Lập trình Python

Bài 1: Giới thiệu về ngôn ngữ python

Nội dung

1. Thông tin chung về môn học
2. Giới thiệu ngôn ngữ python
3. Cách thực hiện câu lệnh, chương trình
4. Biến, khai báo chuỗi, khối lệnh
5. Nhập dữ liệu và xuất dữ liệu
6. Kiểu dữ liệu và phép toán liên quan
7. Vài ví dụ minh họa
8. Bài tập

Phần 1

Thông tin chung về môn học

Giới thiệu môn học

- Tên môn: (Ngôn ngữ) Lập trình Python (Python programming language)
- Số tín chỉ: 3 (30 tiết lý thuyết + 15 tiết bài tập)
- Nội dung chính:
  - Cơ bản về ngôn ngữ lập trình python (kiểu dữ liệu, phép toán, biểu thức, rẽ nhánh, lặp, hàm,...)
  - Các kiểu dữ liệu đặc trưng của python (string, tuple, list,...)
  - Làm việc với tập tin trong python
  - Các tính năng nâng cao của ngôn ngữ python (ngoại lệ, hướng đối tượng,...)
- Giảng viên: Trương Xuân Nam, khoa CNTT
- Email: namtx@tlu.edu.vn / truongxuannam@gmail.com

Tài liệu môn học và phần mềm học tập

- Tài liệu chính: bài giảng của giáo viên
  - Sách giáo trình đang được biên soạn, hiện chưa có
- Phần mềm học tập: python 3.x
  - Có thể sử dụng bất kỳ phần mềm nào, miễn là nó hỗ trợ ngôn ngữ python 3.5 trở lên
  - Trên lớp, thầy giáo sẽ minh họa bằng phần mềm tiêu chuẩn, lấy từ site https://www.python.org
- Bài giảng, bài tập, mã nguồn, điểm số,… sẽ được đưa lên site https://txnam.net mục BÀI GIẢNG
  - Bài giảng (và bài tập) sẽ được đưa lên trước giờ học
  - Giờ thực hành: sinh viên thực hành trên phần mềm FineTest
  - Điểm quá trình cũng sẽ được công bố trên website

Phần mềm học tập

Python + Python IDLE

FineTest

Kiến thức yêu cầu

- Đã biết và sử dụng được một ngôn ngữ lập trình nào đó (C/C++, C#, Java, Javascript, Pascal,…)
  - Vì chúng ta sẽ học khá nhanh, nhiều kiến thức
  - Sử dụng được tức là có thể viết chương trình với ngôn ngữ đó
- Có kiến thức về các khái niệm cơ bản trong lập trình
  - Môn học này giúp sinh viên hiểu hơn về những khái niệm đó
- Biết sử dụng email
  - Nộp bài tập vào email của thầy giáo: cần ghi rõ tên sinh viên, bài nộp là bài nào, của buổi bài tập số mấy
  - Có thể email cho thầy giáo để hỏi thêm các vấn đề về môn học

- Chú ý: copy bài của bạn khác để nộp sẽ bị cấm thi

Đánh giá kết quả

- Điểm môn học:
  - Điểm quá trình: **50%**
  - Điểm thi cuối kỳ: **50%**
- Điểm quá trình:
  - Điểm chuyên cần 10%
  - Điểm kiểm tra 40%
  - Điểm thưởng do chữa bài, giải bài,...
- Thi cuối kỳ:
  - Máy chấm tự động trên FineTest
  - Học gì thi nấy, không hỏi ngoài môn học
  - Không có giới hạn nội dung thi
  - Không sử dụng tài liệu tham khảo

Mục tiêu của môn học này

- Biết cài đặt thuật giải bằng ngôn ngữ lập trình python
- Rèn luyện thói quen lập trình một cách "trong sáng" ;)
- Học lập trình python phần cơ bản, để có thể sử dụng trong các môn học sau này
- Làm quen với lập trình hướng đối tượng
- Làm quen với cách các thuật toán có thể ứng dụng vào bài toán thực tế như thế nào
  - Mô tả bài toán theo cách của dân máy tính
  - Lựa chọn phương pháp xử lý phù hợp

Lời khuyên

ĐI HỌC ĐẦY ĐỦ

LÀM HẾT TẤT CẢ CÁC BÀI TẬP

Chỉ thế thôi!!!

