lecture_7

August 29, 2022

1 Lecture 7 ~ OOP

1.1 Instance vs Class

Property vs Behaviour

[1]:
```
class Dog:
    def __init__(self, name):  # Consrtructor
        self.name = name
```

[2]:
```
bob = Dog(name="Bob")
jack = Dog("Jack")
```

[3]:
```
bob.name
```

[3]:
```
'Bob'
```

[4]:
```
jack.name
```

[4]:
```
'Jack'
```

[5]:
```
id(bob)
```

[5]:
```
4357117744
```

[6]:
```
id(jack)
```

[6]:
```
4357118032
```

[7]:
```
type(bob)
```

[7]:
```
__main__.Dog
```

[8]:
```
type(jack)
```

[8]:
```
__main__.Dog
```

[9]:

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

[11]:

```python
rich = Dog(name="Richie", age=7)
```

[12]:

```python
rich = Dog("Richie", 7)
```

[13]:

```python
rich = Dog(age=7, name="Richie")
```

[14]:

```python
class Dog:
    def __init__(self, name, age, color="black"):
        if not isinstance(name, str):
            raise Exception("name should be a string")
        self.name = name
        self.age = age
```

[15]:

```python
Dog(12, "jack")
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
Input In [15], in <cell line: 1>()
----> 1 Dog(12, "jack")

Input In [14], in Dog.__init__(self, name, age)
      2 def __init__(self, name, age):
      3     if not isinstance(name, str):
----> 4         raise Exception("name should be a string")
      5     self.name = name
      6     self.age = age

Exception: name should be a string
```

[21]:

```python
# BAD
bob = ["Bob", 12, "black"]
jack = ["Jack", 7, "yellow", "Henry"]
```

[22]:

```python
bob = {
    "name": "Bob",
    "age": 12,
    "color": "black"
}
```

[23]:

```python
dir(rich)
```

```
[23]: ['__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__',
       'age',
       'name']
```

```
[24]: vars(rich)
```

```
[24]: {'name': 'Richie', 'age': 7}
```

```
[25]: a = Dog("Jack", 12)
```

```
[26]: b = Dog("Jack", 12)
```

```
[27]: a == b
```

```
[27]: False
```

```
[28]: id(a)
```

```
[28]: 4400174896
```

```
[29]: id(b)
```

```
[29]: 4400176816
```

```
[30]: a is b
```

```
[30]: False
```

```
[31]: a == a
```

```
[31]: True
```

```
[32]: a is a
```

```
[32]: True
```

```
[33]: a.name == b.name
```

```
[33]: True
```

```
[35]: a.age == b.age
```

```
[35]: True
```

```
[36]: a == b
```

```
[36]: False
```

1.2 Instance Attribute vs Class Attribute

```
[43]: class Person:
          no_of_ears = 2  # class attribute

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

```
[44]: person_1 = Person("Jack", 42)
      person_2 = Person("Adam", 24)
```

```
[45]: person_1.name
```

```
[45]: 'Jack'
```

```
[46]: person_1.no_of_ears
```

```
[46]: 2
```

```
[47]: person_3 = Person("Vincent van Gogh", 42)
```

```
[48]: person_3.name

[48]: 'Vincent van Gogh'

[49]: person_3.no_of_ears

[49]: 2

[51]: person_3.no_of_ears -= 1

[52]: person_3.no_of_ears

[52]: 1

[53]: person_2.no_of_ears

[53]: 2

[54]: Person.no_of_ears

[54]: 2

[55]: Person.no_of_ears = 4

[56]: person_1.no_of_ears

[56]: 4

[57]: person_2.no_of_ears

[57]: 4

[58]: person_3.no_of_ears

[58]: 1

[59]: id(person_1.no_of_ears)

[59]: 4301482384

[60]: id(person_2.no_of_ears)

[60]: 4301482384

[61]: id(person_3.no_of_ears)

