

Buổi 3: Python trong Khoa học dữ liệu #2

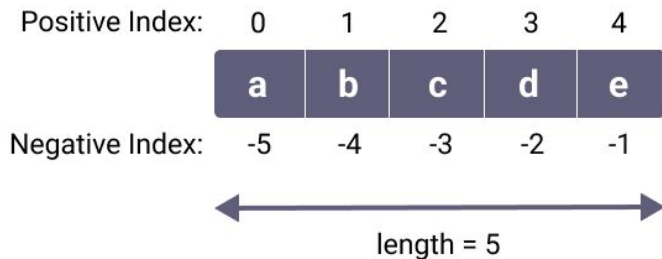
- List
- Tuple
- Dictionary
- Set
- Summary
- Game Lô Đề Học Pro
- Tổng kết

What?

- List hay còn gọi là danh sách.
- Là một dãy có thể thay đổi:
- Linh hoạt, được sử dụng rộng rãi.
- Khai báo cách nhau bằng dấu phẩy.

```
>>> names = [ "MetaMind", "metamind" ]
```

`list = ['a', 'b', 'c', 'd', 'e']`



List Document

```
>>> names = [ "MetaMind", "metamind" ]
```

- (+) thêm các phần tử vào cuối, giống như .extend()

```
>>> names += ["MtM"]
```

```
>>> names
```

```
['MetaMind', 'metamind', 'MtM']
```

- .extend() mở rộng list bằng cách thêm các phần tử từ một iterable khác.

```
>>> names.extend(["MtM", "MetaMine"])
```

```
>>> names
```

```
['MetaMind', 'metamind', 'MtM', 'MetaMine']
```

- .append() thêm một phần tử vào cuối danh sách.

```
>>> names.append("MtM")
```

```
>>> names
```

```
['MetaMind', 'metamind', 'MtM']
```

- Lưu ý, nếu dùng extend()

```
>>> names.extend("MtM")
```

```
>>> names
```

```
['MetaMind', 'metamind', 'M', 't', 'M']
```

```
>>> names = [ "MetaMind", "metamind", "MetaMine"]
```

- (=) thay thế một giá trị trong danh sách.

```
>>> names[0]= ["MtM"]
```

```
>>> names
```

```
['MtM', 'metamind', 'MetaMine']
```

- (=) thay thế một loạt giá trị trong danh sách.

```
>>> names[1:3] = ["metamy", "MetaMai"]
```

```
>>> names
```

```
['MtM', 'metamy', 'MetaMai']
```

- (+=) thêm một phần tử vào giữa, giống như .insert().

```
>>> names[1:1] += ["M"]
```

```
>>> names
```

```
['MetaMind', 'M', 'metamind', 'MtM']
```

- .sort() sắp xếp phần tử.

```
>>> names.sort()
```

```
>>> names
```

```
['M', 'MetaMind', 'MtM', 'metamind']
```

- .del() xóa phần tử.

```
>>> del.names(1)
```

```
>>> names
```

```
['M', 'MtM', 'metamind']
```

- .remove() xóa một phần tử cụ thể, nếu có 2 phần tử giống nhau sẽ xóa 1 cái đầu tiên.

```
>>> names.remove("metamind")
```

```
>>> names
```

```
['M', 'MtM']
```

What?

- Là một dãy không thay đổi (Immutable).
- Có thể chứa bất kỳ loại phần tử nào.
- Một cách sử dụng phổ biến của tuples là biểu diễn đơn giản của các cặp:
 - Vị trí (x, y)
 - Kích thước (w, h)
 - ...

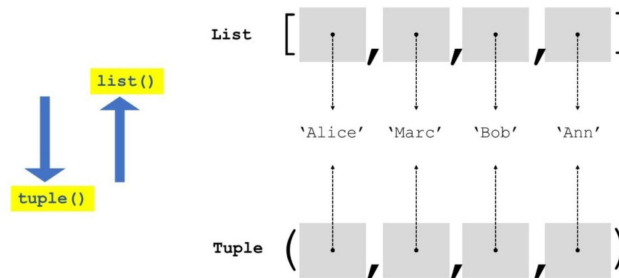
#Tọa độ (x, y)

```
>>> coordinate = 10, 20  
>>> coordinate  
(10, 20)
```

#Kích thước (w, h)

```
>>> size = 30, 40  
>>> size  
(30, 40)
```

Python Convert List → Tuple and Tuple → List



Tuple Document

#Khác: Thông tin sản phẩm (tên, giá, số lượng)

```
>>> product_info = ("Áo", 150000, 50)  
>>> product_info  
(Áo, 150000, 50)
```