Phần 2

Giới thiệu ngôn ngữ python

Giới thiệu ngôn ngữ python

- Python lần đầu được giới thiệu vào tháng 12/1989
- Tác giả là Guido van Rossum (Hà Lan)
  - Sinh năm 1956
  - Hiện đang làm cho Microsoft
- Python kế thừa từ ngôn ngữ ABC
- Python 2 được giới thiệu năm 2000
  - Hỗ trợ unicode
  - Mã python 2 rất phổ biến
- Python 3 được phát hành năm 2008
  - Hiện đã có phiên bản 3.11.4
- Python 4? Đã bị hủy bỏ kế hoạch

Giới thiệu ngôn ngữ python

- Là ngôn ngữ có mã nguồn mở
- Là ngôn ngữ kịch bản (scripting programming language)
  - Thích hợp với DevOps (người viết code cũng là người vận hành)
  - Khai báo biến tự nhiên, phong phú và động
  - Nhiều phép tính cấp cao được cung cấp sẵn
  - Thường được thông dịch thay vì biên dịch
    - Biên dịch: dịch toàn bộ thành mã máy rồi thực thi
    - Thông dịch: dịch từng lệnh, xong lệnh nào chạy lệnh đó
- Những người cuồng python (pythonista) cho rằng ngôn ngữ này trong sáng và tiện dụng đến mức ta có thể dùng nó cho mọi khâu lập trình (chứ không phải chỉ viết script)

Giới thiệu ngôn ngữ python

- 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 khả năng tương tác với các module viết bằng ngôn ngữ lập trình khác
- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface)

Ưu điểm của ngôn ngữ python

- Có ngữ pháp đơn giản, dễ đọc
- Viết mã ngắn gọn hơn những chương trình tương đương được viết trong C, C++, C#, Java,…
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng gần như mọi nhu cầu lập trình
- Có khả năng chạy trên nhiều nền tảng (Windows, Linux, Unix, OS/2, Mac, Amiga, máy ảo .NET, máy ảo Java, Nokia Series 60,…)
- Có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, mã nguồn chia sẻ nhiều

