

Desenvolvimento Web com Python e Django

por Marcos Gabriel da Silva





Olá, Mundo!

Marcos Gabriel

Aficionado por Python



/cafecompython





pythonTM

1.

MAS QUEM É ESSE TAL DE PYTHON?

Do ABC à Prática

Python

UMA SUPER LINGUAGEM CRIADA
EM 1991 POR GUIDO VAN ROSSUM



- DESENVOLVIMENTO DE SISTEMAS OPERACIONAIS (UBUNTU)
- DESENVOLVIMENTO DE APLICAÇÕES WEB (YOUTUBE, INSTAGRAM)
- DESENVOLVIMENTO DE JOGOS (CIVILIZATION IV)
- APRENDIZAGEM DE MÁQUINA E INTELIGÊNCIA ARTIFICIAL (GOOGLE BRAIN)
- CIÊNCIA DE DADOS (NETFLIX)
- GEOANÁLISE (ArcGIS)
- MOBILE (FoodPlus, AppBackup)
- E MUITO MAIS...



the ZEN of PYTHON

*“Bonito é melhor do que feio.
Explícito é melhor do que implícito.
Simples é melhor do que complexo.
Complexo é melhor do que
complicado...”*

Tim Peters



ABC

- » Comandos Básicos
- » Tipos de Dados
- » Operadores Aritméticos e Lógicos
- » Coleções
- » Estruturas de Decisão
- » Laços de Repetição
- » Funções e Classes

Comandos Básicos

`print()`, `import`, `input()`, `quit()`



Superpoderes no Console!

```
print('Olá,  
Mundo!')
```

Imprime no terminal
uma constante ou o
valor de uma
variável

```
import this
```

Carrega um módulo
ou biblioteca

```
input("Digite um  
número: ")
```

Solicita ao usuário
uma entrada de
dados

```
quit()
```

Finaliza o console
do Python

Tipos de Dados

string; int; float; complex; bool



Codificando o Mundo!

string

Sequência de caracteres

'Olá, Mundo'

"Tamo junto"

int

Dados Numéricos

10

aceita também:

- binários: 0**b**10
- octais: 0**o**7
- hexadecimal: 0**x**A

float

Números Reais

1.78

Notação Científica

2**e**2

complex

Números Complexos:

3**j**

bool

Verdadeiro ou Falso

True

False

Conversão de Tipos:

str()

int()

float()

Operadores

Aritméticos e Lógicos



Conte-me, compare-me e diga-me a verdade!

Aritméticos

+ - / // % * **

Atribuição:

=

Atribuição Aditiva:

+= -= *= /= //= %= **=

Lógicos

== != >= <=

> <

AND

NOT

OR

IS

Estruturas de Decisão

If, else, elif



Pra não ficar perdido nas ideias

if condição:

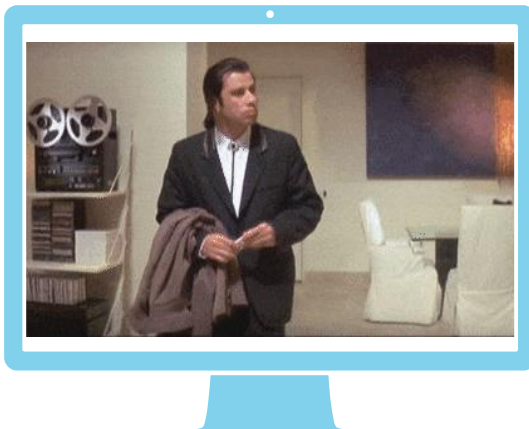
 procedimento

elif condição2:

 procedimento

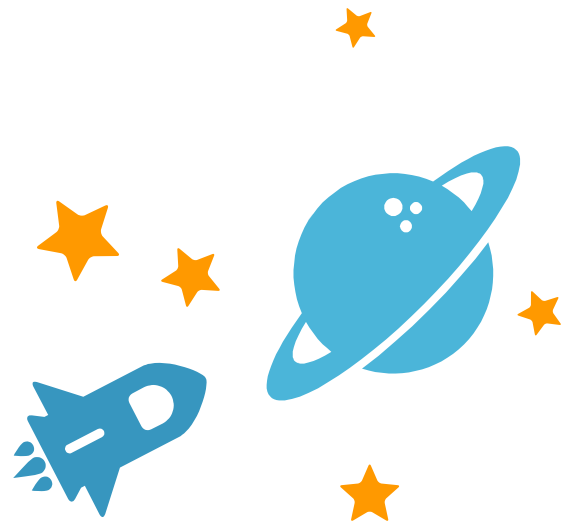
else:

 procedimento



Laços de Repetição

De, while



Correndo atrás do rabo (ou da coleira??)

for ~~sequência~~ (10):
prontidão

while ~~coleta~~ ~~edição~~:
perseguir ~~então~~ coleira



COLEÇÕES

Listas, Tuplas e Dicionários



Guardar para acumular!

Listas

```
lista = [ ]
```

Tuplas

```
tuplas = ( ) # são invariáveis
```

Dicionários:

```
dicionario = {chave:valor}
```





Falar é fácil...

'Bora pro código!



Atenção!

Funções e Classes

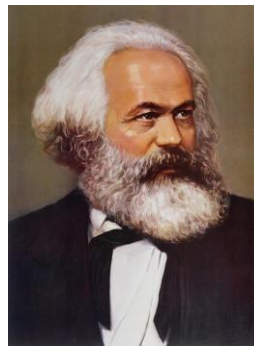
Isso não é doutrinação marxista!

Funções

```
def foo():  
    bar = 1+1  
    return bar
```

Classes

```
class Postagem:  
    def __init__(self, autor, titulo, texto)  
        self.autor = autor  
        self.titulo = titulo  
        self.texto = texto  
  
    def postar(self, data):  
        self.data = data  
        return print(f'Postado {self.data}')
```



django

2.

FINALMENTE: DJANGO

from Django import Blog

Pronto pra virar um Blogger?

- » Criando um Projeto
 - ◇ Estrutura de um projeto
 - ◇ Settings
- » Models - Views - Template (MVT)
- » Criando um App
- » Criando Templates
 - ◇ Gerenciando arquivos estáticos
- » Modelos e Formulários

Criando um projeto

1 - Crie um diretório.

2 - Execute o comando:

```
django-admin startapp meu_site .
```

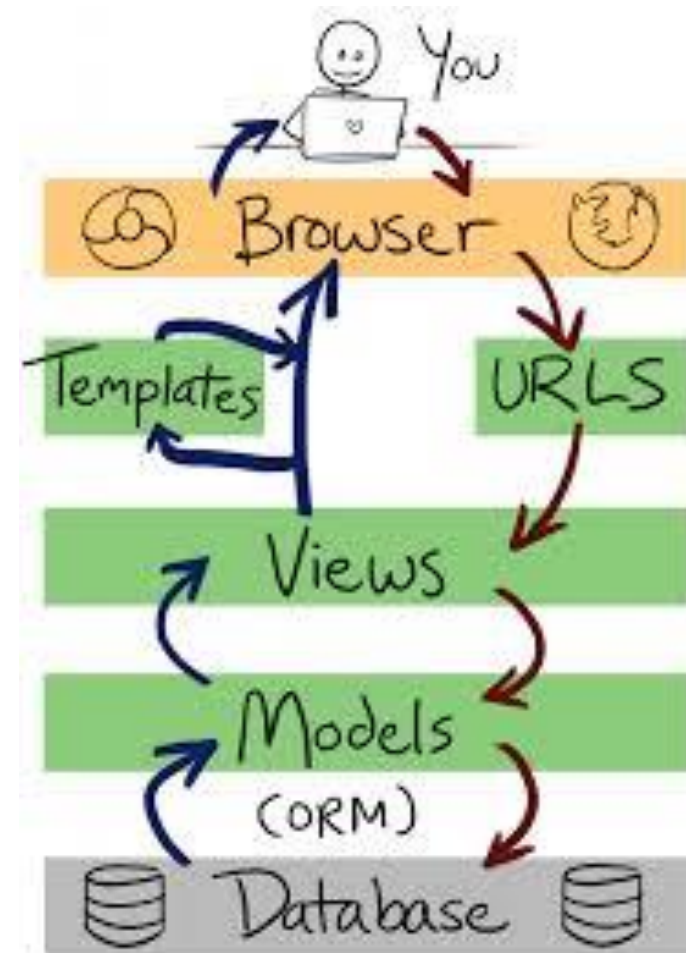
3 – Faça o teste:

```
python manage.py runserver
```

