NHẬP MÔN
TƯ DUY TÍNH TOÁN

Bài 2: Cơ bản về ngôn ngữ lập trình python

Nội dung trình bày

Phần 1

Biến, khai báo chuỗi, khối lệnh

Biến

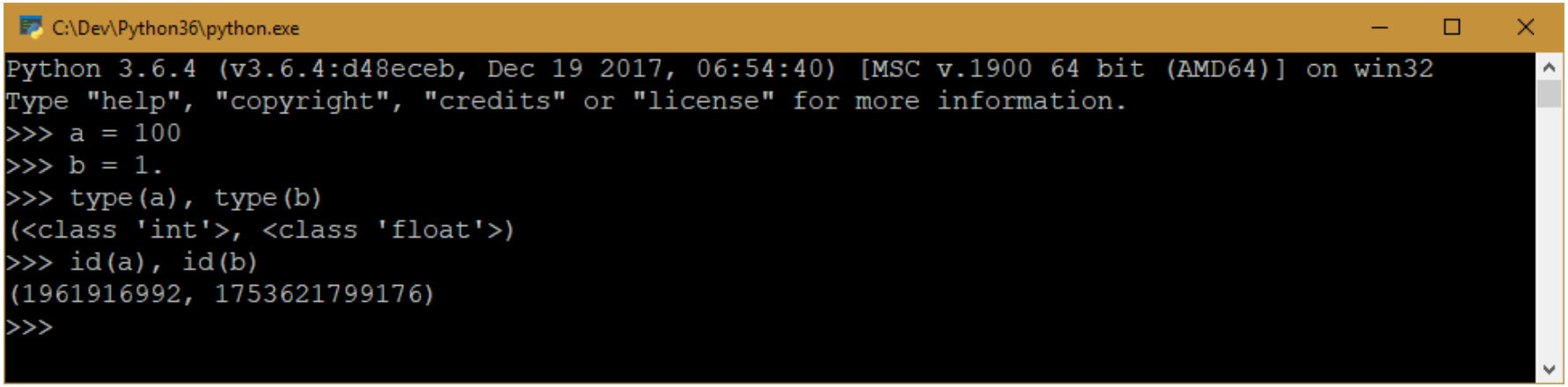
- Biến = vùng bộ nhớ được đặt tên (để dễ thao tác)
- Ví dụ:

```
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
- Chú ý: python cho phép viết ghi chú trong chương trình bằng cách đặt sau dấu thăng (#)

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 kép 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 2

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 3

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 4

Phép toán “if”

Phép toán "if"

```
# X là max của A và B
X = A if A > B else B
# N có phải là số nguyên tố có 1 chữ số hay không
A = "Đúng" if N in [2, 3, 5, 7] else "Sai"
# In ra màn hình "chẵn" nếu n chia hết cho 2,
#   in ra "lẻ" nếu ngược lại
print('chẵn' if (n % 2) == 0 else "lẻ")
# Sinh viên có được thi hay không?
print("được thi" if so_buoi_nghi < 3 else "không được thi")
# Biện luận nghiệm phương trình bậc 2 (if lồng nhau)
KQ = "một nghiệm" if delta == 0 else \
                    "vô nghiệm" if delta < 0 else "hai nghiệm"
```

Phép toán "if"

- Cú pháp: `A if <điều-kiện> else B`
- Thực hiện:
  - Phép toán trả về A nếu <điều-kiện> là đúng, ngược lại trả về B
  - A và B có thể là các giá trị, biểu thức tính toán, lời gọi hàm,…
  - Các phép toán if cũng có thể lồng nhau
- Cách sử dụng if này khá kì cục, nhưng hợp lý nếu xét về mặt ngôn ngữ và cách đọc điều kiện logic
- Bài tập: Biến X để lưu tình trạng gửi SMS, X=0 tức là chưa gửi được, X=1 tức là đã gửi thành công, X=2 tức là đã gửi và người nhận đã đọc. Viết câu lệnh sử dụng phép toán if để in ra màn hình thông báo tương ứng với giá trị của X.

Phần 5

Rẽ nhánh

Rẽ nhánh

```
# In thông báo nếu được điểm số loại giỏi
if diem >= 8:
    print("Chúc mừng bạn đã được điểm giỏi")
# In thông báo xem n chẵn hay lẻ
if (N % 2) == 0:
    print("N là số chẵn")
else:
    print("N là số lẻ")
# Biện luận nghiệm của phương trình bậc 2
if delta == 0:
    print("Phương trình có nghiệm kép")
elif delta < 0:
    print("Phương trình vô nghiệm")
else:
    print("Phương trình có hai nghiệm phân biệt")
```

Rẽ nhánh

```
if expression:
    # If-block
```

```
if expression:
    # If-block
else:
    # else-block
```

```
if expression:
    # If-block
elif 2-expression:
    # 2-if-block
elif 3-expression:
    # 3-if-block
...
elif n-expression:
    # n-if-block
```

```
if expression:
    # If-block
elif 2-expression:
    # 2-if-block
...
elif n-expression:
    # n-if-block
else:
    # else-block
```

Rẽ nhánh

- Python chỉ có một cấu trúc rẽ nhánh duy nhất, sử dụng để lựa chọn làm một trong số nhiều công việc
  - Nhiều ngôn ngữ lập trình khác sử dụng if cho trường hợp 2 lối rẽ nhánh và switch cho trường hợp nhiều lối rẽ nhánh
- Nguyên tắc với rẽ nhánh if-elif-else:
  - Biểu thức điều kiện của if và elif phải có kết quả logic
  - Hệ thống sẽ lần lượt tính giá trị từng biểu thức điều kiện từ trên xuống dưới, bắt đầu từ phát biểu if
  - Nếu biểu thức điều kiện nào đúng thì khối lệnh tương ứng được thực hiện và bỏ qua các khối lệnh khác
  - Trường hợp mọi biểu thức điều kiện đều sai, khối lệnh ứng với else sẽ được thực hiện
  - Khối else là tùy chọn, không nhất thiết phải xuất hiện

Phần 6

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 7

Bài tập

Bài tập

1. 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?
2. Nhập số nguyên n, hãy in ra n ở dạng hệ cơ số 16, hệ cơ số 8 và hệ cơ số 2
3. Nhập 2 số nguyên a và b, hãy tính và in ra giá trị của $\sqrt[b]{a}$
4. Nhập số nguyên X, hãy đếm xem X có bao nhiêu chữ số, in ra chữ số đầu tiên của X
5. (về nhà) 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