

الدرس دا هيتكلم عن ال **Magic Methods** ودي بتكون ال methods اللي اللي بتكون متحاوطه بين اثنين double underscore اللي هي زي ال `__init__` .

النوع دا من ال Methods مبيتمش استدعاءه، دا بيعمل وظيفه ثابتة في ال class بتاعي بناءا على حاجه ثانيه بنفذها ف ال class، وبيكون اسمه معروف وثابت يعني مش اسامي بسميها من نفسي.

طيب يعني ايه بيعمل وظيفه ثابتة ؟ يعني زي ال `__init__` كدا مش بستدعيها هي بتشتغل تلقائي لما بعرف object من ال class.

هناخد كام نوع كدا من ال magic methods ونعرف بيعملوا ايه:

`__str__`

دي بترجعلي خرج اقدر اقراه (انا بحدده) لما احاول اطبع object، يعني على سبيل المثال لو انا عامل object اسمه person1 من class وقلته `print(person1)`، اللي هيطلعلي كخرج هيقلي ان دا object من class كذا ويديني ال location بتاعه في ال memory، بس باستخدام ال `__str__` اقدر اطبعه كلام معين على حسب رغبتني يظهر لما حد يحاول يطبع ال object.

`__len__`

لو كنت معرف object اسمه profile من class وجيت استخدمت عليه فانكشن `len()` هيديني error لانه مش عارف هيرجعلي عدد عناصر لأيه، لكن باستخدام `__len__` بقدر اقله جواها ان لما يتم استخدام فانكشن `len()` على ال object بتاعي رجعلي عدد العناصر اللي موجوده فمتغير كذا.

يعني على سبيل المثال في الكودانا عرفت attribute اسمه skills عباره عن list وجواها 3 عناصر، وجوا ال `__len__` قلته يرجعلي عدد العناصر اللي جوا ال attribute دا، فبالتالي لما استخدم فانكشن `len` على ال object بتاعي هيرجعلي عدد العناصر اللي جوا ال list.

`__class__`

اتكلمنا عنها ف الدروس اللي فاتت وقلنا اني بستخدمها علشان اعرف object معين معمول من انهي class، ودي مش لازم اعرفها تعمل ايه جوا ال class هي بتعمل زي وظيفه static كدا. بستخدمها بأني بكتب اسم ال object اللي عايز اعرف من انهي class وبكتبها جنبه، زي كدا `.print(myObject.__class__)`.

ملحوظه: انواع البيانات اللي هي `str` و `int` و `float` دي عباره عن classes وبيكون جواها methods يعني انا لما بعرف متغير جديد ك `string` مثلا فكد انا بعمل object من class اللي هو `str`. وممكن نعمل `print(dir(str))` وهيرجعلي اللي جواه.

```
# -----
# -- Object Oriented Programming => Magic Methods --
# -----
# Everything in Python is An Object
# __init__ Called Automatically When Instantiating Class
# self.__class__ The class to which a class instance belongs
# __str__ Gives a Human-Readable Output of the Object
# __len__ Returns the Length of the Container
# Called When We Use the Built-in len() Function on the Object
# -----
```

```

class Skill:

    def __init__(self):

        self.skills = ["Html", "Css", "Js"]

    def __str__(self):

        return f"This is My Skills => {self.skills}"

    def __len__(self):

        return len(self.skills)

profile = Skill()
print(profile)
print(len(profile))

profile.skills.append("PHP")
profile.skills.append("MySQL")

print(len(profile))

print(profile.__class__)
my_string = "Muhammad"
print(type(my_string))
print(my_string.__class__)
print(dir(str))
print(str.upper(my_string))

```

CODE

```

This is My Skills => ['Html', 'Css', 'Js']
3
5
<class '__main__.Skill'>
<class 'str'>
<class 'str'>
=====
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewar
gs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce_
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'coun
t', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'isl
ower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffi
x', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', '
upper', 'zfill']
=====
MUHAMMAD

```

OUTPUT