NHẬP MÔN
TƯ DUY TÍNH TOÁN

Bài 5: Kiểu tuần tự trong python, phần 2

Tóm tắt nội dung bài trước

- Có nhiều kiểu dữ liệu tuần tự trong python (string, list, tuple, range, bytes,…)
  - Chứa các dữ liệu con bên trong nó
  - Truy cập vào các dữ liệu con theo thứ tự từng phần tử một
  - Lặp for là vòng lặp phù hợp với việc xử lý dữ liệu tuần tự
- Kiểu string có thể xem như một list các str có 1 kí tự
- Các phần tử trong string được đánh chỉ mục theo 2 cách khác nhau (từ trái sang phải và từ phải sang trái)
- Python có nhiều cách định dạng string: toán tử %, dạng f-string, hàm format,…
- Python cung cấp rất nhiều hàm xử lý chuỗi giúp dễ dàng xử lý mọi nhu cầu của lập trình viên

Nội dung

1. Kiểu dữ liệu tuần tự (sequential data type)
2. String (chuỗi)
3. Bài tập về xử lý chuỗi
4. List (danh sách)
5. Tuple (hàng)
6. Range (miền)
7. Bài tập về dữ liệu tuần tự

Phần 4

List (danh sách)

Giới thiệu và khai báo

- List = dãy các đối tượng (một loại array đa năng)
- Các phần tử con trong list không nhất thiết phải cùng kiểu dữ liệu
- Khai báo trực tiếp: liệt kê các phần tử con đặt trong cặp ngoặc vuông ([]), ngăn cách bởi dấu phẩy (,)

```
[1, 2, 3, 4, 5]              # list 5 số nguyên
['a', 'b', 'c', 'd']         # list 4 chuỗi
[[1, 2], [3, 4]]             # list 2 list con
[1, 'one', [2, 'two']]       # list hỗ hợp
[]                           # list rỗng
```

- Kiểu chuỗi (str) trong python có thể xem như một list đặc biệt, bên trong gồm toàn các str độ dài 1

Khởi tạo list

- Tạo list bằng constructor

```
l1 = list([1, 2, 3, 4])    # list 4 số nguyên
l2 = list('abc')           # list 3 chuỗi con
l3 = list()                # list rỗng
```

- Tạo list bằng list comprehension: một đoạn mã ngắn trả về các phần tử thuộc list

```
# list 1000 số nguyên từ 0 đến 999
X = [n for n in range(1000)]
# list gồm 10 list con là các cặp [x, x²]
# với x chạy từ 0 đến 9
Y = [[x, x*x] for x in range(10)]
```

Phép toán, chỉ mục và cắt

- Giữa list và str có sự tương đồng nhất định
  - List cũng hỗ trợ 3 phép toán: ghép nối (+), nhân bản (*) và kiểm tra nội dung (in)
  - List sử dụng hệ thống chỉ mục và các phép cắt phần con tương tự như str
- Điểm khác biệt: nội dung của list có thể thay đổi

```
# khởi tạo list ban đầu
l1 = list([1, 2, 3, 4])
# thay đổi giá trị của phần tử cuối cùng
l1[-1] = list('abc')
# in nội dung list: [1, 2, 3, ['a', 'b', 'c']]
print(l1)
```

Các phương thức của list

- Một số phương thức thường hay sử dụng
  - count(sub, [start, [end]]): đếm số lần xuất hiện của sub
  - index(sub[, start[, end]]): tìm vị trí xuất hiện của sub, trả về ValueError nếu không tìm thấy
  - clear(): xóa trắng list
  - append(x): thêm x vào cuối list
  - extend(x): thêm các phần tử của x vào cuối list
  - insert (p, x): chèn x vào vị trí p trong list
  - pop(p): bỏ phần tử thứ p ra khỏi list (trả về giá trị của phần tử đó), nếu không chỉ định p thì lấy phần tử cuối

Các phương thức của list

- Một số phương thức thường hay sử dụng
  - copy(): tạo bản sao của list (tương tự list[:])
  - remove(x): bỏ phần tử đầu tiên trong list có giá trị x, báo lỗi ValueError nếu không tìm thấy
  - reverse(): đảo ngược các phần tử trong list
  - sort(key=None, reverse=False): mặc định là sắp xếp các phần tử từ bé đến lớn trong list bằng cách so sánh trực tiếp giá trị

```
x = "Trương Xuân Nam".split()
x.sort(key=str.lower)
print(x)
```

