

# List, Tuple, Dictionary, String



TH.S TRẦN ĐỨC LỢI  
PYTHONVIETNAM.INFO

# Ôn tập bài cũ



- Điều khiển luồng
- Vòng lặp
- Ngoại lệ

# Mục đích bài học



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

# Tập hợp



Có thể thao tác với  
từng phần tử trong  
vali

Vali có thể chứa  
nhiều loại đồ đạc  
khác nhau

# 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

# Dictionary



Một danh bạ với  
tên contact và  
thông tin contact



# Dictionary



Đồ đạc 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: 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



- List là một kiểu dữ liệu tập hợp trong python.
- Định nghĩa một list bằng dấu []
- Có thể tạo một list rỗng
- `A = []`
- `B = [1,2,3]`

# List



## TO DO LIST

☐☐☐☐☐

# List



Sổ hồng/nhãn:  
môn văn

Khe số n/bút  
cam

Khe số  
1/bút hồng

# 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



List

mutable

Gán giá trị cho  
từng phần tử

String

immutable

Không thay đổi  
được giá trị  
từng phần tử

Dictionary?

# List: Mutable



- 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 cuối list

- Ví dụ

```
A = list()
```

```
A.append('x')
```

```
A.append(1)
```

```
Print A
```

# List: append()



append()



# 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 được 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)



# 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: 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()



- `Str.startswith('key', begin=0, end=len(str))`
- Kiểm tra các ký tự bắt đầu của một chuỗi.
- Ví dụ:
  - `"important: I love you, Loi!".startswith('important:')`

# String: endswith()



- `Str.startswith('key', begin=0, end=len(str))`
- Kiểm tra các ký tự kết thúc của một chuỗi.
- Trả về True/False

# String: find()



- `.find('key', begin=0, end=len(str))`
- Kiểm tra sự xuất hiện của chuỗi trong chuỗi lớn
- Trả về giá trị index hoặc -1

# String: Replace()



- `.replace('old', 'nouvou', max=2)`
- Thay thế chuỗi 'old' bằng chuỗi 'mouvou' tối đa 2 lần trong chuỗi nếu chỉ định max.

# String: join()



- `.join(list_strings_to_join)`
- Nối tất cả các chuỗi trong list với nhau

# String: Kiểm tra



- **.isalnum()**
  - Ít nhất 1 ký tự và là alphanumeric (chữ hoặc số)
- **.isalpha()**
  - Ít nhất 1 ký tự và là alphabetic (chữ)
- **.isnumeric()**
  - Nếu chuỗi **Unicode** chỉ gồm các ký tự số
- **.isdigit()**
  - Nếu chuỗi chỉ gồm số
- **.isspace()**
- **.islower()**
- **.isupper()**
- Trả về True/False



# String: Print a string



- Print “I am %s and I am %d”%(‘Loi’, 30)

Format	Ý nghĩa
%c	character
%s	string conversion via str() prior to formatting
%i	signed decimal integer
%d	signed decimal integer
%u	unsigned decimal integer
%o	octal integer
%f	floating point real number

# 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
- Ví dụ:
  - `T[1] = 'xxxxx'?`

# Tuple



List

mutable

Gán giá trị cho  
từng phần tử

String

immutable

Không thay đổi  
được giá trị  
từng phần tử

Dictionary?

Tuple

# Tuples: immutable



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

# 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