3 – Abra o navegador em:

```
localhost:8000
```


Models - Views - Template (MVT)



Criando um app

1 – Navegue até o diretório do arquivo `manage.py`

2 - execute o comando:

```
python manage.py startapp blog
```

3 – Instale o app no `Settings.py`

Criando a rota para o seu app

1 – Inclua seu app o caminho no urls.py:

```
urls.py x
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('blog.urls')),
7 ]
8
```

2 – Crie o arquivo urls.py no seu app:

```
urls.py x
1 from django.urls import path
2 from . import views
3 app_name = 'blog'
4
5 urlpatterns = [
6     path('', views.home, name='home'),
7 ]
```

Criando um view para o seu app

1 – Crie uma função home no views.py do seu app:

```
views.py x
1  from django.shortcuts import render
2
3
4  # Create your views here.
5  def home(request):
6      return render(request, 'blog/home.html')
```

Criando um template para o seu app

- 1 – Crie uma pasta chamada **templates** no diretório do seu app
- 2 – Crie uma pasta **blog** no diretório templates
- 3 – Crie um arquivo chamado **home.html**
- 4 – Abra o seu projeto no **Visual Studio Code**
- 5 – Escreva seu **html!**

Gerenciando arquivos estáticos

- 1 – Crie uma pasta **static** no diretório **blog**
- 2 – Crie uma pasta **blog** no diretório **static**
- 3 – Crie um arquivo chamado **estilo.css**
- 4 – Escreva seu **css**!
- 5 – Carregue o diretório **static** e o arquivo **estilo.css** no seu template **home.html**:

```
{% load static %}  
<link rel="stylesheet" href="{% static 'blog/css/estilo.css' %}">
```

Criando um Modelo

1 – Modifique o arquivo models.py do seu app

```
models.py x
1  from django.conf import settings
2  from django.db import models
3  from django.utils import timezone
4  # Create your models here.
5
6  class Postagem(models.Model):
7      autor = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
8      post_titulo = models.CharField(max_length=200)
9      post_texto = models.TextField()
10     data_publicacao = models.DateTimeField('data de publicação', default=timezone.now)
11
```

Registrando seu Modelo no Admin

1 – Modifique arquivo admin.py do seu app

```
admin.py x
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Postagem
5
6  admin.site.register(Postagem)
```


Migrando seu Modelo para o Banco de Dados

1 – Execute o comando no terminal:

```
python manage.py makemigrations
```

2 – Execute o comando no terminal:

```
python manage.py migrate
```

Acessando o Painel do Administrador

1 – Execute o comando no terminal:

```
python manage.py createsuperuser
```

2 – Crie um usuário de administrador

3 – Execute o comando no terminal:

```
python manage.py runserver
```

4 – Acesse o endereço no navegador:

```
localhost:8000/admin
```

Criando um Modelo

1 – Modifique o arquivo models.py mais uma vez

```
models.py x
1  from django.conf import settings
2  from django.db import models
3  from django.utils import timezone
4  # Create your models here.
5
6  class Postagem(models.Model):
7      autor = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
8      post_titulo = models.CharField(max_length=200)
9      post_texto = models.TextField()
10     data_publicacao = models.DateTimeField('data de publicação', default=timezone.now)
11
12     class Meta:
13         verbose_name_plural = 'Postagens'
14         ordering = ['-data_publicacao']
15
16     def __str__(self):
17         return self.post_titulo
```