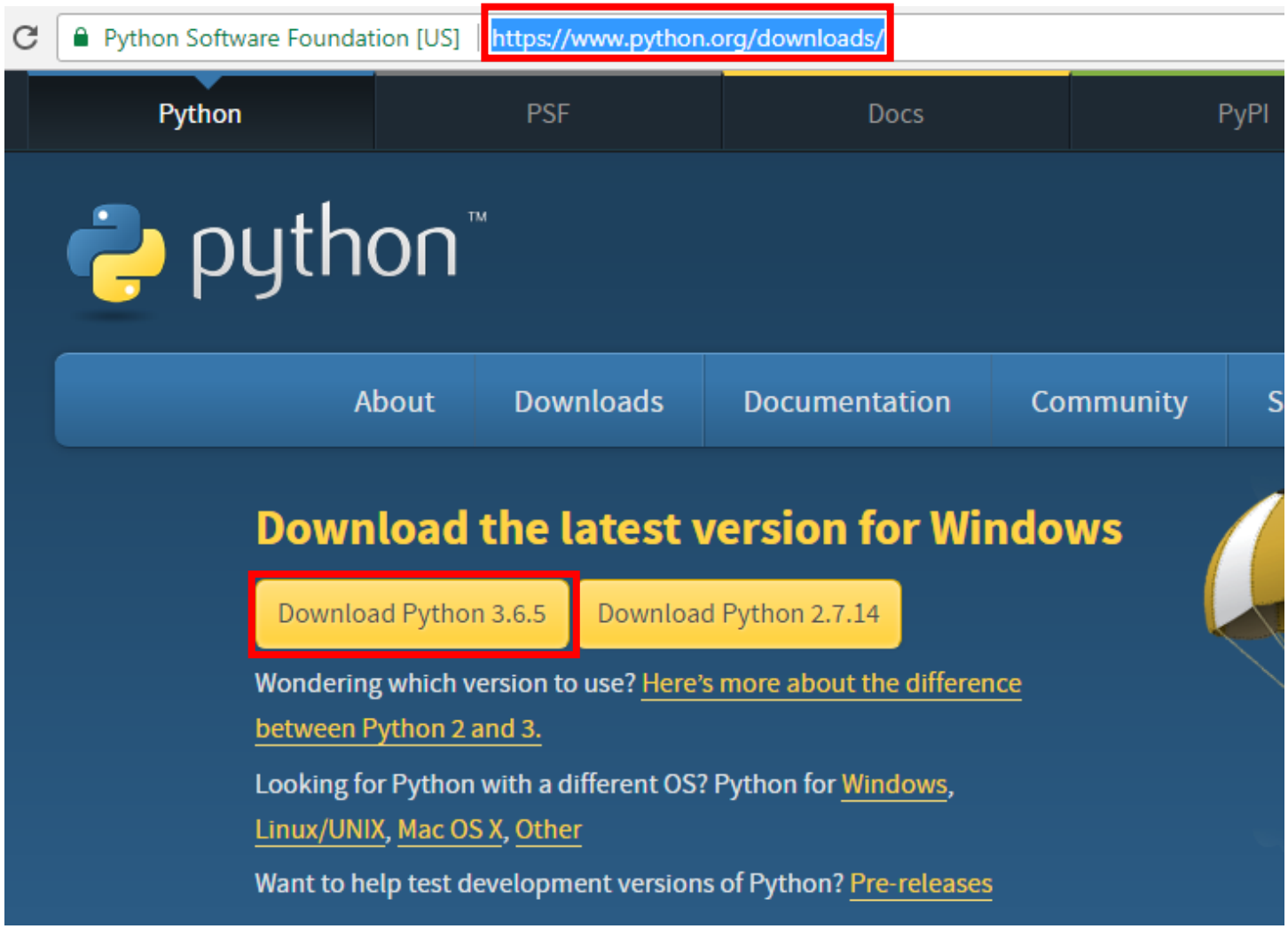
Nhưng python cũng có nhược điểm

- Chương trình chạy chậm
- Giao tiếp với các thư viện viết bằng các ngôn ngữ khác tương đối khó khăn
- Yếu trong hỗ trợ tính toán trên di động
- Cách viết khối lệnh dễ gây nhầm lẫn cho người mới bắt đầu lập trình
- Gỡ lỗi đòi hỏi kinh nghiệm
- Kém hỗ trợ các cơ sở dữ liệu
- Xử lý song song kém

