

Bài 11 : Class

Class, là một đối tượng trong Python. Nó là một tập hợp của các dữ liệu mình đề ra. Mình có thể lấy thông tin từ nó và chỉnh thông tin của nó.

Class có thể chứa được các thể loại dữ liệu và biến. Còn có thể chứa được riêng các hàm.

Sau khi tạo class thì ta phải có thêm một hàm `__init__` khởi đầu class đó. Bạn có thể làm như thế này để tạo một class :

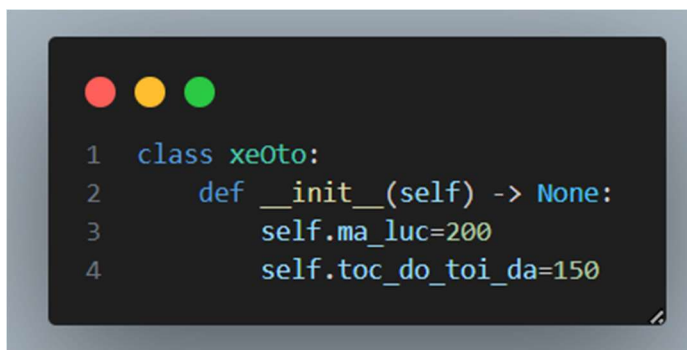


```
1 class xeOto:
2     def __init__(self) -> None:
3         pass
```

Những dòng code trên sẽ tạo ra cho bạn một class có tên là xeOto và trong class đó chưa có gì cả.

Sau khi bạn tạo một class và có cả hàm `__init__` thì bạn có thể tạo ra các thông số trong hàm `__init__` của class đó bằng cách gõ : `self.<tên thông số>`.

VD :



```
1 class xeOto:
2     def __init__(self) -> None:
3         self.ma_luc=200
4         self.toc_do_toi_da=150
```

Những dòng mã trên sẽ tạo ra 2 thông số cho class xeOto và 2 thông số đó là :

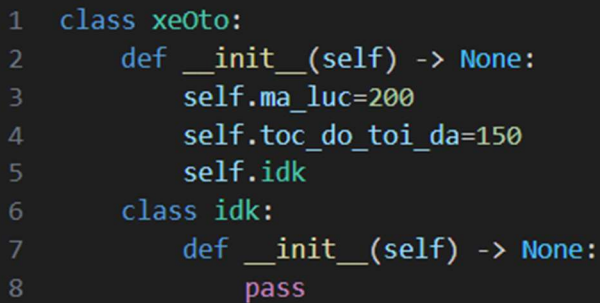
-`ma_luc` : Thông số này được gán dữ liệu là 200

-`toc_do_toi_da` : Thông số này được gán dữ liệu là 150

Thông số là biến của class đó.

Bạn cũng có thể tạo ra một class khác hoặc hàm, v.v.

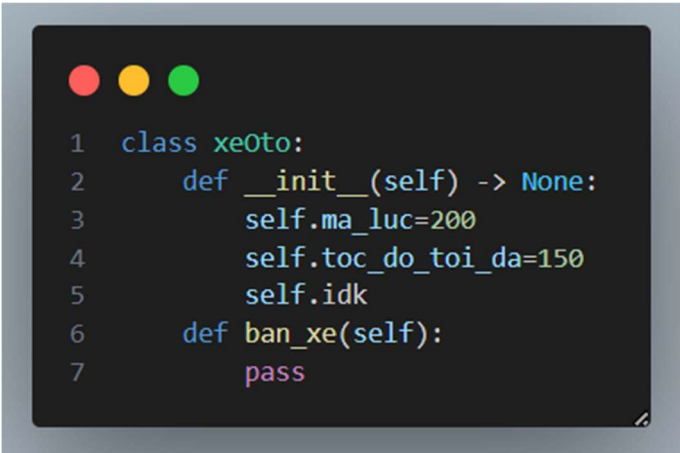
Nếu như bạn muốn tạo một class trong một class thì bạn chỉ cần làm như sau :



```
1 class xeOto:
2     def __init__(self) -> None:
3         self.ma_luc=200
4         self.toc_do_toi_da=150
5         self.idk
6     class idk:
7         def __init__(self) -> None:
8             pass
```

Vậy thì class idk sẽ thuộc class xeOto.

Nhưng mà đối với hàm thì nó sẽ khác, cách tạo vẫn thế nhưng mà bạn phải thêm self vào parameter đầu, VD :



```
1 class xeOto:
2     def __init__(self) -> None:
3         self.ma_luc=200
4         self.toc_do_toi_da=150
5         self.idk
6     def ban_xe(self):
7         pass
```

Khi bạn tạo class trong class thì cái class con đó sẽ không thể sử dụng các thông số của class lớn nhưng class lớn có thể sử dụng của class con.

Nhưng đối với hàm thì hàm có thể sử dụng các thông số của class chủ.