

Python OOP 1

Concatenating

- عملية ال concat يعني بدمج حاجتين مع بعض واللي دايمًا ممكن تستخدم لما ايجي اجمع بين كلمتين مع بعض في مجال البرمجة عموماً مش البايثون بس تعالي نشوف مثال

```
>>> a = " hello "  
>>> b = " world "  
>>> print(a+b)  
hello world
```

- خلي بالك هنا المتغيرين a و b هما string فا قالك تمام وطلعك الناتج تمام
- افرض لو ال b مثلاً كانت 1 طبيعي هيطلعك ايررور فا انت عشان تخليه يحللك المشكله دي حطو str(1)
- كدا انت بتقول للكومبيلر دي string مش int او ممكن تريخ دماغك وتحطها بين ""
- المهم اني لازم لو هتعمل concat يكون المتغيرين string

```
>>> a = 3  
>>> b = 10  
>>> print(a+b)  
13
```

- دي عملية جمع عادية جداً



Array

- الarray عبارة عن متغير فيه شوية قيم

```
wordlist = ["admin", "123", "password123", "administrator", "qwerty"]  
  
for word in wordlist:  
    print(word)
```

- بص هنا ال wordlist دي عبارته عن array ودي القيم اللي جواها
- اللي تحتها دي for loop بعملها علي ال array عشان لو عاوز اطبعها او اعمل عملية علي كل عنصر وكدا
- هنا بقولو `for word in wordlist` يعني بقولو المتغير word هيعدي علي كل value جوا ال wordlist وبعد كدا `print(word)` يعني اطبعلي القيمة دي وبكدا هيطبعلي كل قيم ال array

```
admin
123
password123
administrator
qwerty
```

```
wordlist = ["admin", "123", "password123", "administrator", "qwerty"]

for word in wordlist:
    if "123" in word:
        print(word)
```

- ركز معايا هنا انا هنا عملت array عاديه خالص وبعد كدا عملت for loop بس مش هطبع زي الاول عادي كدا
- انا هنا بقولو `if "123" in word` بص ياباشا احنا اتفقنا اني ال word ده المتغير اللي هيعدي علي كل قيمه ف ال array فا انا هنا بقولو ال 123 دي في ال word اطبعها
- يعني معني كلامك اني اي كلمه فيها 123 هيطبعها؟ اقدر اقولك اه بالضبط كدا

```
123
password123
```



Function With Return

- احنا اتعلمنا نعمل function قبل كدا بس متعلمناش نعملها ب return
- الفانكشن اللي بتكون ب return مهمه عشان القيمة اللي هتطلعلي من الفانكشن هستخدمها ف حاجه ثانيه فا هيا مهمه جدا تعالى بقي اقولك بتتعمل ازاي

```
def sum():
    x = 3
    y = 4
```

```
return x + y
```

```
new = sum()  
print(new)
```

- بص هنا انا عملت فانكشن عاديہ جدا اسمها sum واديتها متغير اسمه x ومتغير اسمه y واديتلهم قيم
- وبعد كذا قولت للفانكشن `return x + y` يعني بقولو يا باشا اجمعلي ال x وال y والناتج بتاعهم اعلمي ليہ return خزنهولي فالفانكشن
- بعد كذا قولت `new = sum()-ds` يعني بقولو الفانكشن دي حطهالي ف متغير اسمه new يعني معني كذا اني new بقي قيمتها دلوقتي كام؟ 7؟ صح كذا
- بعد كذا اطبعلي ال new دي
- خلي بالك انا كان ممكن اقولو اطبعلي الفانكشن علاطول وهيطبعها بس حبيت اعمل كذا عشان اقولك اني القيمه الراجعہ من الفانكشن ممكن احطها ف اي متغير عادي

```
def sum(x):  
    y = 4  
    return x + y  
  
new = sum(100)  
print(new)
```

- دا نفس الكلام بس فانكشن ب paramater لو فاكهه لو مش فاكهه ارجع شوف شرحه ف المحاضره اللي قبل دي

```
def sum(x,z):  
    y = 4  
    return x + y + z  
  
new = sum(100,200)  
print(new)
```

```
def sum(x,z):  
    word = f"the sum of {x} and {z} is {x+z}"  
    return word
```

```
new = sum(100,200)  
print(new)
```

```
root@team:~/hossam# python3 file2.py  
the sum of 100 and 200 is 300  
root@team:~/hossam#
```

٥٥

Class

- بص يا باشا دلوقتي احنا اتعلمنا حاجات كتير فالبايثون الحمد لله عاوزين بقي نخش فال OOP
- ال OOP دا اختصار ل Object Oriented Programming ودا نظام كتابه عشان انسق الكود بتاعي بيشكل حلو ورايق واي حد هيقراه ورايا يكون سهل عليه ومنتظم
- ال OOP عبارته عن شوية classes فيها شوية methods بتعمل حاجات معينه وببدأ استدعي ال classes عن طريق اني باخد منها object وببدأ استدعي الفانكشنز اللي جواها
- انا هنا بقسم البرنامج بتاعي ل classes يعني مثلاً لو بعمل سيستم جامعته فا بقسم الكلاسات مثلاً كلاس للاداره وكلاس للدكاتره وكلاس للمعيدين وكلاس للطلاب وجوا كل كلاس فيها بقي methods او functions خاصه لكل قسم
- يلا نخش عالكوود علاطول يا باشا
- عشان اعرف اعمل class سهله جدا بيبقي استخدم كلمة class سهله اهي `class` :
- يعني دلوقتي لما اقول `class newClass` يعني بقولو اعمل كلاس اسمه newClass
- بس كذا جوا ال class بقي بعمل functions بس هنسميها هنا methods بقي
- هنعمل فانكشن عاديته جدا زي ما اتعلمنا بس خلي بالك لازم اي فانكشن هعملها ف الكلاس لازم احطها parameter اسمه self بس كذا

```
class NewClass:
    def sum(self, x, y , z):
        print(x,y,z)
```

