

```
class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    date_posted = models.DateTimeField(default=timezone.now)
    author = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.title
```

В Django есть класс Models, от которого наследуются все модели, которые вы создаете. В классе models есть метод save, который создает соответствующую таблицу в базе данных и сохраняет туда объекты.

С базами данных вы пока не умеете работать, но умеете работать с json.

Ваша задача:

Создайте класс Model, в котором будет метод save. Метод save должен создавать словарь со всеми значениями атрибутов класса и записывать его в файл json.

В данном примере это выглядит так: {'title': 'Какой-то заголовок', ..., 'author': 'какой-то автор'}.

Все атрибуты класса можно получить следующим образом:

In [1]:

```
class C1:
    title = '1'
    text = '2'
    author = '3'
```

In [2]:

```
dir(C1)
```

Out[2]:

```
['__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__',  
'author',  
'text',  
'title']
```

Или сразу так, получая только пользовательские атрибуты:

In [5]:

```
list(filter(lambda x: not x.startswith('__'), dir(C1)))
```

Out[5]:

```
['author', 'text', 'title']
```