Phần 3

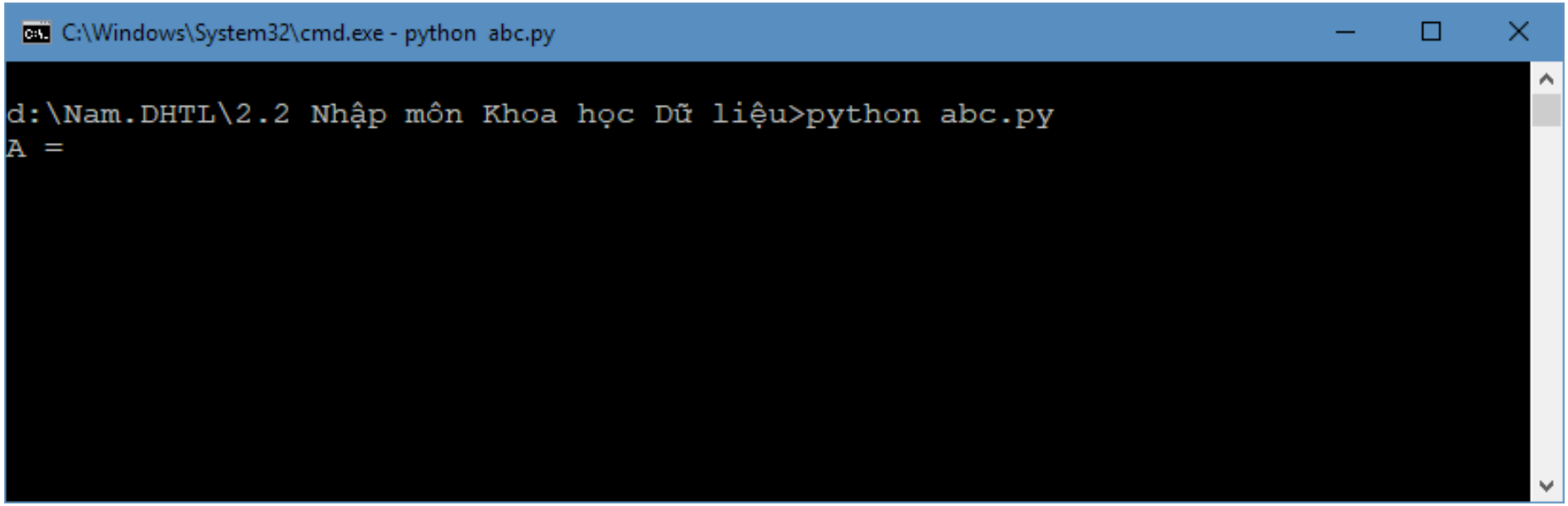
Cách thực hiện câu lệnh, chương trình

Cài đặt


Python Software Foundation [US]
https://www.python.org/downloads/
Python
PSF
Docs
PyPI
python™
About
Downloads
Documentation
Community
Download the latest version for Windows
Download Python 3.6.5
Download Python 2.7.14
Wondering which version to use? Here's more about the difference between Python 2 and 3.
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other
Want to help test development versions of Python? Pre-releases

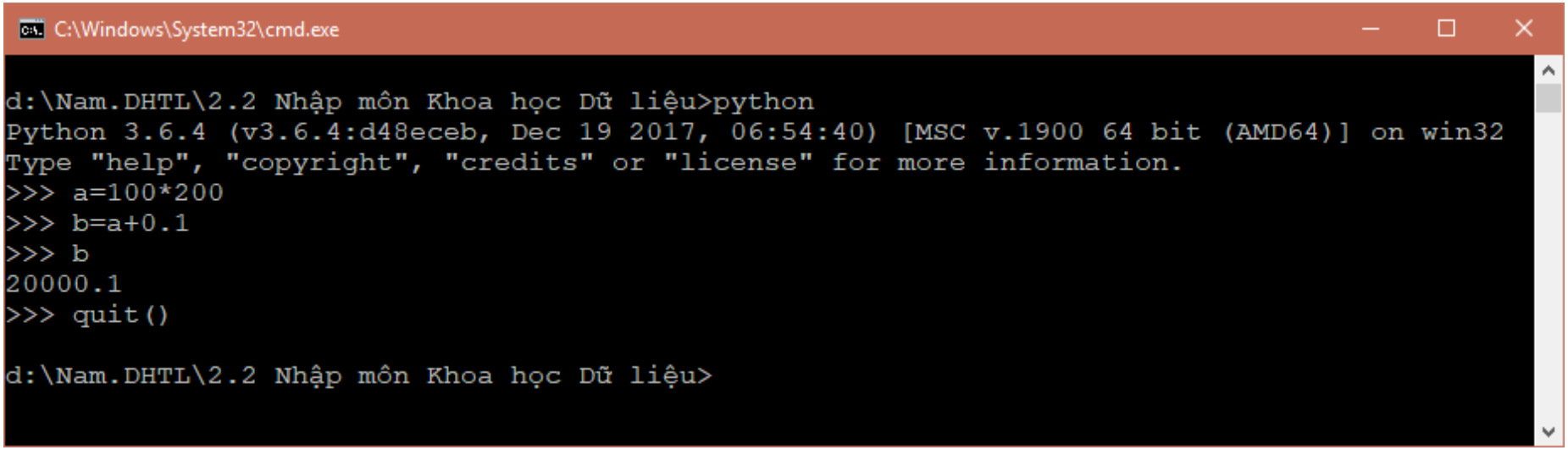
Khởi chạy

- Python có 2 chế độ thực thi
  - Chế độ chương trình: chỉ ra chương trình cần thực hiện
    - Trình dịch python sẽ nạp, dịch và chạy chương trình đó
  - Chế độ dòng lệnh: gõ và chạy từng lệnh một
- Chế độ thực thi: "python abc.py" chạy file abc.py