- Tạo tuple bằng cách sử dụng dấu phẩy

```
>>> p = 10, 20
```

```
>>> p  
(10, 20)
```

```
>>> p = (20, 40)
```

```
>>> p  
(20, 40)
```

```
>>> type(p)  
<class 'tuple'>
```

```
>>> p[1]=1
```

Traceback (most recent call last): File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does **not** support item assign

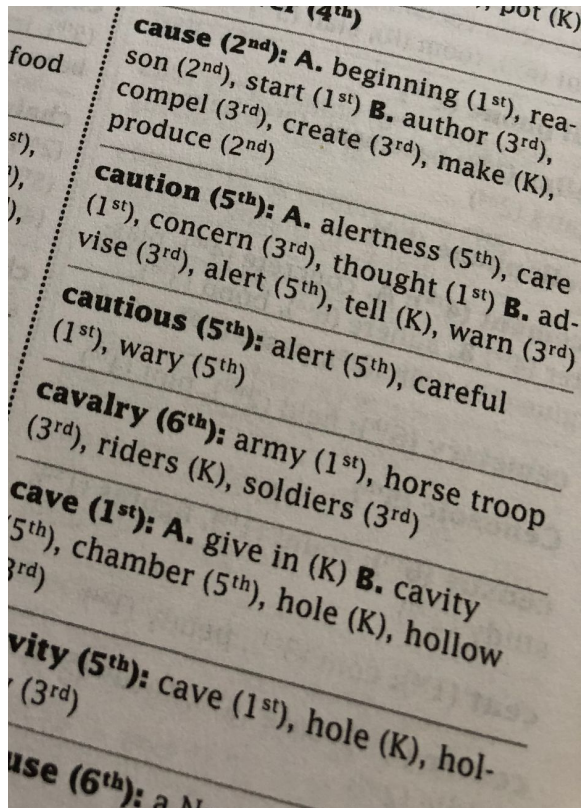
Operations	examples	description
Creating a tuple	>>>a=(20,40,60,"apple","ball")	Creating the tuple with elements of different data types.
Indexing	>>>print(a[0]) 20 >>> a[2] 60	Accessing the item in the position 0 Accessing the item in the position 2
Slicing	>>>print(a[1:3]) (40,60)	Displaying items from 1st till 2nd.
Concatenation	>>> b=(2,4) >>>print(a+b) >>>(20,40,60,"apple","ball",2,4)	Adding tuple elements at the end of another tuple elements
Repetition	>>>print(b*2) >>>(2,4,2,4)	repeating the tuple in n no of times
Membership	>>> a=(2,3,4,5,6,7,8,9,10) >>> 5 in a True >>> 100 in a False >>> 2 not in a False	Returns True if element is present in tuple. Otherwise returns false.
Comparison	>>> a=(2,3,4,5,6,7,8,9,10) >>>b=(2,3,4) >>> a==b False >>> a!=b True	Returns True if all elements in both elements are same. Otherwise returns false

What?

- Dictionary - từ điển.
- Được sử dụng để ánh xạ các giá trị (values) tới các khóa (keys) tương ứng, → Cặp key-value.
- Mỗi key phải là duy nhất và dùng để truy cập các value tương ứng.
- Value có thể là bất kỳ loại dữ liệu nào (int, string, list, tuple, dict, ...).
- Cho phép thêm, cập nhật hoặc xóa các cặp khóa-giá trị.
- Khởi tạo một từ điển bằng cặp ngoặc nhọn (`{}`) và cặp khóa-giá trị được phân tách bằng dấu hai chấm (`:`).

```
my_dict = {"name": "MetaMind"}
```

Dictionary Document



How?

```
my_dict = {"name": "MetaMind"}
```

- Truy cập value thông qua key:

```
my_dict["name"] → MetaMind
```

- Thêm một cặp key-value mới:

```
my_dict["courses"] = 30 → my_dict = {"name": "MetaMind", "courses": "30"}
```

- Cập nhật một khóa đã tồn tại:

```
my_dict["courses"] = 50 → my_dict = {"name": "MetaMind", "courses": "50"}
```

- Xoá một cặp key-value:

```
del my_dict["courses"] → my_dict = {"name": "MetaMind"}
```

What?

