

class method vs static method vs instance method

class method

کلاس متدها به جای یک آبجکت با خود کلاس سروکار دارند و همچنین با دکوریاتور `@classmethod` تعریف می شوند. اینگونه متدها زمانی کارآمد هستند که نیاز داشته باشیم تا با یک متغیر یا یک عملکردی که مربوط به کل کلاس است و از آبجکت سوا است کار داشته باشیم.

کلاس متدها نیز مانند آبجکت متدها (متدهای معمولی) یک ورودی خاص دارند که به طور معمول `cls` نامیده می شود. (مشابه `self` در آبجکت متدها) و برخلاف `self` که به آبجکت اشاره می کرد، `cls` به خود کلاس اشاره می کند و این متدها بر روی خود کلاس صدا زده می شوند.

```
1 class Foo:
2     sum = 0
3
4     def __init__(self, num):
5         self.num = num
6
7     @classmethod
8     def add(cls, num):
9         cls.sum += num
10        return cls.sum
11
12
13 print(Foo.sum)
14 print(Foo.add(5))
15 print(Foo.add(10))
```

[Copy](#) [Python](#)

خروجی چیه؟ ▼

0
5
15

▼ نکته

کلاس متدها قابلیت این رو دارند که روی آبجکت صدا زده باشن و ارور داده نمیشه اما عملکرد اصلی خودشون رو نشون نمیدن.

static method

استاتیک متدها نه با آبجکت و نه با خود کلاس کاری ندارند و در اصل تابع هایی هستند که داخل namespace کلاس تعریف شده اند و با دکوریاتور @staticmethod تعریف می شوند و همچنین ورودی ای مشابه self و یا cls ندارند پس در نتیجه هیچ وابستگی ای به کلاس یا آبجکت ندارند.

```

1 class Foo:
2     def __init__(self, num):
3         self.num = num
4
5     @staticmethod
6     def add(a, b):
7         return a + b
8
9 print(Foo.add(5, 15))
10 print(Foo.add(10, 13))

```

▼ خروجی چیه؟

20

23

یک مثال بهتر

```

1
2 from datetime import date
3
4
5 class Person:

```

```

6     def __init__(self, name, age):
7         self.name = name
8         self.age = age
9
10    @classmethod
11    def from_birth_year(cls, name, year):
12        return cls(name, date.today().year - year)
13
14    @staticmethod
15    def is_adult(age):
16        return age >= 18
17
18
19    person1 = Person('Sam', 17)
20    person2 = Person.from_birth_year('Reza', 1996)
21
22    print(person1.age)
23    print(person2.age)
24    print(Person.is_adult(20))

```

▼ خروجی چیه؟

```

17
29
True

```

مقایسه

۱. تعریف

- کلاس متد: @classmethod
- استاتیک متد: @staticmethod
- آبجکت متد: ندارد

۲. دسترسی

- کلاس متد: دسترسی به متغیرهای کلاس و تغییر آن ها

- استاتیک متد: نمی تواند به متغیرهای کلاس یا آبجکت دسترسی پیدا کرده و آن ها را تغییر دهد.
- آبجکت متد: دسترسی به متغیرهای آبجکت و تغییر آن ها

۳. ورودی اول

- کلاس متد: `cls` که نشان دهنده کلاس است
- استاتیک متد: ندارد.
- آبجکت متد: `self` که نشان دهنده آبجکت است.

۴. استفاده

- کلاس متد: اغلب در جاهایی که نیاز است تا متغیرهای کلاس را تغییر دهیم یا با آن ها تعامل داشته باشیم استفاده می شود.
- استاتیک متد: اغلب در جاهایی که کار ما از آبجکت و کلاس مستقل است.
- آبجکت متد: اغلب در جاهایی که عملیات وابسته به آبجکت است.

۵. ارث بری

- کلاس متد: قابلیت `override` شدن در کلاس فرزند را دارد و `cls` به کلاس فرزند اشاره می کند.
- استاتیک متد: قابلیت `override` شدن در کلاس فرزند را دارد اما `override` نمی شود.
- آبجکت کلاس: قابلیت `override` شدن در کلاس فرزند را دارد و `self` به آبجکت از جنس کلاس فرزند اشاره می کند.

۶. مثال

- کلاس متد: محاسبه مقدارهایی که بین کل آبجکت ها مشترک است مثل مجموعه سائزهای لاستیک
- استاتیک متد: انجام عملیات های ریاضی
- آبجکت کلاس: محاسبه مقدارهایی که مربوط به یک آبجکت است مثل سائز یک لاستیک