phân 1, 1.4, -123, 69.96,...
- **Giá trị**: phân biệt với số nguyên bởi các số thập phân
 - Trong Python, một số có '.' là một số thực (Ví dụ: 2.0)
- Các toán tử: +, −, *, /, **, toán tử 1 ngôi −
 - Phép chia / khác với số nguyên //
 - Ví dụ: 1.0 // 2.0 được tính bằng 0

18

18

Kiểu phân số

- Tạo phân số trong Python: `Fraction(<Tử_số>, <Mẫu_số>)`
- Ví dụ: Nhập phân số $\frac{1}{4}$, $\frac{3}{9}$, $\frac{3}{4}$,

19

19

Kiểu chuỗi

- Dữ liệu kiểu chuỗi rất quan trọng trong lập trình python và trong các vấn đề của khoa học dữ liệu
- Khai báo dữ liệu kiểu chuỗi có thể nằm bên trong cặp nháy đơn ('), hoặc nháy kép (") hoặc 3 dấu nháy kép liên tiếp (""")

20

20

Kiểu Boolean

- Thường gọi tắt là kiểu Bool/ kiểu Đúng – Sai/ kiểu Luận lý
- Là kiểu giá trị này chỉ có hai giá trị là đúng (True) và sai (False)

21

21

Chú thích

- Python sử dụng kí tự # để chú thích các đoạn code
- Tất cả các nội dung sau kí tự # sẽ không được dịch

22

22

Nhập-Xuất dữ liệu

- ☐ Sử dụng hàm input để nhập dữ liệu từ bàn phím
- ☐ Sử dụng hàm print để in dữ liệu ra màn hình
- ☐ Ví dụ:

```
a=input(' Nhập a =');  
Print(a);
```

23