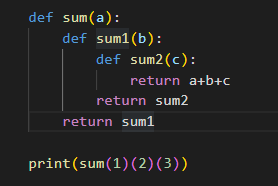
|  |  |  |  |
| --- | --- | --- | --- |
| **List** | **Tuple** | **Set** | **Dict** |
| **- Ordered**  **- Changeable**  **- Duplicate** | **- Ordered**  **- Unchangeable**  **- Duplicate** | **- Unordered**  **- Unchangeable**  **- No Duplicate** | **- Ordered**  **- Changeable**  **- Duplicate (no with key, yes with value, update to last key)** |
| * **Unorder:** cannot be referred to by index or key. * **Unchangeable:** cannot change the items after being created * **No Duplicate:** cannot have two items with the same value. | | | |

1. **Difference between data structures**
2. **Function:**

* Khi khai báo 1 function mà return về 1 hàm khác thì khi ta truyền tham số thì ta có thể truyền số tham số tương ứng với số lượng hàm đã khai báo.



* Hoặc chúng ta có thể dùng lambda:

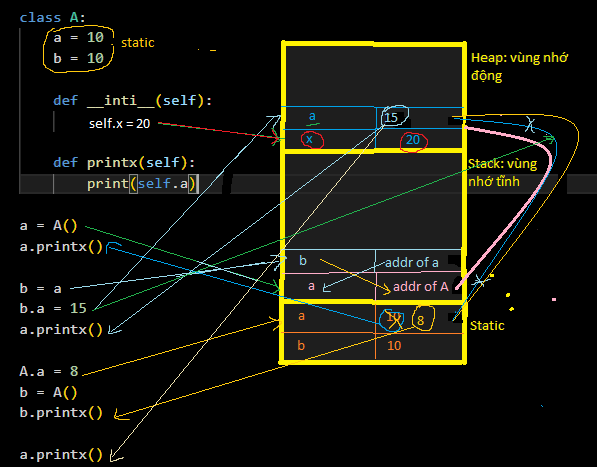


1. **Try/Except:**

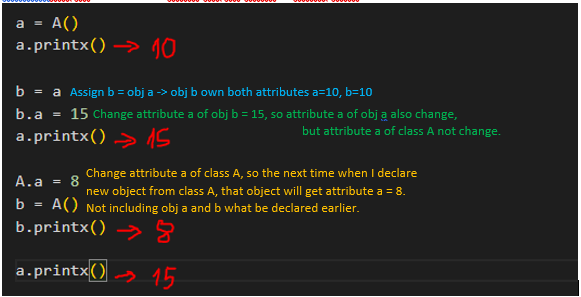
* Try dùng để tìm lỗi, except dùng để xử lý lỗi, nếu lệnh trong except báo sai thì hệ thống báo sai luôn.
* Xóa file/folder: os.remove(‘file.txt’), dùng os.rmdir(folder).
* Không thể xem string là list vì string is immutable, còn list is mutable.

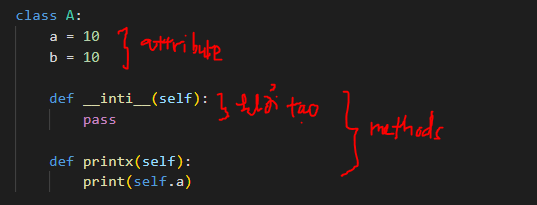
1. **OOP**

* Class: là một bản mô tả bao gồm các phương thức(methods) và đặc tính(attributes). Trong đó đặc tính thì giống như biến, còn phương thức thì giống như hàm, nhưng khác với biến và hàm là các attribute và method khai báo trong class thì không chiếm bộ nhớ, còn hàm và biến khai báo ngoài hàm main thì chiếm bộ nhớ.
* Object (Instance): là đối tượng được gọi từ Class, chỉ khi nào ta gọi một object từ Class thì lúc này các attributes và methods mới được cấp phát bộ nhớ.



* Vùng nhớ Tĩnh (Stack) là vùng nhớ mà khi khai báo object hoặc là các biến, hàm ngoài hàm main.
* Vùng nhớ động (Heap) là vùng nhớ chứa địac chỉ của attributes và methods của Class, giá trị attributes thực sự sẽ nằm ở vùng nhớ Static.

