List, Tuple, Dictionary, String

Th.S Trần Đức Lợi
Pythonvietnam.info

Ôn tập bài cũ

- Chữa bài python-stickers

Mục đích bài học

- Tìm hiểu về các kiểu dữ liệu list, tuple, dictionary và một số phần mở rộng của string

Tập hợp

- Một tập hợp là giống như một vali xách đồ khi thực hiện đi du lịch
- Ta có thể có nhiều giá trị trong một tập hợp tồn tại dưới dạng một biến
- Có thể tìm kiếm và thao tác với từng phần tử trong tập hợp

Tập hợp

- X = 2
- X =4
- X không phải là một tập hợp vì nó chỉ có một giá trị và khi thêm 1 giá trị khác vào giá trị cũ bị ghi đè

Dictionary

- Dictionary là **tập hợp** của các phần tử mà mỗi phần tử bao gồm key và value.
- Java: Hashmap, Properties
- Một danh bạ với tên contact và thông tin contact
- Một túi đồ với các đồ được để **lộn xộn** nhưng đều **dán nhãn**

Dictionary: Cú pháp

- **Cú pháp**:
- *D = {key1: value1, key2: value2}*
  - *D={'x':1, 'y':2}*
- Có thể khai báo một dictionary rỗng:
  - D = {}
- Yêu cầu đối với key: là **immutable object**
  - Thử với trường hợp gán key là biến

Dictionary

- Dictionary là tập hợp dữ liệu mạnh nhất trong python so với list, tuple và string
- Cho phép có các thao tác tương tự database
- Không có sắp xếp dữ liệu mà dựa hoàn toàn vào nhãn (khác với list)

Dictionary

- Truy xuất dữ liệu:
  - D[‘key’]
- Sẽ phát sinh **lỗi** nếu truy xuất đến một **key** chưa có trong dictionary
- Có thể sử dụng toán tử **in** để kiểm tra xem key đã có trong dictionary chưa
  - ‘key’ in D

Dictionary

- Bài tập ứng dụng:
  - Hãy sử dụng dictionary để **đếm** số lần xuất hiện của các từ trong 10 lần nhập của người dùng

Dictionary: get()

- Kiểm tra xem một key đã có trong dict chưa đồng thời tự động gán giá trị mặc định cho key nếu như key này chưa có trong dict
- D.get('key', 'dval')

Dictionary

- Bài tập ứng dụng:
  - Hãy sử dụng dictionary để **đếm** số lần xuất hiện của các từ trong 10 lần nhập của người dùng

Dictionary và vòng lặp

- Yêu cầu: Cần in tất cả các giá trị của các phần tử trong dictionary?
  - Chương trình cần đi qua tất cả các key của dict
  - Với mỗi key thực hiện in value tương ứng
- For key in dict:
  - Print key, dict[key]

Dictionary: items()

- Trả về một list các phần tử dưới dạng tuples của một dictionary
- Ví dụ:
  - D={‘x’:1}
  - Print D.items()

Dictionary: items và loop

- Dùng vòng lặp nhưng với cả hai tham số key, value
- For key,val in d.items():
  - Print key, val

List: Danh sách

- Định nghĩa một list bằng dấu []
- Các thành viên trong list có thể là bất cứ object nào kể cả một list khác
- Có thể tạo một list rỗng
- A = []
- B = [1,2,3]

List & Loop

- For I in [1,2,3,4,5]:
  - Print I
- Print ‘ended.’
- For x in [‘johnny’, ‘loitd’]:
  - Print x

List

- Truy xuất dữ liệu trong list với dấu [index]
- Ví dụ: list[1]
- List = [‘Loitd’, ‘python’,’vietnam’]
- Giống với string, list đánh index từ 0

List: Mutable

- String là immutable
- Ta phải tạo một string mới nếu muốn có sự thay đổi
- List là mutable
- Có thể thay đổi các phần tử của list bằng cách truy xuất và gán giá trị
- Ví dụ:
  - X = ‘abc’
  - X[0] = ‘d’
  - X = [‘a’,’b’,’c’]
  - X[0] = ‘d’

List: len()

- Để xác định độ dài của list sử dụng hàm len(list)
- Len() cũng áp dụng cả với string

List: range()

- Hàm range(x) trả về 1 list các số từ 0 đến x-1
- Bài tập:
  - Viết lại ví dụ với for loop sử dụng hàm range() và len()

List

- Chúng ta có thể nối hai chuỗi bằng toán tử +
- List1 = [1,2,3]
- List2 = [4,5,6]
- List1 + list2 = list3

Cắt một list sử dụng **:**

- T[a:b] được hiểu là: lấy từ phần tử thứ **a** đến phần tử **b-1**
- T[1:3]
- T[:4]
- T[2:]
- T[:]

List: append()

- Có thể xây dựng list từ list rỗng và hàm append() để thêm các phần tử vào list
- Append() sẽ thêm các phần tử vào **cuối** list
- Ví dụ
  - A = list()
  - A.append('x')
  - A.append(1)
  - Print A