Mostrando suas postagens no Navegador

1 – Modifique o arquivo views.py

```
views.py x
1  from django.shortcuts import render
2  from .models import Postagem
3
4  # Create your views here.
5  def home(request):
6      posts = Postagem.objects.all()
7
8      return render(request, 'blog/home.html', {'posts':posts})
```



Mostrando suas postagens no Navegador

1 – Modifique o arquivo home.html

```
<> home.html x
1  {% load static %}
2
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">
9      <title>Document</title>
10     <link rel="stylesheet" href="{% static 'blog/css/estilo.css' %}">
11 </head>
12 <body>
13     <h1>Olá, Mundo2 !</h1>
14
15     {% for post in posts%}
16         <p>{{post.post_titulo}}</p>
17         <p>{{post.autor}}</p>
18         <p>{{post.data_publicacao}}</p>
19         <p>{{post.post_texto}}</p>
20         <hr>
21
22     {% endfor %}
23
24 </body>
25 </html>
```



3.

BLOGANDO PRA VALER

from Django import Blog



¡Mi casa és su casa!

- 1 – Substitua o arquivo `home.html` pelo **nov**o `home.html`.
- 2 – Substitua o diretório `static` pelo **nov**o `static`,
- 3 – Delete o diretório **css** `antigo`.

Limpendo a Casa!

1 – Faça uma cópia de home.html e renomeie como base.html

2 – Use a mágica do template!

Tag lógica: {% %}

Tag de Variável: {{ }}

Pera lá! Que foto é essa?

1 – Modifique o arquivo models.py:

```
models.py x
1 from django.conf import settings
2 from django.db import models
3 from django.utils import timezone
4 # Create your models here.
5
6 class Postagem(models.Model):
7     autor = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
8     post_titulo = models.CharField(max_length=200)
9     imagem = models.ImageField(upload_to='imagens', blank=True, null=True)
10    post_texto = models.TextField()
11    data_publicacao = models.DateTimeField('data de publicação', default=timezone.now)
12
13    class Meta:
14        verbose_name_plural = 'Postagens'
15        ordering = ['-data_publicacao']
16
17    def __str__(self):
18        return self.post_titulo
```



Servindo arquivos dos usuários!

2 – Modifique o arquivo settings.py:

```
settings.py x
104 # Internationalization
105 # https://docs.djangoproject.com/en/2.2/topics/i18n/
106
107 LANGUAGE_CODE = 'pt-BR'
108
109 TIME_ZONE = 'America/Sao_Paulo'
110
111 USE_I18N = True
112
113 USE_L10N = True
114
115 USE_TZ = True
116
117
118 # Static files (CSS, JavaScript, Images)
119 # https://docs.djangoproject.com/en/2.2/howto/static-files/
120
121 STATIC_URL = '/static/'
122
123 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
124 MEDIA_URL = '/media/'
```

Servindo arquivos dos usuários! Fotos ou eu não acredito!

3 – Modifique o arquivo `urls.py` do seu projeto:

```
urls.py x
1  from django.contrib import admin
2  from django.urls import path, include
3  from django.conf import settings
4  from django.conf.urls.static import static
5
6  urlpatterns = [
7      path('admin/', admin.site.urls),
8      path('', include('blog.urls')),
9  ]
10
11  urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
12
```

Servindo arquivos dos usuários! Fotos ou eu não acredito!

4 – Modifique o arquivo home.html do seu app:

```
<!-- Postagens -->
<div class="row text-center">

  {% for post in posts%}
  <div class="col-lg-3 col-md-6 mb-4">
    <div class="card h-100">
      
      <div class="card-body">
        <h4 class="card-title">{{post.post_titulo}}</h4>
        <p class="card-text">{{post.post_texto|truncatewords:12}}</p>
      </div>
      <div class="card-footer">
        <a href="#" class="btn btn-primary">Leia mais!</a>
      </div>
    </div>
  </div>
  {% endfor %}
<!-- /.row -->
</div>
```

Cadê o resto do meu texto???