Ví dụ về làm việc với list

```
# tạo và in nội dung của một list
danhsach = ["apple", "banana", "cherry"]
for x in danhsach:
    print(x)
# thêm một phần tử vào cuối list và lại in ra
danhsach.append("orange")
print(danhsach)
# thêm một phần tử vào giữa list và in ra
danhsach.insert(1, "orange")
print(danhsach)
# sắp xếp lại danh sách và in ra
danhsach.sort()
print(danhsach)
```

Một số thao tác thông dụng với list

1. Tạo một list
2. Duyệt list
3. Truy cập phần tử theo chỉ số
4. Thay đổi nội dung phần tử
5. Cắt list
6. Thêm phần tử vào list
7. Bỏ phần tử khỏi list
8. Tìm kiếm phần tử trong list
9. Sắp xếp các phần tử trong list
10. List hỗ hợp

Phần 5

Tuple (hàng)

Tuple là một dạng readonly list

- Tuple = dãy các đối tượng (list), nhưng không thể bị thay đổi giá trị trong quá trình tính toán
- Như vậy str giống tuple nhiều hơn list
- Khai báo trực tiếp bằng cách liệt kê các phần tử con đặt trong cặp ngoặc tròn (), ngăn cách bởi phẩy

```
(1, 2, 3, 4, 5)              # tuple 5 số nguyên
('a', 'b', 'c', 'd')         # tuple 4 chuỗi
(1, 'one', [2, 'two'])       # tuple hỗ hợp
(1,)                         # tuple 1 phần tử
()                           # tuple rỗng
```

Tuple và list nhiều điểm giống nhau

- Tuple có thể tạo bằng constructor hoặc tuple comprehension
- Tuple hỗ trợ 3 phép toán: +, *, in
- Tuple cho phép sử dụng chỉ mục và cắt
- Các phương thức thường dùng của tuple
  - count(v): đếm số lần xuất hiện của v trong tuple
  - index(sub[, start[, end]]): tương tự như str và list
- Tuple khác list ở điểm nào?
  - Chiếm ít bộ nhớ hơn
  - Nhanh hơn

Phần 6

Range (miền)

Range là một tuple đặc biệt?

- Chúng ta đã làm quen với range khi dùng vòng for
  - range(stop): tạo miền từ 0 đến stop-1
  - range(start, stop[, step]): tạo miền từ start đến stop-1, với bước nhảy là step
    - Nếu không chỉ định thì step = 1
    - Nếu step là số âm sẽ tạo miền đếm giảm dần (start > stop)
- Vậy range khác gì một tuple đặc biệt
  - Range chỉ chứa số nguyên
  - Range nhanh hơn rất nhiều
  - Range chiếm ít bộ nhớ hơn
  - Range vẫn hỗ trợ chỉ mục và cắt (nhưng khá đặc biệt)

Phần 7

Bài tập

Bài tập

1. Người dùng nhập từ bàn phím liên tiếp các từ tiếng Anh viết tách nhau bởi dấu cách. Hãy nhập chuỗi đầu vào và tách thành các từ sau đó in ra màn hình các từ đó theo thứ tự từ điển.
2. Người dùng nhập từ bàn phím chuỗi các số nhị phân viết liên tiếp được nối nhau bởi dấu phẩy. Hãy nhập chuỗi đầu vào sau đó in ra những giá trị được nhập.
3. Nhập số n, in ra màn hình các số nguyên dương nhỏ hơn n có tổng các ước số lớn hơn chính nó.
4. Nhập vào một chuỗi từ người dùng, kiểm tra xem đó có phải địa chỉ email hợp lệ hay không?

Bài tập

5. Hãy nhập số nguyên n, tạo một list gồm các số fibonacci nhỏ hơn n và in ra
   - Dãy fibonacci là dãy số nguyên được định nghĩa một cách đệ quy như sau: f(0)=0, f(1) = 1, f(1<n) = f(n-1) + f(n-2)
6. Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu
   - Số nguyên tố là số tự nhiên có 2 ước số là 1 và chính nó.
7. Nhập n, in n dòng đầu tiên của tam giác pascal
8. Liệt kê các chuỗi độ dài ít hơn 20 của tuple X gồm các chuỗi được định nghĩa như sau:
   - Chuỗi M = '()' thuộc X
   - Nếu chuỗi A thuộc X thì chuỗi (A) cũng thuộc X
   - Nếu chuỗi A và B thuộc X thì chuỗi AB cũng thuộc X