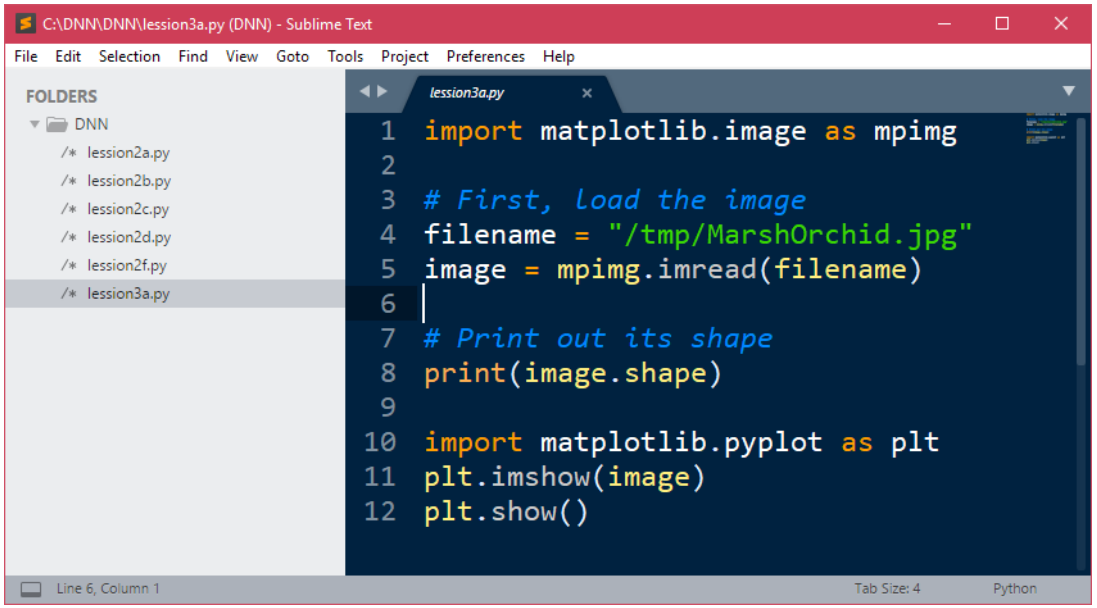
Khởi chạy

- Chế độ dòng lệnh: “python”
  - 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()”



Soạn thảo mã python

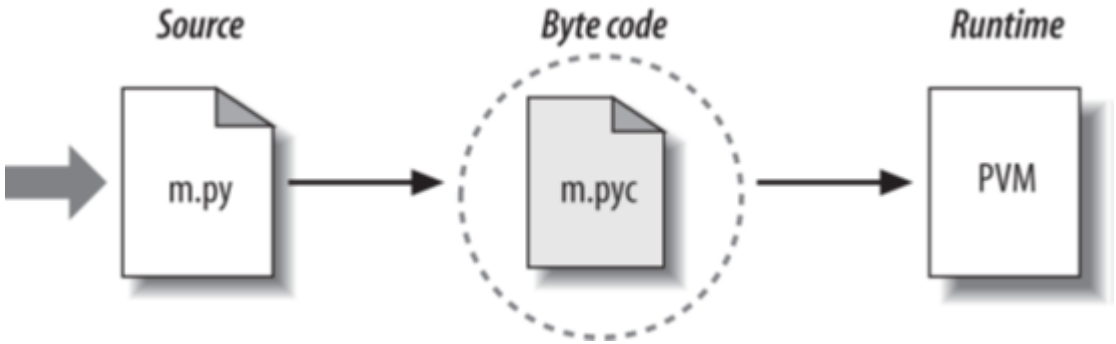
- Làm thế nào để viết chương trình python (.py)?
  - Dùng phần mềm soạn thảo văn bản thô (txt) bất kỳ để soạn và lưu file ở dạng .py rồi dịch bằng python
- Có những phần mềm thích hợp cho việc này hơn
  - IDLE
  - Sublime Text
  - Notepad++
  - PyCharm
  - Spyder
  - Rodeo
  - …