- Set - tập hợp: là một cấu trúc dữ liệu **không thể thao tác sắp xếp, không có thứ tự, không thể lập chỉ mục** và **không chứa các phần tử trùng lặp**.
- Set được sử dụng để lưu trữ một tập hợp các giá trị duy nhất.
- Các phần tử trong set có thể là bất kỳ loại dữ liệu không thay đổi nào.
- Set hỗ trợ các phép toán set như hợp (union), giao (intersection), hiệu (difference) và đối xứng (symmetric).
- Để khai báo một set, bạn sử dụng cặp dấu ngoặc nhọn **{ }** với **một giá trị bên trong** hoặc hàm **set()** với các giá trị được phân tách bằng dấu **phẩy**,

Set Document

How? `my_set = {'M', 'e', 't', 'a'}`

- Không thể truy cập bằng chỉ mục với Set:

`my_set[0] → TypeError: 'set' object is not subscriptable`

- Thêm một phần tử mới:

`my_set.add(50) → my_set = {'M', 'e', 't', 'a', 50}`

- Xoá một phần tử:

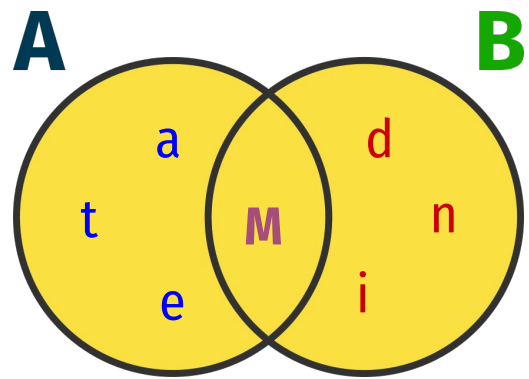
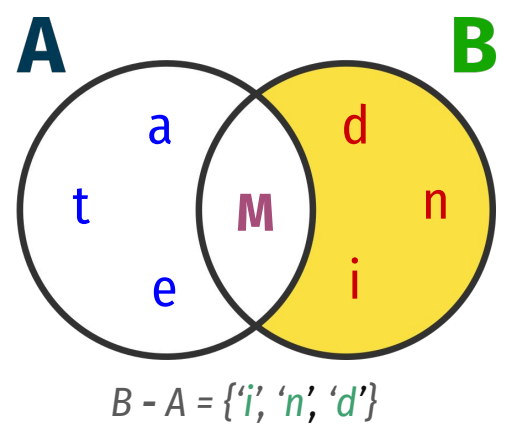
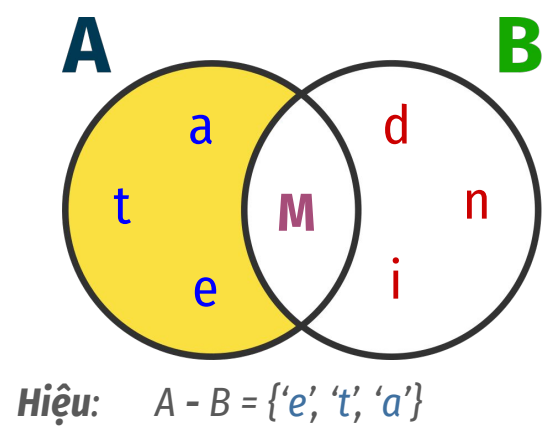
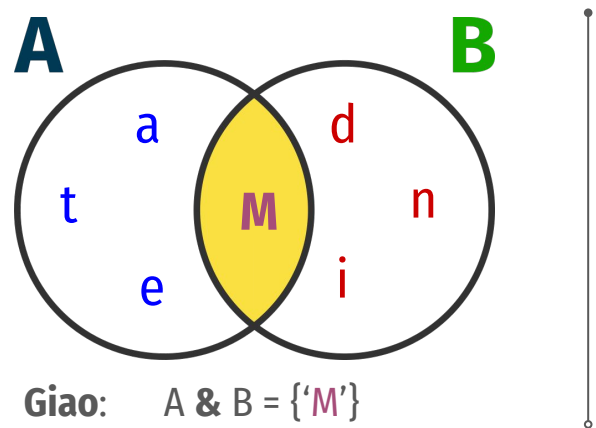
`my_set.remove(50) → my_set = {'M', 'e', 't', 'a'}`

- Phép toán với Set: (→ Slide kế tiếp →)