- هنا self ده ملوش اي لازمه بس لازم اخطو وبعد كذا اخط ال Params بتوعي عادي جدا
- طيب انا دلوقتي عملت ال class ازاي استخدمه؟ لازم عشان اعرف استخدم ال class لازم اخد منه object او نسخه يعني

```
class NewClass:
    def sum(self):
        print("hello world")
```

```
object1 = NewClass()
```

```
print(object1.sum())
```

- بص هنا طريقة اخد object بقولو يا باشا `obj1 = NewClass()` طبعا ممكن اغير obj دي اخليه اي اسم انا عاوزه
- عاوز اقولك حاجه اول لما باخد object كذا وباجي اعمل run اول حاجه بتتنفذ حاجه اسمها ال constructor ودا هنعرف شرحه كمان شويه بس عاوزك يكون عند المعلومه دي بس
- خلي بالك الكود لحد هنا لو عملتلو run مفيش اي فانكشن جواه هتشتغل

```
class NewClass:
    def sum(self):
        print("hello world")
```

```
object1 = NewClass()
```

- طب انا عاوز بقي اشغل الفانكشنز اللي جواه اقولك بقي استدعيها بال object اللي انت عملتو اللي هوا هنا اسمه object1

```
print(object1.sum())
```

```
root@team:~/hossam# python3 test2.py
hello world
```

```
class NewClass:
    def sum(self, x , y , z):
        print(x+y+z)

object1 = NewClass()

adding = object1.sum(1,3,10)

print(adding)
```

- بص دي كذا وحاول تفهمها
- ودا الناتج اللي هيطلع

```
root@team:~/hossam# python3 test2.py
14
```

- طيب دلوقتي تمام بص كذا عاوز اقولك حاجه

```
class NewClass:
    def newfun(self):
        x = 4
        self.x = x

    def another(self):
        print(self.x)

object1 = NewClass()
object1.newfun() # Call newfun to initialize self.x
object1.another() # Now call another to print self.x
```

- بص الصوره دي
- انا هنا عاوز استخدم المتغير x برا الفانكشن بتاعته فالطبيعي مينفعش وهيچبك error
- طب ولو عاوز استخدمه برا الفانكشن بتاعته؟ يبقي لازم اعمل الحته دي `self.x = x` كذا اقدر استخدمه برا الفانكشن بتاعته عادي

- جيت بقي فالفانكشن بتاعت another قولتله اطبعلي self.x اللي هو المتغير x يعني بس لو قولتله x بس هيطلعلي error فا لازم اكتب self.x يارب تكون النقطه دي وضحت

```
root@team:~/hossam# python3 file.py
```

4

س

Constructor

- فاكّر بقي ال constructor اللي قولتلك عليه فالبدايه طبعاً انت دلوقتي بتفكر عامل تقول ايه ال constructor ده
- بص يسطاً ال constr دا عبارّه عن فانكشن عاديّه جدا بس هي بتننّفذ اول لما تاخد object من الكلاس
- بص كدا

```
class NewClass:
    def __init__(self):
        print("this is first step")

object1 = NewClass()
```

- هيا بيكون ليها تعريف مختلف شويه كدا

```
:def __init__(self)
```

- طب ولو مكتبتهاش؟ يبقي خلاص البرنامج هيتنّفذ عادي من غير مايحصل حاجه هيا اصلاً بتكون موجوده default بس لو عاوز تعدل عليها لازم تكتبها وخلي بالك تقدر تضيف ليها params عادي جدا وتعمل اللي انت عاوزه هيا عبارّه عن فانكشن عاديّه
- وبص اول لما اخدت object علاطول لما اعمل run هيطبعلي this is first step

```
root@team:~/hossam# python3 file2.py
this is first step
```

```
class NewClass:
    def __init__(self):
        self.funtwo()

    def funtwo(self):
        print("this is function two not the constructor")

object1 = NewClass()
```

- تقدر تقولي فهمت ايه من الصوره دي؟ هسيبك تفكر ومش هقولك الناتج

```
root@team:~/hossam# cat file2.py
class NewClass:
    def __init__(self, x):

        print(x)

object1 = NewClass(1234)
```

- طب هنا انا حطيت param فالكونستراكتور لما اجي احطله قيمه بقي اللي هوا ال argument لو فالفانكشن العاديه كان بنحطها واحنا بنستدعي الفانكشن
- انما هنا هنحطلو ال argument ف استدعاه ال class بين قوسين ال class

```
root@team:~/hossam# python3 file2.py
1234
```

```
class NewClass:
    def __init__(self, x=0):

        print(x)

object1 = NewClass(12345)
```

- انا هنا مأجربتتش اليوزر انو يحط argument يعني علي فكره لو متحطتش ال 12345 دي عادي البرنامج هيرن من غير ابررورز علي فكره لأنني مديله قيمه اصلا فوق بصفر