- Jupyter: chạy trên trình duyệt, thích hợp với thử nghiệm và giảng dạy

Biên dịch mã python

- Mã python có thể được biên dịch, kết quả biên dịch là chương trình dạng bytecode cho máy ảo python
  - Tương tự như trường hợp của ngôn ngữ java
- Mã lệnh dịch được lưu vào file với đuôi .pyc
- Việc biên dịch có nhiều lợi điểm, chẳng hạn như khi sử dụng câu lệnh import một thư viện nào đó, thì có thể sử dụng luôn mã pyc có sẵn thay vì phải dịch lại từ đầu
  - Tăng tốc độ thực hiện chương trình



Phần 4

Biến, khai báo chuỗi, khối lệnh

Biến

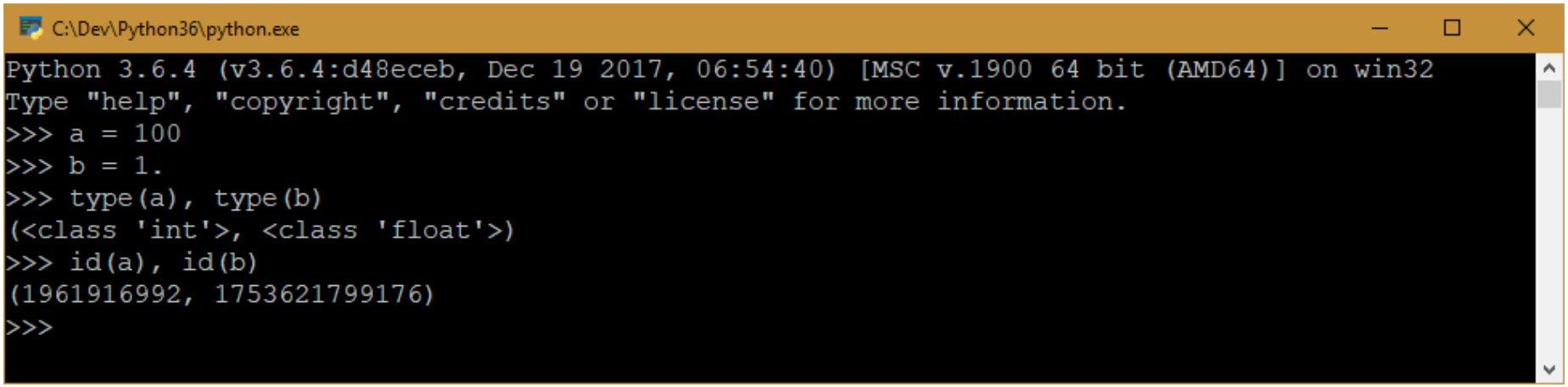
- Biến = vùng bộ nhớ được đặt tên (để dễ thao tác)
- Ví dụ:

```
n = 12          # biến n là kiểu nguyên
n = n + 0.1     # biến n chuyển sang kiểu thực
```

- Biến trong python:
  - 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
- Python viết ghi chú trong chương trình bằng cách đặt sau dấu thăng (#), đây là loại ghi chú một dòng

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 (tất nhiên)
  - Từ python 3 được dùng chữ cái unicode
- Tất cả mọi biến trong python đều là các đối tượng, vì thế nó có kiểu và vị trí trong bộ nhớ (id)



Khai báo chuỗi

- Dữ liệu kiểu chuỗi rất quan trọng trong lập trình python, tương tự như các ngôn ngữ lập trình khác
- Ví dụ:

```
# chuỗi thông thường
name = 'matt'
# chuỗi trong nó có chứa dấu nháy đơn
with_quote = "I ain't gonna"
# chuỗi có nội dung nằm trên nhiều dòng
longer = """This string has
multiple lines in it"""
```