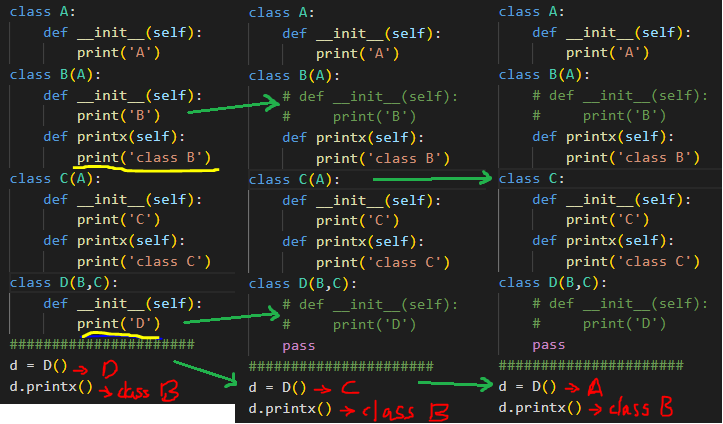


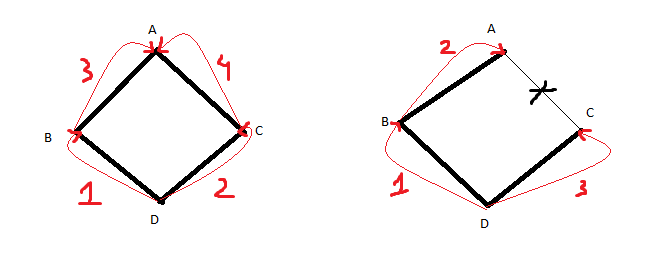


* Method: Trong Class có các hàm mà chúng ta gọi là method, trong đó có 1 hàm \_\_init\_\_(self) là hàm khởi tạo. Khi chúng ta tạo một object từ class thì nó sẽ thực hiện hàm khởi tạo đầu tiền, khi đó nó sẽ khởi tạo tất cả những attriute mà mình đã khai báo trong class.
* Attributes: Khi ta tạo 1 object a từ class A, thì lúc này object a sẽ trỏ tới địa chỉ của class A, nơi chứa những attributes và method.

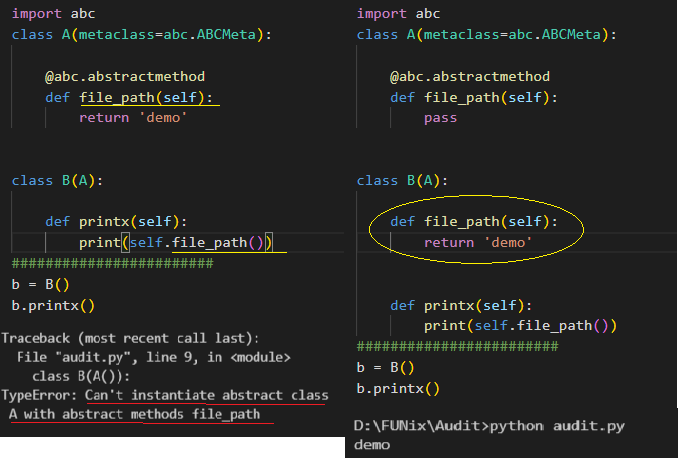
**Conclusion:** Nên hạn chế khai báo attribute trong class ở ngoài (tức vùng static), vì như thế ta rất khó kiểm soát và quản lý.

* **Inheritance: Kế thừa**
* Nếu hai hàm cùng tên, thì nó sẽ sử dụng class được khai báo thừa kế trước, nó có xu hướng duyệt tới class gốc(A) theo thuật toán BFS nếu vòng kín, còn DFS nếu vòng không kín.

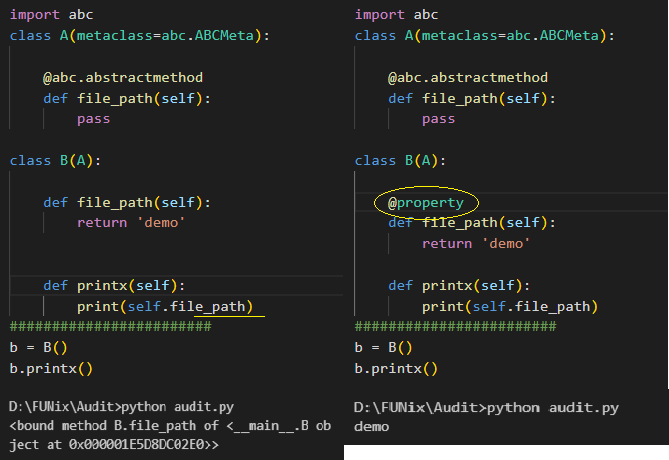




* **Abstractmethod**: bắt những Class nào kế thừa class đó phải định nghĩa lại hàm đó. Mục đích của ABC metaclass là giúp chúng ta phát hiện ra lỗ hổng trong quá trình thực thi.



* **@property** – là hình thức của hàm get, vì thế nó sẽ trả về attribute là giá trị trả về của hàm ngay sau nó, nôm na là sử dụng 1 hàm như là 1 biến.



1. **Pointer in Python**

* Phép gán ‘=’ trong python sẽ tư động tạo ra 1 pointer hiệu quả - ngoại trừ basic type (called an immutable): **int, float, str, bool** là kiểu basic types, và đây không là con trỏ của bất kỳ mô tả nào.
* **Lists, dicts, class objects etc** là những con trỏ hiệu quả (effectivey pointer).

