

МАГИЧЕСКИЕ МЕТОДЫ КЛАССА

ВВЕДЕНИЕ

Задачи урока

- 01 Познакомиться с магическими методами
- 02 Научиться использовать магические методы
- 03 Понять, где можно применять те или иные магические методы

Магические методы класса

Магический метод `__init__` в Python является особым методом, который автоматически вызывается при создании нового объекта класса. Он используется для инициализации атрибутов объекта и задания начального состояния

Метод `__init__` принимает `self` и набор аргументов, которые могут быть переданы при создании объекта. Обычно в методе `__init__` инициализируются атрибуты объекта, которые будут доступны для использования в других методах этого класса

Пример использования метода `__init__`

```
...  
class Foo:  
    Age = None  
    Salary = 0  
  
    def __init__(self, name=None) -> None:  
        print("Мы в конструкторе")  
        self.name = name  
        self.Motivation = 0  
  
foo = Foo()  
foo.name = "Заполняем Имя"  
foo.Age = 30  
foo.Salary = 50_000
```

Магический метод `__call__` в Python используется для того, чтобы объекты класса можно было вызывать как функции. Когда объект вызывается с помощью круглых скобок, интерпретатор Python автоматически ищет и вызывает метод `__call__` этого объекта, если он определен

Пример использования метода `__call__`

```
...  
  
class Foo:  
    def __call__(self, a):  
        print("__call__", a)  
  
foo = Foo()  
foo(5) __call__ 5
```

Магический метод `__setattr__` в Python используется для определения поведения при присваивании значения атрибуту объекта. Этот метод вызывается автоматически при попытке присвоить значение атрибуту

Когда вы присваиваете значение атрибуту объекта, например, `obj.attr = value`, интерпретатор Python автоматически ищет и вызывает метод `__setattr__` для объекта `obj`

Пример использования метода `__setattr__`

```
...  
  
class Foo:  
    Age = None  
    Salary = 0  
  
    def __setattr__(self, name, value):  
        print(f"{name=}", f"{value=}")  
        if name == 'Salary':  
            self.__dict__['Motivation'] = value * 0.5  
  
        self.__dict__[name] = value
```

Магический метод `__enter__` в Python используется в контекстных менеджерах. Контекстные менеджеры предоставляют управление ресурсами, такими как файлы или соединения с базой данных, с помощью блока кода, ограниченного операторами `with`

Метод `__enter__` определяет, что должно происходить при входе в блок контекстного менеджера. Когда интерпретатор Python встречает оператор `with`, вызывается метод `__enter__` объекта контекстного менеджера

Пример использования метода `__enter__`

```
class Foo:
    Age = None
    Salary = 0

    def __enter__(self):
        return self.name

with Foo(name='asd') as f:
    print(f)
```

ИТОГИ

- ✓ Конструктор класса — это магический метод который присутствует всегда, даже если вы его не определили. Он поможет вам инициализировать класс с необходимым вам “настройками”
- ✓ Магические методы помогают обрабатывать различные события внутри класса не прибегая к написанию сторонних разработчиков для тех или иных событий, что помогает сократить время разработки