- Nguyên tắc khai báo chuỗi: mở đầu sao - kết thúc vậy
  - Nội dung trên 1 dòng: dùng cặp nháy đơn (') hoặc nháy kép (")
  - Nội dung nằm trên nhiều dòng: 3 dấu nháy liên tiếp (""" / ''')

Chuỗi thoát (escape sequence)

- Escape sequence là một phương pháp để viết các kí tự đặc biệt (không thể viết theo lối thông thường)
  - Tương tự như các ngôn ngữ lập trình khác

Cách viết	Ý nghĩa	Thuật ngữ
\a	Kí tự cảnh báo (phát ra một tiếng bíp nếu in ra)	Alert
\b	Kí tự xóa trước (dịch con trỏ về phía trước 1 ô)	Backspace
\n	Kí tự dòng mới (dịch con trỏ xuống dòng dưới)	Linefeed
\r	Kí tự trở về (dịch con trỏ về đầu dòng)	Carriage return
\t	Kí tự tab (dịch con trỏ đi 1 dấu tab)	Tab
\\	Kí tự gạch chéo (\)	Blackslash
\'	Kí tự dấu nháy đơn (')	Single quote
\"	Kí tự dấu nháy kép (")	Double quote
\uxxxx	Kí tự unicode bất kì có mã xxxx (dạng hex value)	

Chuỗi thô (raw string)

- Vấn đề: dễ nhầm lẫn khi các chuỗi có dấu gạch chéo (\)
  - Chẳng hạn như khi viết tên file `"c:\teamview"`
- Python cho phép bỏ qua các chuỗi thoát bằng cách đánh dấu chữ r vào trước chuỗi, định dạng này gọi là chuỗi thô
  - Cú pháp: `r'nội dung chuỗi'`

```
>>> a = 'c:\teamview'
>>> print(a)
c:      eamview
>>> b = 'c:\\teamview'
>>> print(b)
c:\teamview
>>> c = r'c:\teamview'
>>> print(c)
c:\teamview
```

Khối lệnh

- Python sử dụng khoảng trắng để phân biệt khối lệnh

```
age = int(input("Bạn bao nhiêu tuổi? "))
print("Ồ bạn đã", age, "tuổi rồi!")
if age>=18:
    print("Đủ tuổi đi bầu")
    if age>100:
        print("Có vẻ sai sai!")
else:
    print("Nhỏ quá")
```

- Chú ý:
  - Không quy định số lượng khoảng trắng phải sử dụng
  - Các lệnh cùng một khối phải sử dụng cùng số khoảng trắng
  - Sử dụng tab hoặc space đều được, nhưng phải thống nhất

Phần 5

Nhập dữ liệu và xuất dữ liệu

Xuất dữ liệu

- Sử dụng hàm print để in dữ liệu ra màn hình

```
>>> print(42)
42
>>> print("a = ", a)
a =  3.564
>>> print("a = \n", a)
a =
 3.564
>>> print("a", "b")
a b
>>> print("a", "b", sep="")
ab
>>> print(192, 168, 178, 42, sep=".")
192.168.178.42
>>> print("a", "b", sep=":-)")
a:-)b
```

Nhập dữ liệu

- Sử dụng hàm input để nhập dữ liệu từ bàn phím

```
name = input("Tên bạn là gì? ")
print("Xin chào bạn " + name + "!")
age = input("Bạn bao nhiêu tuổi? ")
print("Ồ, bạn đã " + age + " tuổi rồi!")
```

- Có thể kết hợp chuyển kiểu nếu muốn tường minh

```
age = int(input("Bạn bao nhiêu tuổi? "))
print("Ồ bạn đã %d tuổi rồi!" % age)
```

Phần 6

Kiểu dữ liệu và phép toán liên quan

Kiểu số

- Python cho phép viết số nguyên theo một số hệ cơ số thông dụng trong lập trình

```
A = 1234        # hệ cơ số 10
B = 0xAF1       # hệ cơ số 16
C = 0o772       # hệ cơ số 8
D = 0b1001      # hệ cơ số 2
```

