Работа с методами экземпляра

In [15]:
```
class Human:

    def __init__(self, name, age=0):
        self.name = name
        self.age = age


class Planet:

    def __init__(self, name, population=None):
        self.name = name
        self.population = population or []

    def add_human(self, human):
        print(f"Welcome to {self.name}, {human.nam
e}!")
        self.population.append(human)
```

In [16]:
```
mars = Planet("Mars")

bob = Human("Bob")

mars.add_human(bob)
```

```
Welcome to Mars, Bob!
```

In [17]:
```
print(mars.population)
```

```
[<__main__.Human object at 0x10e416780>]
```

Вызов методов из методов

In [9]:
```
class Human:
    def __init__(self, name, age=0):
        self._name = name
        self._age = age

    def _say(self, text):
        print(text)

    def say_name(self):
        self._say(f"Hello, I am {self._name}")

    def say_how_old(self):
        self._say(f"I am {self._age} years old")
```

In [10]:
```
bob = Human("Bob", age=29)
```

In [11]:
```
bob.say_name()
bob.say_how_old()
```

```
Hello, I am Bob
I am 29 years old
```

In [14]:
```
# не рекомендуется!
print(bob._name)

# не рекомендуется!
bob._say("Whatever we want")
```

```
Bob
Whatever we want
```

Метод класса (@classmethod)

In [3]:
```
class Event:

    def __init__(self, description, event_date):
        self.description = description
        self.date = event_date

    def __str__(self):
        return f"Event \"{self.description}\" at {
self.date}"
```

In [5]:
```
from datetime import date

event_description = "Рассказать, что такое @classmet
hod"
event_date = date.today()

event = Event(event_description, event_date)
print(event)
```

```
Event "Рассказать, что такое @classmethod" at 20
17-07-09
```

In [ ]:
```
def extract_description(user_string):
    return "открытие чемпионата мира по футболу"


def extract_date(user_string):
    return date(2018, 6, 14)


class Event:

    def __init__(self, description, event_date):
        self.description = description
        self.date = event_date

    def __str__(self):
        return f"Event \"{self.description}\" at {
self.date}"

    @classmethod
    def from_string(cls, user_input):
        description = extract_description(user_inp
ut)
        date = extract_date(user_input)
        return cls(description, date)
```

In [8]:
```
event = Event.from_string("добавить в мой календарь
открытие чемпионата мира по футболу на 14 июня 2018
года")
print(event)
```

```
Event "открытие чемпионата мира по футболу" at 2
018-06-14
```

```
In [56]: dict.fromkeys("12345")

Out[56]: {'1': None, '2': None, '3': None, '4': None, '
         5': None}
```

В этом видео:

- Научились объявлять и вызывать методы экземпляров
- Посмотрели на метод класса (@classmethod)