**MAXSUS METODLAR**

Pythonda obyektlar bilan ishlashni yanada qulay qilish uchun bir nechta maxsus metodlar bor. Bu metodlarning nomi ikki pastki chiziq bilan yozilgani uchun, **d**ouble **under**score yoki qisqa qilib dunder metodlar deb ataladi. Dunder metolar yordamida obyektlarga qo'shimcha qulayliklar va vazifalar qo'shishimiz mumkin. Klass yoki obyektga oid dunder metodlar ro'yxatini ko'rish uchun dir() funksiyasidan foydalanamiz:

Copy

>>> dir(Avto)

['\_Avto\_\_num\_avto',

'\_\_class\_\_',

'\_\_delattr\_\_',

'\_\_dict\_\_',

'\_\_dir\_\_',

'\_\_doc\_\_',

'\_\_eq\_\_',

'\_\_format\_\_',

'\_\_ge\_\_',

'\_\_getattribute\_\_',

'\_\_gt\_\_',

'\_\_hash\_\_',

'\_\_init\_\_',

'\_\_init\_subclass\_\_',

'\_\_le\_\_',

'\_\_lt\_\_',

'\_\_module\_\_',

'\_\_ne\_\_',

'\_\_new\_\_',

'\_\_reduce\_\_',

'\_\_reduce\_ex\_\_',

'\_\_repr\_\_',

'\_\_setattr\_\_',

'\_\_sizeof\_\_',

'\_\_str\_\_',

'\_\_subclasshook\_\_',

'\_\_weakref\_\_',

'make',

'model',

'narh',

'rang',

'yil']

Dunder metodlardan biz \_\_init\_\_ metodi bilan tanishdik. Bu metod klassdan obyekt yaratishda chaqiriladi va obyektning xususiyatlarini belgilaydi. Ushbu darsimizda esa maxsus metodlarning ba'zilari bilan tanishamiz.

**OBYEKT HAQIDA MA'LUMOT**

Obyektga print() yoki str() orqali murojat qilinganda obyekt haqida tushunarli ma'lumot qaytarish uchun \_\_repr\_\_va \_\_str\_\_ metodlaridan foydalanamiz. Tushunarli bo'lishi uchun avvalgi darsimizdagi Avto klassiga qaytamiz:

Copy

class Avto:

\_\_num\_avto = 0

"""Avtomobil klassi"""

def \_\_init\_\_(self,make,model,rang,yil,narh):

"""Avtomobilning xususiyatlari"""

self.make = make

self.model = model

self.rang = rang

self.yil = yil

self.narh = narh

Avto.\_\_num\_avto += 1

Yuqoridagi klassdan yangi obyekt yaratamiz va obyekt haqida ma'lumot olish uchun print() funksiyasini chaqiramiz:

Copy

avto1 = Avto("GM","Malibu","Qora",2020,40000)

print(avto1) # obyekt haqida ma'lumot

Natija: <\_\_main\_\_.Avto object at 0x00000238A6DAE0C8>

Qandaydur tushunarsiz ma'lumot. Ekrandagi natijadan biz faqat avto1 obyektimiz Avto klassiga tegishli ekanini ko'ramiz. Qanday qilib shuning o'rniga obyekt haqida tushunarliroq ma'lumot olishimiz mumkin?

Gap shundaki biz har gal obyketga print() (yoki str() yoki repr()) orqali murojat qilganimizda, Python obyket ichida \_\_str\_\_ yoki \_\_repr\_\_ metodlariga murojat qiladi. Agar biz bu metodlarni yozmagan bo'lsak, yuqoridagi kabi umumiy ma'lumot qayataradi.

Biz ushbu metodlarni yangidan yozib, biz istagan ma'lumotni qayataradian qilishimiz mumkin. Odatda, yuqoridagi ikki metoddan birini yozish kifoya. Odatda, \_\_repr\_\_ umumiyorq, \_\_str\_\_ esa batafsilroq ma'lumot olish uchun ishlatiladi.

Ikkalasidan birini tanlaganda, \_\_repr\_\_metodiga yon bosiladi, sababi bu metod print(), str() va repr() funksiyalarining hammasi bilan ishlaydi. Keling biz ham yuoqirdagi klassimizga\_\_repr\_\_metodini qo'shamiz:

Copy

class Avto:

\_\_num\_avto = 0

"""Avtomobil klassi"""

def \_\_init\_\_(self,make,model,rang,yil,narh):

"""Avtomobilning xususiyatlari"""

self.make = make

self.model = model

self.rang = rang

self.yil = yil

self.narh = narh

Avto.\_\_num\_avto += 1

def \_\_repr\_\_(self):

"""Obyekt haqida ma'lumot"""

return f"Avto: {self.rang} {self.make} {self.model}"

Qaytadan print() funksiyasini chaqiramiz:

Copy

avto1 = Avto("GM","Malibu","Qora",2020,40000)

print(avto1)

Natija: Avto: Qora GM Malibu

Mana endi natijamiz ancha tushunarli ko'rinishda chiqdi.