- Sử dụng các hàm phù hợp để chuyển đổi từ số nguyên thành string ở các hệ cơ số 10, 16, 8 hoặc 2

```
K = str(1234)   # chuyển thành str ở hệ cơ số 10
L = hex(1234)   # chuyển thành str ở hệ cơ số 16
M = oct(1234)   # chuyển thành str ở hệ cơ số 8
N = bin(1234)   # chuyển thành str ở hệ cơ số 2
```

Kiểu số

- Từ python 3, số nguyên không có giới hạn số chữ số
- Số thực (float) trong python có thể viết theo dạng thông thường hoặc dạng khoa học

```
X = 12.34
Y = 314.15279e-2  # dạng số nguyên và phần mũ 10
```

- Python hỗ trợ kiểu phức, với chữ j đại diện cho phần ảo

```
A = 3+4j
B = 2-2j
print(A+B)          # sẽ in ra (5+2j)
```

Phép toán

- Python hỗ trợ nhiều phép toán số, logic, so sánh, phép toán bit và phép kiểm tra tập
  - Các phép toán số thông thường: `+, -, *, %, **`
  - Python có 2 phép chia:
    - Chia đúng (/): `10/3 # 3.3333333333333335`
    - Chia nguyên (//): `10//3 # 3 (nhanh hơn phép /)`
  - Các phép logic: `and, or, not`
    - Python không có phép xor logic, trường hợp muốn tính phép xor thì thay bằng phép so sánh khác (bool(a) != bool(b))
  - Các phép so sánh: `<, <=, >, >=, !=, ==`
  - Các phép toán bit: `&, |, ^, ~, <<, >>`
  - Phép kiểm tra tập (in, not in): `1 in [1, 2, 3]`

Phần 7

Vài ví dụ minh họa

Giải phương trình bậc 2

```
a = float(input("A = "))
b = float(input("B = "))
c = float(input("C = "))
delta = b*b-4*a*c

if delta==0:
    print("Nghiem kep: x = ", str(-b/2/a))

if delta<0:
    print("Phuong trinh vo nghiem")

if delta>0:
    print("X1 = " + str((-b+delta**0.5)/2/a))
    print("X2 = " + str((-b-delta**0.5)/2/a))
```

Nhập a,b,c kiểu số thực và tính delta

Biện luận các trường hợp của delta

Các khối lệnh con được viết thụt vào so với khối cha

Tính căn bậc 2 bằng phép lũy thừa 0.5

Tính n!

```
def giaithua(n):
    gt = 1
    for i in range(2, n+1):
        gt = gt * i
    return gt

a = int(input("Nhập giá trị n: "))
print("N! =", giaithua(a))
```

Định nghĩa hàm với tham số n

Vòng lặp cho i chạy từ 2 đến n

Trả về kết quả

Nhập số n nguyên

Gọi hàm tính và in ra kết quả

Tính UCLN (thuật toán euclid)

```
a = int(input("A = "))
b = int(input("B = "))

while (b > 0):
    if (a > b):
        a, b = b, a % b
    else:
        a, b = a, b % a

print("Ước số chung lớn nhất là:", a)
```

Nhập 2 số nguyên a và b

Vòng lặp chừng nào b > 0

Xử lý khi a > b

Xử lý khi a <= b

In kết quả

Phần 8

Bài tập

Bài tập

1. Gõ 3 ví dụ phía trước vào 3 file, đặt tên đuôi .py, sau đó sử dụng trình thông dịch python để chạy thử xem kết quả ra sao
2. Bạn có 10 triệu đồng trong tài khoản ngân hàng, với lãi xuất 5,1% hàng năm. Tính xem:
   - Sau 10 năm bạn có bao nhiêu tiền?
   - Sau bao nhiêu năm bạn sẽ có ít nhất 50 triệu đồng?
3. Nhập số nguyên n, hãy in ra n ở dạng hệ cơ số 16, hệ cơ số 8 và hệ cơ số 2
4. Nhập 2 số nguyên a và b, hãy tính và in ra $\sqrt[b]{a}$
5. Nhập số nguyên X, hãy đếm xem X có bao nhiêu chữ số, in ra chữ số đầu tiên của X