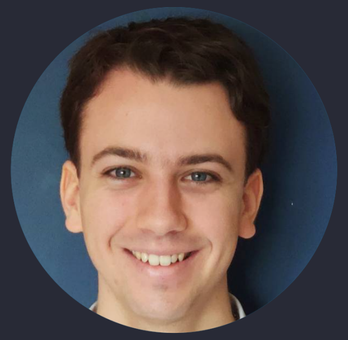
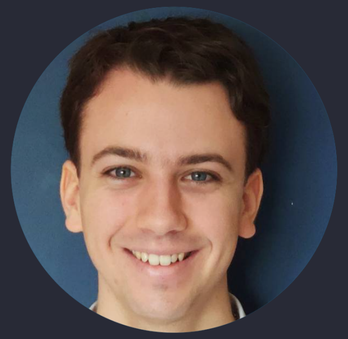
What are descriptors and why does Django need them?

PyCon Lithuania 🇱🇹 2024

by Rodrigo Girão Serrão



Rodrigo 🐍🚀



Rodrigo 🐍🚀

mathspp.com

@mathsppblog

What are descriptors and why does Django need them?

What are descriptors and why does ~~Django~~ Python need them?

Django uses descriptors

Proof:

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

Proof:

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

Django code ✅

Proof:

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

Descriptors 😐❓

Proof:

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

Descriptors 😐❓

Proof:

```
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

Descriptors 😐❓

WHY does Django need descriptors?

There must be “*something*” between saying an attribute exists and actually using it.

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

```python
from django.db import models


class City(models.Model):
    name = models.CharField(max_length=200)


class Person(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
    first_name = models.CharField(max_length=100)
```

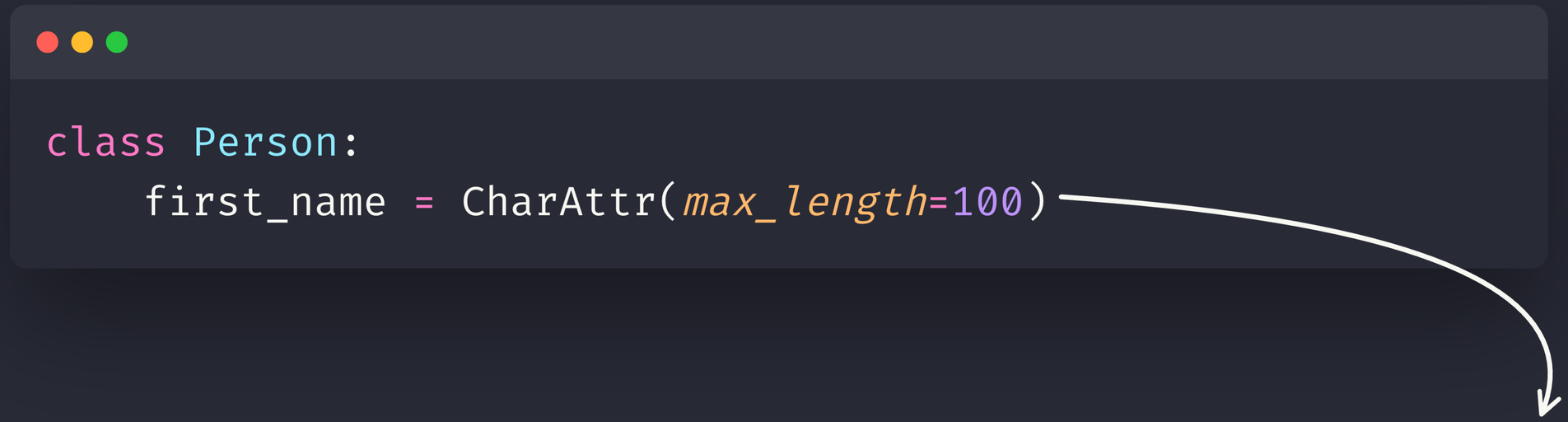
```python
from django.db import models




class Person(models.Model):
    first_name = models.CharField(max_length=100)
```

