



**Jhonny Michel**

Developer

**@Evolux**

@jhonnymichel

jhonnymichel@outlook.com

---

# Aprendendo Python e Django Enquanto Front-end developer

Não achei um nome menor

---

# Background

- Hobbist game developer
- Já foi designer
- Entusiasta de UX
- Front-end developer
- Javascript <3
- Mistura inglês com português sem perceber, dsc!p
- Não sabia o que era um JOIN num banco de dados relacional até pouco tempo atrás

**NO FIM É TUDO TEXTO, OME**

# Ei, deu problema aqui no sistema

- Python
- Django
- PostgreSQL



# Flask

web development,  
one drop at a time

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6      return 'Hello, World!'
7
```

**django**

```
1  from django.conf.urls import url
2  from django.http import HttpResponse
3
4  def home(request):
5      return HttpResponse('Hello World')
6
7  urlpatterns = [
8      url(r'^$', home),
9  ]
10
```



# Django Basics to make web pages

- URL Patterns
- Django ORM
- Django Forms
- Views

# Django Basics to make web pages

- URL Patterns (tipo uma REGEX pensada pelo capeta)
- Django ORM (Ok, legal, abstração em cima do banco, êê)
- Django Forms (ModelForm, uma abstração em cima do DjangoORM)
- Views (as\_view, o que diabo é isso?)

```
1  from django.db import models
2
3  class Person(models.Model):
4      class Meta:
5          db_table="myprefix_person"
6
7          name = models.CharField(max_length=30)
8          # ...
9          mom = models.ForeignKey(
10             Person,
11             db_column='mom_id'
12         )
13
```

```
1  from django import forms
2  from .models import Person
3
4  class PersonForm(forms.ModelForm):
5      name = forms.CharField(required=False)
6      mom = forms.ModelChoiceField(
7          |   queryset=Person.objects.filter(gender='f')
8          |   )
9      class Meta:
10         |   fields = ('name', 'gender', 'nickname', 'age', 'mom')
11
```

```
def register_person(request):  
    form_data = {  
        'name': request.body.get('name'),  
        'gender': request.body.get('gender'),  
        'mom': request.body.get('mom_id'),  
    }  
  
    person_form = PersonForm(form_data)  
    if person_form.is_valid():  
        person_form.save()
```

# Python just feels good to code with

- Multiple Inheritance
- List Comprehension (eu quero chorar)
- Named arguments, \*args and \*\*kwargs
- Spread/Unpack operator (Python 3)

```
1
2      # Named arguments
```

```
3      create_person(name='fulano', mom_id=2)
```

```
4      # Javascript has to
```

```
5      create_person('fulano', null, null, null, 2)
```

```
6
7      # List Comprehension
```

```
8      used_names = [p.name for p in Person.objects.all()]
```

```
9
10     # Unpack Operator (Python 3)
```

```
11     merged_objects = { **obj1, **obj2 }
```

Obrigado!

@jhonnymichel  
jhonnymichel@outlook.com