

في الدرس دا هنتكلم عن ال **Encapsulation**، وهو عبارته عن اني بتحكم في ال access على ال Attributes وال Methods بتاعتي.

يعني ايه بتحكم في ال access؟ يعني بحدد نقدر نعمل عليهم تعديل منين في البرنامج. عندنا 3 انواع، اول نوع هو ال **public** ودا اللي بنستخدمه من اول م بدأنا دروس ال OOP، ودا ال attributes العاديه وال Methods العاديه اللي بقدر اتحكم فيهم من جوا ومن برا ال class، يعني اقدر ادي متغير قيمه معينه من برا ال class.

تاني نوع وهو ال **protected** ودا عبارته عن اني ببلغ ال developer اللي هيشغل على الكود معايا ان المتغير دا **protected** يعني متستخدموش الا في ال class بتاعه او ال sub classes (ال inherited classes يعني)، بس البرنامج مبيمنعوش انه يستخدمه برا ال class لا دا ينفع يستخدمه عادي بس دي ما هي الا وسيلة تبليغ بس. بحدد ان المتغير دا **protected** عن طريق اني بحط قبل اسمه **underscore** يعني لو عندي متغير اسمه **name** فهو كدا **public** انما لو عملته كدا **_name** فهو كدا **protected**.

ثالث حاجه وهو ال **Private** ودا اللي بيمنع استخدام اي **method** او **attribute** الا من جوا ال class ولو حد حاول يستخدمها من برا ال class هيديله **error**. بخلي المتغير يكون **private** عن طريق اني بحط قبله اتنين **underscore** يعني كدا **__name**.

طيب يعني ايه مقدرش استخدم المتغير الا من جوا ال class ؟ يعني بعمل المتغير علشان استخدمه في الفانكشنز اللي جوا ال class بس.

لكن للأسف برضو ال **private** مبيمنعش الوصول فعلا للمتغير لكن فيه طريقه للوصول ليه والتعديل عليه كمان من برا ال class عن طريق اني بكتب قبله اسم ال class زي اخر سطر في الكود.

ملحوظه مهمه: للأسف البايثون مبتعملش **restrict** (تمنع وصول) فعلي للمتغير فنقدر نقول ان هي بتكون مجرد طريقه لأعلام ال developer اللي شغال ع الكود معايا ان دا **protected** او **private**. لكن في بعض اللغات الأخرى بيختلف ال **syntax** ويكون فيه **keywords** اسمها **protected** و **private** ومش مجرد اعلام للي شغال معاك، لا دي بتعمل **restrict** فعلي ويتمنع الوصول للمتغيرات دي.

```
# -----
# -- Object Oriented Programming => Encapsulation --
# -----
# Encapsulation
# - Restrict Access To The Data Stored in Attributes and Methods
# Public
# - Every Attribute and Method That We Used So Far Is Public
# - Attributes and Methods Can Be Modified and Run From Everywhere
# - Inside Our Outside The Class
# Protected
# - Attributes and Methods Can Be Accessed From Within The Class And Sub Classes
# - Attributes and Methods Prefixed With One Underscore _
# Private
# - Attributes and Methods Can Be Accessed From Within The Class Or Object Only
# - Attributes Cannot Be Modified From Outside The Class
# - Attributes and Methods Prefixed With Two Underscores __
# -----
# - Attributes = Variables = Properties
# -----
```

```

class Member:

    def __init__(self, name):

        self.name = name # Public

one = Member("Ahmed")
print(one.name)
one.name = "Sayed"
print(one.name)

class Member:

    def __init__(self, name):

        self._name = name # Protected

one = Member("Ahmed")
print(one._name)
one._name = "Sayed"
print(one._name)

class Member:

    def __init__(self, name):

        self.__name = name # Private

    def say_hello(self):

        return f"Hello {self.__name}"

one = Member("Ahmed")
# print(one.__name)
print(one.say_hello())
print(one._Member__name)

```

CODE

```

Ahmed
Sayed
Ahmed
Sayed
Hello Ahmed
Ahmed

```

OUTPUT