[61]: 4301482288
```

[62]:
```python
id(Person.no_of_ears)
```

[62]:
```
4301482384
```

[72]:
```python
class Employee:
    company_domain = "aca.am"

    def __init__(self, name, surname, external_domain=None):
        self.name = name
        self.surname = surname
        if external_domain is not None:
            self.company_domain = external_domain

    def get_company_domain(self):
        return f"{self.name}.{self.surname}@{self.company_domain}"
```

[73]:
```python
employee_1 = Employee("Henry", "Harutyunyan")
```

[74]:
```python
employee_1.get_company_domain()
```

[74]:
```
'Henry.Harutyunyan@aca.am'
```

[75]:
```python
employee_2 = Employee("Adam", "Smith")
```

[76]:
```python
employee_2.get_company_domain()
```

[76]:
```
'Adam.Smith@aca.am'
```

[77]:
```python
Employee.company_domain = "aca.com"
```

[78]:
```python
employee_1.get_company_domain()
```

[78]:
```
'Henry.Harutyunyan@aca.com'
```

[79]:
```python
employee_3 = Employee("Jack", "Beckett", external_domain="gmail.com")
```

[80]:
```python
employee_3.get_company_domain()
```

[80]:
```
'Jack.Beckett@gmail.com'
```

1.3 Methods

[109]:
```python
class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def speak(self):  # instance method
        return "woof woof"
```

[110]:
```
dog_1 = Dog("Bob", 12)
```

[111]:
```
dog_1.name
```

[111]:
```
'Bob'
```

[112]:
```
dog_1.age
```

[112]:
```
12
```

[113]:
```
dog_1.speak()   # Method
```

[113]:
```
'woof woof'
```

[114]:
```
print("hello")   # Function
```

```
hello
```

[115]:
```
class Employee:
    company_domain = "aca.am"

    def __init__(self, name, surname):
        self.name = name
        self.surname = surname
        self.email = f"{self.name}.{self.surname}@{self.company_domain}"
```

[116]:
```
employee_1 = Employee("Adam", "Smith")
```

[117]:
```
employee_1.email   # >>> adam.smith@aca.am
```

[117]:
```
'Adam.Smith@aca.am'
```

[118]:
```
Employee.company_domain = "google.com"
```

[119]:
```
employee_1.email
```

[119]:
```
'Adam.Smith@aca.am'
```

[120]:
```
employee_1.name = "Henry"
```

[121]:
```
employee_1.email
```

[121]:
```
'Adam.Smith@aca.am'
```

[125]:

```
class Employee:
    compnay_domain = "aca.am"

    def __init__(self, name, surname):
        self.name = name
        self.surname = surname

    @property
    def email(self):
        return f"{self.name.lower()}.{self.surname.lower()}@{self.
 ↪compnay_domain}"
```

[126]:

```
employee_4 = Employee("Samuel", "Buckett")
```

[127]:

```
employee_4.email
```

[127]:

```
'samuel.buckett@aca.am'
```

[128]:

```
Employee.compnay_domain = "google.com"
```

[129]:

```
employee_4.email
```

[129]:

```
'samuel.buckett@google.com'
```

[130]:

```
employee_4.name = "Samvel"
```

[131]:

```
employee_4.email
```

[131]:

```
'samvel.buckett@google.com'
```

[153]:

```
class Employee:
    compnay_domain = "aca.am"

    def __init__(self, name, surname, email=None):
        self.name = name
        self.surname = surname
        self.custom_email = email

    @property
    def email(self):
        if self.custom_email:
            return self.custom_email
        return f"{self.name.lower()}.{self.surname.lower()}@{self.
 ↪compnay_domain}"

    @email.setter
    def email(self, email):
        self.custom_email = email
```

[154]:
```
employee_4 = Employee("Samuel", "Buckett", "samuel1975@gmail.com")
```

[155]:
```
employee_4.email  # >>> samuel1975@gmail.com
```

[155]:
```
'samuel1975@gmail.com'
```

[156]:
```
employee_4.email = "samuel@yahoo.com"
```

[157]:
```
employee_4.email
```

[157]:
```
'samuel@yahoo.com'
```

[158]:
```
employee_5 = Employee("Adam", "Smith")
```

[159]:
```
employee_5.email
```

[159]:
```
'adam.smith@aca.am'
```

[160]:
```
employee_5.email = "adam@gmail.com"
```

[161]:
```
employee_5.email
```

[161]:
```
'adam@gmail.com'
```

[ ]:
```

```