

Введение в Data Science

Этап 1. Язык программирования Python

С. В. Сергеенко Д. В. Бирюкова

ООО «ЛАЦИТ — Лаборатория цифровых технологий»

2021

Магические функции — для чего нужны, описание наиболее используемых

Оператор определения класса

```
class X:  
2   a = 1  
    def __init__(self):  
        pass  
5   def k(self):  
        return self.a  
x = X()  
8 print(x.k(), X.a)  
x.a=2  
print(x.k(), X.a)
```

Специальные методы

Создание и удаление объектов

- ▶ `object.__new__(cls[, ...])` — создания нового экземпляра класса `cls`, применяется для изменения процесса создания экземпляров классов производных от иммутабельных типов. Следует возвращать экземпляр или подкласс класса `cls`.
- ▶ `object.__init__(self[, ...])` — инициализация экземпляра
- ▶ `object.__del__(self)` вызывается перед уничтожением объекта, может предотвратить, но, возможно, не будет вызван повторно.

Специальные методы

Сериализация

- ▶ `object.__repr__(self)`
- ▶ `object.__str__(self)`
- ▶ `object.__bytes__(self)`
- ▶ `object.__format__(self, format_spec)`

Специальные методы

Сравнение

- ▶ `object.__lt__(self, other)`
- ▶ `object.__le__(self, other)`
- ▶ `object.__eq__(self, other)`
- ▶ `object.__ne__(self, other)`
- ▶ `object.__gt__(self, other)`
- ▶ `object.__ge__(self, other)`

Специальные методы

Получение специальных значений

- ▶ `object.__hash__(self)`
- ▶ `object.__bool__(self)`

Специальные методы

Работа с атрибутами

- ▶ `object.__getattr__(self, name)`
- ▶ `object.__getattribute__(self, name)`
- ▶ `object.__setattr__(self, name, value)`
- ▶ `object.__delattr__(self, name)`
- ▶ `object.__dir__(self)`

Специальные методы

Поведение вызываемого объекта

► `object.__call__(self[, args ...])`

Специальные методы

Поведение контейнеров

- ▶ `object.__len__(self)`
- ▶ `object.__length_hint__(self)`
- ▶ `object.__getitem__(self, key)`
- ▶ `object.__setitem__(self, key, value)`
- ▶ `object.__delitem__(self, key)`
- ▶ `object.__missing__(self, key)`
- ▶ `object.__iter__(self)`
- ▶ `object.__reversed__(self)`
- ▶ `object.__contains__(self, item)`

Специальные методы

Поведение чисел

- ▶ `object.__add__(self, other)`
- ▶ `object.__sub__(self, other)`
- ▶ `object.__mul__(self, other)`
- ▶ `object.__matmul__(self, other)`
- ▶ `object.__truediv__(self, other)`
- ▶ `object.__floordiv__(self, other)`
- ▶ `object.__mod__(self, other)`
- ▶ `object.__divmod__(self, other)`
- ▶ `object.__pow__(self, other[, modulo])`
- ▶ `object.__lshift__(self, other)`

Специальные методы (прод.)

Поведение чисел

- ▶ `object.__rshift__(self, other)`
- ▶ `object.__and__(self, other)`
- ▶ `object.__xor__(self, other)`
- ▶ `object.__or__(self, other)`

Специальные методы

Поведение чисел: парные операции

- ▶ `object.__radd__(self, other)`
- ▶ `object.__rsub__(self, other)`
- ▶ `object.__rmul__(self, other)`
- ▶ `object.__rmatmul__(self, other)`
- ▶ `object.__rtruediv__(self, other)`
- ▶ `object.__rfloordiv__(self, other)`
- ▶ `object.__rmod__(self, other)`
- ▶ `object.__rdivmod__(self, other)`
- ▶ `object.__rpow__(self, other[, modulo])`
- ▶ `object.__rlshift__(self, other)`

Специальные методы (прод.)

Поведение чисел: парные операции

- ▶ `object.__rrshift__(self, other)`
- ▶ `object.__rand__(self, other)`
- ▶ `object.__rxor__(self, other)`
- ▶ `object.__ror__(self, other)`

Специальные методы

Поведение чисел: составное присваивание

- ▶ `object.__iadd__(self, other)`
- ▶ `object.__isub__(self, other)`
- ▶ `object.__imul__(self, other)`
- ▶ `object.__imatmul__(self, other)`
- ▶ `object.__itruediv__(self, other)`
- ▶ `object.__ifloordiv__(self, other)`
- ▶ `object.__imod__(self, other)`
- ▶ `object.__ipow__(self, other[, modulo])`
- ▶ `object.__ilshift__(self, other)`
- ▶ `object.__irshift__(self, other)`

Специальные методы (прод.)

Поведение чисел: составное присваивание

- ▶ `object.__iand__(self, other)`
- ▶ `object.__ixor__(self, other)`
- ▶ `object.__ior__(self, other)`

Специальные методы

Поведение чисел: унарные операции

- ▶ `object.__neg__(self)`
- ▶ `object.__pos__(self)`
- ▶ `object.__abs__(self)`
- ▶ `object.__invert__(self)`

Специальные методы

Поведение чисел: преобразование типа

- ▶ `object.__complex__(self)`
- ▶ `object.__int__(self)`
- ▶ `object.__float__(self)`
- ▶ `object.__index__(self)`

Специальные методы

Поведение чисел: округление

- ▶ `object.__round__(self[, ndigits])`
- ▶ `object.__trunc__(self)`
- ▶ `object.__floor__(self)`
- ▶ `object.__ceil__(self)`