In [6]:
```
print(dict)
```

```
<class 'dict'>
```

In [7]:
```
print(int)
```

```
<class 'int'>
```

In [9]:
```
print(int)
```

```
<class 'int'>
```

In [12]:
```
num = 13.0
print(type(num))
```

```
<class 'float'>
```

isinstance

In [15]:
```
num = 13
isinstance(num, int)
```

Out[15]:
```
True
```

In [16]:
```
numbers = {}
isinstance(numbers, dict)
```

Out[16]:
```
True
```

Объявление класса

In [2]:
```
class Human:
    pass
```

In [3]:
```
class Robot:
    """Данный класс позволяет создавать роботов"""
```

In [4]:
```
print(Robot)
```

```
<class '__main__.Robot'>
```

In [6]:
```
print(dir(Robot))
```

```
['__class__', '__delattr__', '__dict__', '__di
r__', '__doc__', '__eq__', '__format__', '__ge
__', '__getattribute__', '__gt__', '__hash__',
'__init__', '__init_subclass__', '__le__', '__
lt__', '__module__', '__ne__', '__new__', '__r
educe__', '__reduce_ex__', '__repr__', '__seta
ttr__', '__sizeof__', '__str__', '__subclassho
ok__', '__weakref__']
```

Создание экземпляра (объекта) класса

In [8]:
```
class Planet:
    pass
```

In [9]:
```
planet = Planet()
```

In [10]:
```
print(planet)
```

```
<__main__.Planet object at 0x10e8722b0>
```

In [11]:
```
solar_system = []
for i in range(8):
    planet = Planet()
    solar_system.append(planet)

print(solar_system)
```

```
[<__main__.Planet object at 0x10e872780>, <__m
ain__.Planet object at 0x10e8722b0>, <__main__
.Planet object at 0x10e8727f0>, <__main__.Plan
et object at 0x10e872828>, <__main__.Planet ob
ject at 0x10e872860>, <__main__.Planet object
at 0x10e872898>, <__main__.Planet object at 0x
10e8728d0>, <__main__.Planet object at 0x10e87
2908>]
```

In [14]:
```
solar_system = {}
for i in range(8):
    planet = Planet()
    solar_system[planet] = True

print(solar_system)
```

```
{<__main__.Planet object at 0x10e872978>: True
, <__main__.Planet object at 0x10e872908>: Tru
e, <__main__.Planet object at 0x10e8727f0>: Tr
ue, <__main__.Planet object at 0x10e872828>: T
rue, <__main__.Planet object at 0x10e872860>:
True, <__main__.Planet object at 0x10e872898>:
True, <__main__.Planet object at 0x10e8729e8>:
True, <__main__.Planet object at 0x10e872940>:
True}
```

Инициализация экземпляра

In [16]:
```
class Planet:

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

In [17]:
```
earth = Planet("Earth")
print(earth.name)
print(earth)
```

```
Earth
<__main__.Planet object at 0x10e8796d8>
```

In [10]:
```
class Planet:

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

    def __str__(self):
        return self.name


earth = Planet("Earth")
print(earth)
```

```
Earth
```

In [11]:
```
solar_system = []

planet_names = [
    "Mercury", "Venus", "Earth", "Mars",
    "Jupiter", "Saturn", "Uranus", "Neptune"
]

for name in planet_names:
    planet = Planet(name)
    solar_system.append(planet)

print(solar_system)
```

```
[<__main__.Planet object at 0x10477f160>, <__m
ain__.Planet object at 0x10477f278>, <__main__
.Planet object at 0x10477f198>, <__main__.Plan
et object at 0x10477f1d0>, <__main__.Planet ob
ject at 0x10477f208>, <__main__.Planet object
at 0x10477f240>, <__main__.Planet object at 0x
1048637b8>, <__main__.Planet object at 0x10486
37f0>]
```

In [2]:
```
class Planet:

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

    def __repr__(self):
        return f"Planet {self.name}"
```

In [3]:
```
solar_system = []

planet_names = [
    "Mercury", "Venus", "Earth", "Mars",
    "Jupiter", "Saturn", "Uranus", "Neptune"
]

for name in planet_names:
    planet = Planet(name)
    solar_system.append(planet)

print(solar_system)
```

```
[Planet Mercury, Planet Venus, Planet Earth, P
lanet Mars, Planet Jupiter, Planet Saturn, Pla
net Uranus, Planet Neptune]
```

Работа с атрибутами экземпляра

In [4]:
```
mars = Planet("Mars")
print(mars)
```

```
Planet Mars
```

In [5]:
```
mars.name
```

Out[5]:
```
'Mars'
```

In [6]:
```
mars.name = "Second Earth?"
mars.name
```

Out[6]:
```
'Second Earth?'
```

In [7]:
```
mars.mass
```

```
---------------------------------------------
-----------------------------
AttributeError
Traceback (most recent call last)
<ipython-input-7-3c1085af8f48> in <module>()
----> 1 mars.mass

AttributeError: 'Planet' object has no attribu
te 'mass'
```

In [8]:
```
del mars.name
```

In [9]:
```
mars.name
```

```
---------------------------------------------
-----------------------------
AttributeError
Traceback (most recent call last)
<ipython-input-9-202092835a22> in <module>()
----> 1 mars.name

AttributeError: 'Planet' object has no attribu
te 'name'
```

Мы с вами:

- Посмотрели как объявлять классы
- Научились создавать экземпляры (объекты) классов
- Рассмотрели как инициализировать экземпляр класса
- Научились работать с атрибутами экземпляра класса