

في الدرس دا هنتكلم عن ال **inheritance** وبالعربي يعني الوراثة، ودا مفهومه ف ال OOP هو اني بورث class ثاني في الخصائص بتاعته، يعني بعمل class يكون في خصائص ال class الاول وممكن ازود عليها.

طيب ايه استخداماته؟ يعني ليه اخلي class ياخذ خواص class ثاني ؟ هناخد مثلا الجروبات بتاع السوشال ميديا كمثال، الجروب بيكون فيه اعضاء (members) وبيكون فيه admins، ال member بيكون ليه بعض الصلاحيات وال methods اللي يقدر ينفذها في الجروب، وال admin بيكون عنده نفس الصلاحيات اللي موجوده لل member وبيزيد عليه بعض الصلاحيات الزيادة يعني معاه ال methods بتاع ال member وزيادة عليها كمان شوية methods، فهنا بجي مبدأ ال inheritance اللي هو بدل م اعمل class لل member لوحيدده فيه بعض ال methods وارجع اعمل class ثاني لل admin فيه نفس ال methods وازود عليه، لا بخلي ال class بتاع ال admin يورث ال methods اللي موجوده في ال class ال member من غير م اعيد كتابتها وازود بس ال methods اللي مسموحه لل admins ومش لل members. دا كان مجرد مثال يوضح مبدأ ال inheritance.

طيب ازاي بحقق ال inheritance ؟ في بعض اللغات البرمجييه بكتب اسم ال class اللي عايز يورث وبعدها extends وبعدها اسم ال class اللي هيورث منه، يعني كدا **MyClass2 extends MyClass1{}** كدا خليت MyClass2 يورث MyClass1.

في البايثون بخلي class يورث واحد ثاني عن طريق اني بكتب اسم ال class وافتح قوسين () واكتب جواهم اسم ال class اللي هيورث منه، يعني كدا **MyClass2(MyClass1):**

طيب نيحي بقا للشغل جوا ال class اللي وارث، اول حاجه وراثه ال **methods** ودي بتتم تلقائي بمجرد م اقول ان ال class اللي اسمه B بيورث A، كدا بقا عندي B يقدر يوصل لأي method في A ويستخدمها وينفذها عادي.

وراثه ال attributes بقا بتتم بطريقتين، الأولى هي اني اكتب جوا ال constructor بتاع الوارث اسم اللي هورث منه وبعد كدا **__init__()** وجوا الاقواس هكتب اسم ال attributes اللي عايز اورثها منه بشرط اول parameter يكون **self** يعني هيكون كدا **Myclass2.__init__(self, param1, param2)**، وطبعاً لو فيه attribute مش عايز اورثه ميبكتبوش من ضمن ال parameters، بس لازم لاحظ اني احط اسامي ال attributes برضو ك parameters جوا اقواس ال constructor.

اما الطريقه الثانيه فبدل كتابة اسم اللي هورث منه في الاول بكتب بداله method اسمها **super()** واكتب بعدها **__init__()** ومش شرط اكتب **self** ك parameter جوا الاقواس، يعني بتكون كدا **super().__init__(param1, param2)**.

طبعاً بقدر في ال class الجديد اعرف فيه attributes ثاني و methods ثانيه واستخدمهم وفنفس الوقت استخدم بتوع اللي وارث منه.

ملحوظه مهمه: لو انا عامل attribute في ال class الأساسي ومديله قيمه وبعد كدا جيت ورثت ال attribute ب class ثاني بس اديتله قيمه مختلفه في الجديد هيعمل **override** على القيمه بتاع اللي جوالي من ال class الاساسي وهتكون دي قيمته الحاليه بس اكيد مش هياثر على قيمة الاساسي حاجه.

```

# -----
# -- Object Oriented Programming => Inheritance --
# -----

class Food: # Base Class

    def __init__(self, name, price):

        self.name = name

        self.price = price

        print(f"{self.name} Is Created From Base Class")

    def eat(self):

        print("Eat Method From Base Class")

class Apple(Food): # Derived Class

    def __init__(self, name, price, amount):

        # Food.__init__(self, name) # Create Instance From Base Class

        super().__init__(name, price)

        self.amount = amount

        print(f"{self.name} Is Created From Derived Class And Price Is {self.price} A
nd Amount Is {self.amount}")

    def get_from_tree(self):

        print("Get From Tree From Derived Class")

# food_one = Food("Pizza")
food_two = Apple("Pizza", 150, 500)
food_two.eat()
food_two.get_from_tree()

```

CODE

```

Pizza Is Created From Base Class
Pizza Is Created From Derived Class And Price Is 150 And Amount Is 500
Eat Method From Base Class
Get From Tree From Derived Class

```

OUTPUT