```
class Person:
    first_name = CharAttr(max_length=100)
```

```
class Person:
    first_name = CharAttr(max_length=100)
```



"*Something*" between saying the attribute exists and its actual value.

Let's write some code

Summary

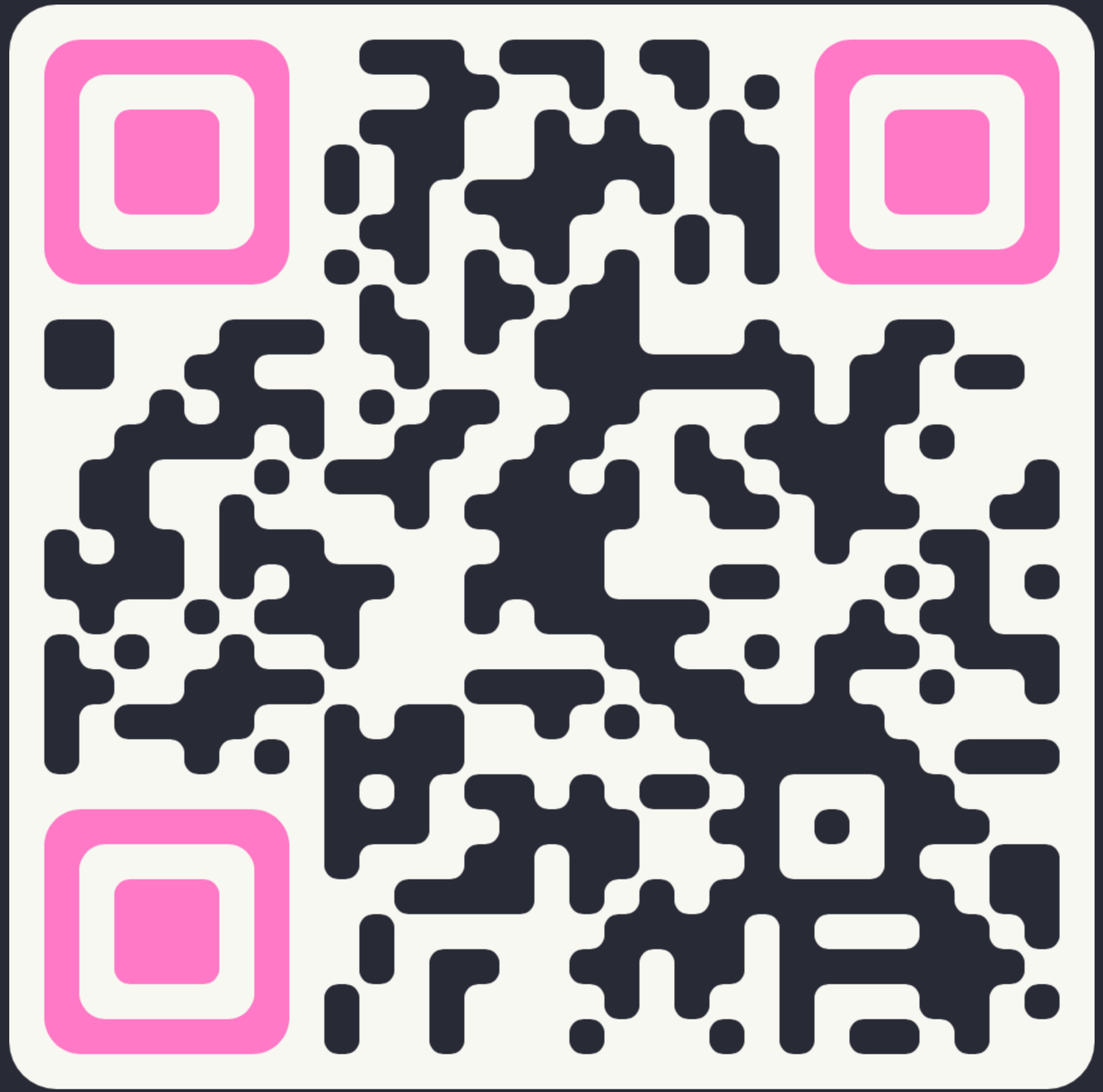
- `__get__` runs on attribute access;

- `__get__` runs on attribute access;
- `__set__` runs on attribute assignment;

- `__get__` runs on attribute access;
- `__set__` runs on attribute assignment; and
- `__set_name__` notifies descriptor of attribute name;

presented with snappify*

*no affiliation, they're just awesome



mathspp.com/talks