List: kiểm tra phần tử

- Kiểm tra một phần tử có hay không thuộc list ta sử dụng **in** và **not in**?
- Ví dụ:
  - A = [1,2,3,4,5]
  - 1 in A
  - 6 not in A

List: Ordered

- Phương thức append() sẽ thêm các phần tử mới vào cuối list
- Một list có thể đước sắp xếp lại trật tự với phương thức list.sort()
- Ví dụ:
  - L = [1,2,4,3]
  - L.sort()

List: Một số hàm lấy list làm tham số

- Đây là các hàm built-in:
  - Len(l)
  - Max(l)
  - Min(l)
  - Sum(l)
- Bài tập:
  - Hãy tính giá trị trung bình của n số do người dùng nhập vào đến khi người dùng nhập “over”?

String: split() into list

- Hàm split() của một chuỗi sẽ trả về 1 list
- Chỉ định delimiter cho split()
- Ví dụ:
  - Str = “cuoc doi tuoi dep”
  - Str.split()
  - Str = “cuoc,doi,tuoi,dep”
  - Str.split()?

String: split nhiều lần

- Có chuỗi sau:
  - “fromXwithLove: I love Python. Send email to loitd@pythonvietnam.info”
- Bài tập:
  - Hãy tách chuỗi này để đưa ra được tên người gửi, thông điệp, và email người nhận.

String: rstrip()

- Cắt các ký tự trắng ở trái (lstrip) và phải (rstrip) chuỗi
- Ví dụ:
  - “ I love emin ”.rstrip()
  - “ I love emin ”.lstrip()
  - “ I love emin ”.strip()

String: startswith()

- Kiểm tra các ký tự bắt đầu của một chuỗi.
- Ví dụ:
  - “important: I love you, Loi!”.startswith(‘important:’)
- Bài tập:
  - Viết chương trình đọc chuỗi nhập vào từ người dùng và chỉ xử lý in ra những chuỗi nào bắt đầu với từ “chatmsg:”, thoát chương trình bằng phím “q”

Tuples

- Tuples là một dạng tập hợp có cách hoạt động tương tự như list() trong đó đánh index từ 0
- Tuples được đặt trong dấu () thay vì [] như list
- Ví dụ:
  - T = (‘x’, ‘y’, 1,2)
  - Print t[1]

Tuples: immutable

- Tuples là kiểu dữ liệu immutable giống như string
- Đây là điểm khác biệt giữa list và tuple
- Ví dụ:
  - T[1] = ‘xxxxx’?

Tuples: immutable

- Vì dữ liệu không thể thay thế nên có một số phương thức không thể thực hiện với tuples
  - Sort()
  - Append()
  - Reverse()
- Lệnh dir() với list và tuple

Gán giá trị với tuples

- Xem xét ví dụ:
  - (x,y) = (123,456)
  - Print x
  - Print y
  - X,y = 123,567
  - Print x?
  - Print y?

Tuples: phương thức items của dictionary

- For (key,val) in dict.items():
  - Print key, val
- For t in dict.items()
  - Print t

So sánh tuples

- Toán tử so sánh có làm việc với tuples và các dạng dữ liệu liệt kê
- Nguyên tắc so sánh giữa tuples
- Ví dụ:
  - (0,1,100) **?** (0,2,1)
  - (‘a’,’c’,1) **?** (‘a’,’b’,1)

List of tuples: Sort()

- L = [('a',1), ('c',2), ('b',3)]
- L.sort()
- Dictionary -> list of tuples -> sorted

Sorted()

- Sắp xếp dictionary theo key hay value?
- D = {‘1’:’c’, ‘A’:’d’, ‘a’:’b’}
- Sắp xếp theo bất kỳ tiêu chí nào?
  - Dictionary -> list of tuples -> append in reverse order -> sorted

.sort() vs sorted()

- List.sort() vs sorted(iterable)
- D = {‘1’:’c’, ‘A’:’d’, ‘a’:’b’}
  - D.sort()?
  - Sorted(d, key=str.lower)
  - Sorted(d, key=str.lower, reverse=True)
- D2 = {1:’c’, ‘A’:’d’, ‘a’:’b’}
  - Sorted(d, key=str.lower, reverse=True)?
  - List.sort(key=str.lower, reverse=True)?

Sorted()

- T = [(‘a’, 10, ‘pop’), (‘b’, 2, ‘rock’), (‘c’, 5, ‘country’)]
  - T.sort()
  - T.sort(reverse=True)
  - T.sort(key=lambda T:T[1])
  - From operator import itemgetter
  - T.sort(key=itemgetter(2))

Tuples

- Tại sao python tạo ra tuples?
- Các ngôn ngữ khác có dạng tuples?
- Sự ưu việt giữa tuples với list ở khía cạnh nào?

Tổng kết bài học

- Tập hợp
- Dictionary
- List
- Tuple

Bài tập

- Chương trình ***WordsCounting***
- Chương trình ***Quản lý học sinh***