1 – Modifique o arquivo views.py do seu app:

```
views.py x
1  from django.shortcuts import render, get_object_or_404
2  from .models import Postagem
3
4  # Create your views here.
5  def home(request):
6      posts = Postagem.objects.all()
7
8      return render(request, 'blog/home.html', {'posts':posts})
9
10 def postagem(request, pk):
11     post = get_object_or_404(Postagem, pk=pk)
12     return render(request, 'blog/post_detalhe.html', {'post':post})
13
```

Cadê o resto do meu texto???

2 – Modifique o arquivo urls.py do seu app:

```
urls.py  x
1  from django.urls import path
2  from . import views
3  app_name = 'blog'
4
5  urlpatterns = [
6      path('', views.home, name='home'),
7      path('postagem/<int:pk>/', views.postagem, name='postagem')
8  ]
9  |
```

Mostre o texto no template:

3 – Modifique o arquivo
post_detalhe.html do seu app:

```
<? post_detalhe.html x
6  {% block content%}
7  <!-- Conteúdo da Página -->
8  <div class="container">
9    <div class="row">
10     <!-- Postagem -->
11     <div class="col-lg-8">
12       <!-- Título -->
13       <h1 class="mt-4">{{post.post_titulo}}</h1>
14       <!-- Autor -->
15       <p class="lead">
16         by
17         <a href="#">{{post.autor}}</a>
18       </p>
19
20     <hr>
21
22     <!-- Data/Hora -->
23     <p>{{post.data_publicacao}}</p>
24
25     <hr>
26
27     <!-- Imagem -->
28     
29
30     <hr>
31
32     <!-- Texto -->
33     <p class="lead">{{post.post_texto}}</p>
34     <!-- Fim do Texto -->
35     <!-- Fim da Postagem -->
```

Formulários: Comenta lá, plz!

1 – Modifique o arquivo models.py do seu app:

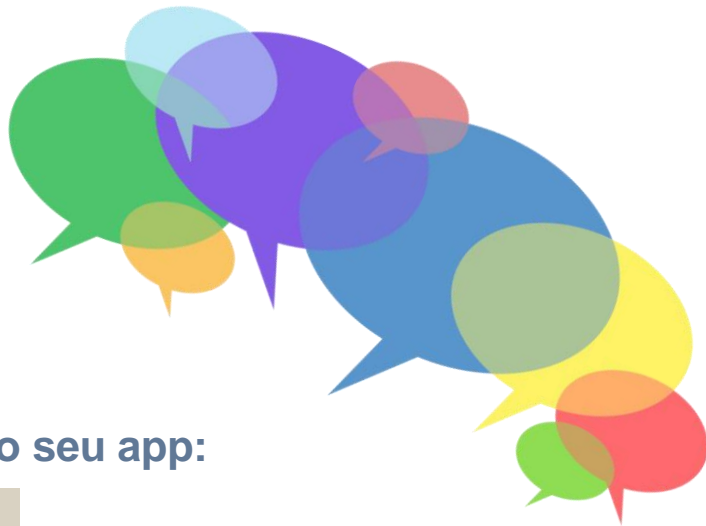
```
models.py x
17     def __str__(self):
18         |     return self.post_titulo
19
20
21     class Comentario(models.Model):
22         |     post_ref = models.ForeignKey('Postagem', on_delete=models.CASCADE)
23         |     coment_autor = models.CharField(max_length=40)
24         |     coment_texto = models.TextField()
25         |     coment_publicacao = models.DateTimeField('data de publicação', default=timezone.now)
26
27         |     class Meta:
28         |         |     verbose_name = 'Comentário'
29         |         |     ordering = ['-coment_publicacao']
30
31     def __str__(self):
32         |     return self.coment_autor
```



Comenta vai, vai...

