

Class-Decorators

`classmethod, staticmethod, property`

Klassen Dekorateure

Python stellt uns im Zusammenhang mit der Arbeit mit Klassen einige Dekoratoren zur Verfügung, die nützlich sind.

`@classmethod` gibt uns die Referenz auf die Klasse, macht aus der Methode also eine Klassenmethode und keine Objektmethode (Instanzmethode)

`@staticmethod` gibt uns weder Referenz auf die Klasse noch auf das Objekt.

`@property` lässt eine Methode von außen ansprechen wie eine Variable

@classmethod

Der @classmethod-Dekorator ermöglicht uns, in einer Klasse statt der Referenz auf das Objekt, d.h. die Klasseninstanz eine Referenz auf die Klasse selbst zu erhalten. Wird z.B. genutzt, um eine Instanz der Klasse zu erstellen (Fabrikmethode).

Class Person:

```
def __init__(self, name, age):  
    self.name = name  
    self.age = age
```

@classmethod

```
def from_dict(cls, person):  
    name, age, *_ = person.values()  
    return cls(name, age)
```

```
p = Person.from_dict({"name": "peter", "age": 3, "city": "Berlin"})
```

@staticmethod

Statische Methoden haben weder eine Referenz auf die Klasse noch auf die Instanz (Objekt). Sie werden hauptsächlich für sogenannte Utility-Funktionen eingesetzt, die zwar irgendwie zur Klasse gehören, aber nicht direkt eine Instanz betreffen.

@staticmethod

```
def centimeter_to_meter(centimeters: int) -> float:  
    return centimeters / 100
```

@property

Der property-Dekorator ist eine Möglichkeit, eine Methode von außen so anzusprechen, dass sie sich im Zugriff wie eine Variable verhält. Beim Setzen der Variable über `@energy.setter` können zum Beispiel Validierungen oder ähnliches vorgenommen werden. Der Zugriff von außen über den Dot-Operator ändert sich nicht.

Class Person:

```
@property
```

```
def energy():  
    return self._energy
```

```
@energy.setter
```

```
def energy(value):  
    if value < 0:  
        raise ValueError("negative energy is not possible")  
    self._energy = value
```

```
p = new Person()  
p.energy = 101  
print(p.energy)
```