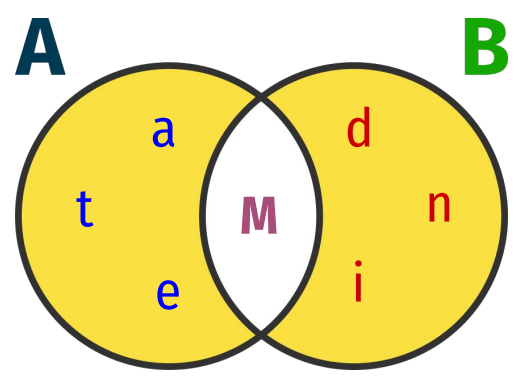
SET / Các phép toán với Set

$A = \{ 'M', 'e', 't', 'a' \}$

$B = \{ 'M', 'i', 'n', 'd' \}$



Hợp: $A \cup B = \{ 'M', 'e', 't', 'a', 'i', 'n', 'd' \}$



Đối xứng: $A \Delta B = \{ 'e', 't', 'a', 'i', 'n', 'd' \}$

SUMMARY

Tiêu chí	Lists	Tuples	Sets	Dictionary
Khái niệm	Lists là một danh sách chứa các phần tử có thể sắp xếp, có thể thay đổi các phần tử.	Tuple là một tập hợp có thứ tự, không thể thay đổi các phần tử.	Sets là một tập hợp không có thứ tự của các phần tử duy nhất.	Dictionary là một tập hợp không có thứ tự của các cặp khóa-giá trị.
Cú pháp	Cú pháp bao gồm dấu ngoặc vuông [,] với dữ liệu được phân tách bằng dấu “.	Cú pháp bao gồm dấu ngoặc đơn (,) với dữ liệu được phân tách bằng dấu “.	Cú pháp bao gồm dấu ngoặc nhọn { , } với dữ liệu được phân tách bằng dấu “.	Cú pháp bao gồm dấu ngoặc nhọn { , } với dữ liệu được phân tách bằng dấu “, cho cặp khóa-giá trị.
Tạo	Một danh sách có thể được tạo bằng cách sử dụng hàm list() hoặc gán đơn giản cho [].	Tuple có thể được tạo bằng cách sử dụng hàm tuple().	Một tập hợp có thể được tạo bằng cách sử dụng hàm set().	Một Dictionary có thể được tạo bằng cách sử dụng hàm dict().
Thứ tự	Có	Có	Không	Không
Chỉ mục	Có chỉ mục dựa trên số nguyên bắt đầu từ '0'.	Cũng có chỉ mục dựa trên số nguyên bắt đầu từ '0'.	Không có cơ chế chỉ mục.	Có một cơ chế chỉ mục dựa trên khóa, tức là các khóa xác định giá trị.
Có thể sửa đổi?	Có	Không	Không	Có
Cho phép trùng lặp?	Có	Có	Không	Có (chỉ giá trị)
Đặc điểm duy nhất	-	Có thể là một khóa Dictionary	Các phép toán toán học	Cặp khóa-giá trị
Biểu diễn	[1, 2, 3]	(1, 2, 3)	{1, 2, 3}	{'a': 1, 'b': 2, 'c': 3}

Đề bài: Từ Game Lô Đề Basic. Bạn hãy nâng cấp Game Lô Đề Pro với yêu cầu như sau:

- **Bổ sung chức năng đăng nhập:**

- Khi chạy chương trình, màn hình đăng nhập sẽ hiển thị ra đầu tiên. Nếu người dùng nhập đúng tài khoản và mật khẩu thì mới được vào màn hình chính (tài khoản mật khẩu mặc định là admin - admin)
- Nếu người dùng nhập sai mật khẩu, yêu cầu nhập lại. Người dùng chỉ cần đăng nhập 1 lần cho đến khi nào đăng xuất mới phải đăng nhập lại.

- **Chức năng chơi lô:**

- Cho phép người dùng nhập nhiều số lô, mỗi số lô cách nhau bằng dấu “,” (VD: 12,23)
- Số giải sẽ có tổng cộng 5 chữ số và có thể có cả chữ số 0 ở đầu
- Các số quay được sẽ truyền vào Dictionary với Key là tên giải, Value là con số quay được. Điều kiện trúng lô vẫn giữ nguyên nhưng tổng tiền lúc này sẽ = tiền cược * (số lô trúng * 70) - tiền cược số lô trượt. Bạn cũng cần in ra những số lô nào là số lô trúng.

- **Chức năng đăng xuất:**

- Sửa chức năng Thoát thành chức năng đăng xuất. Người dùng đăng xuất xong nếu chơi lại game sẽ phải đăng nhập lại từ đầu.

Hoàn thành Form điểm danh sau (10'): [Form điểm danh](#)

Bài tập sau buổi học:

- [List](#)
- [Tuple](#)
- [Dictionary](#)
- [Set](#)
- [Bài tập tổng hợp](#)
- [Game Lô Đề Pro](#)

Yêu cầu:

- Đặt tên folder theo mẫu DSMMC_<Buổi học>_<Tên> (Ví dụ: *DSMMC_Buoi3_DangNH*)
- Hoàn thành các bài lab sau đó nộp lên Git cá nhân, rồi submit link folder git trên [Form bài tập](#)
- Hạn nộp: Trước ngày 14/04/2024

THANK YOU !