2 – Crie e modifique um arquivo forms.py no seu app:

```
forms.py x
1  from django import forms
2  from .models import Comentario
3
4
5  class ComentarioForm(forms.ModelForm):
6      class Meta:
7          model = Comentario
8          fields = ['coment_autor', 'coment_texto']
```



Criando a regra de comentários!


3 – Modifique o arquivo views.py do seu app:

```
views.py x
1 from django.shortcuts import render, get_object_or_404
2 from .models import Postagem, Comentario
3 from .forms import ComentarioForm
4
5 # Create your views here.
6 def home(request):
7     posts = Postagem.objects.all()
8
9     return render(request, 'blog/home.html', {'posts': posts})
10
11 def postagem(request, pk):
12     form = ComentarioForm()
13     post = get_object_or_404(Postagem, pk=pk)
14     coments = Comentario.objects.filter(post_ref=post)
15     context = {'post': post, 'form': form, 'coments': coments}
16
17     if request.method == 'POST':
18         form = ComentarioForm(request.POST)
19         if form.is_valid():
20             coment=form.save(commit=False)
21             coment.post_ref = post
22             coment.save()
23             form = ComentarioForm()
24             return render (request, 'blog/post_detalhe.html', context)
25     else:
26         return render(request, 'blog/post_detalhe.html', context)
27
```

Registrando os comentários no Admin!

4 – Modifique o arquivo admin.py do seu app:

```
admin.py x
1  from django.contrib import admin
2
3  # Register your models here.
4  from .models import Postagem, Comentario
5
6  admin.site.register(Postagem)
7  admin.site.register(Comentario)
```



Criando o formulário e mostrando os comentários na página!

5 – Modifique o arquivo `post_detalhe.html` do seu app:

```
< post_detalhe.html x
38
39 <!-- Formulário de Comentário -->
40 <div class="card my-4">
41   <h5 class="card-header">Deixe um comentário:</h5>
42   <div class="card-body">
43     <form method="POST">
44       {% csrf_token %}
45       <div class="form-group">
46         {{ form.as_p }}
47         <button type="submit" class="btn btn-primary">Enviar</button>
48       </div>
49     </form>
50   </div>
51 </div>
52 <!-- Fim do Formulário de Comentário -->
53
54 <!-- Comentários -->
55 {% for com in coments%}
56 <div class="media mb-4">
57
58   <div class="media-body">
59
60     <h5 class="mt-0">{{ com.coment_autor }} <i class="far fa-comment"></i></h5>
61     {{ com.coment_texto }}
62   </div>
63 </div>
64 {% endfor %}
65 <!-- Fim de Comentários -->
66
```

Deixando tudo belo!

6 – Modifique o arquivo forms.py no seu app:

```
forms.py x
1  from django import forms
2  from .models import Comentario
3
4
5  class ComentarioForm(forms.ModelForm):
6      class Meta:
7          model = Comentario
8          fields = ['coment_autor', 'coment_texto']
9          labels = {
10             'coment_autor': '',
11             'coment_texto': '',
12
13         }
14         widgets = {
15             'coment_autor': forms.TextInput(
16                 attrs={
17                     'class': 'form-control',
18                     'type': 'text',
19                     'placeholder': 'Digite seu nome',
20                 }
21             ),
22             'coment_texto': forms.Textarea(
23                 attrs={
24                     'class': 'form-control',
25                     'rows': '3',
26                 }
27             ),
28         }
29
```

Conseguimos!!

Será? Bola pra frente!

Dúvidas? Sugestões? Googla lá:

www.djangoproject.com

www.python.org

www.github.com/cafecomp/python

Créditos

Um viva ao pessoal que disponibilizou recursos digitais gratuitos na internet:

- » Templates por [SlidesCarnival](#), [BlackrockDigital](#)
- » Imagens via [Unsplash](#), [Google Images](#)
- » Gifs via [Giphy](#)