

Criação de página web com Django

Criação de um template global e importar elementos da app *

- ▷ Criar uma nova pasta com o nome do projeto 'meusite' e criar um ambiente virtual nela.
- ▷ Instalar o django (>> pip install django)
- ▷ Criar o projeto meusite (>>django-admin startproject meusite .)
- ▷ Criar uma app de nome blog (>>python manage.py startapp blog)
- ▷ Registrar a app blog nos ficheiros settings.py e urls.py do projeto meusite
- ▷ Repete os passos executados nas páginas 7 e 8 da ficha anterior (1.º a 6.º, incluindo a criação do ficheiro index.html), aquando da criação da app produto.
- ▷ Inicia o servidor e testa se a página index.html abre corretamente.

*

- ▷ Cria, na raiz do projeto 'meusite' (a 1.ª pasta criada) uma pasta de nome 'paginas', onde serão armazenadas as páginas html.

- ▷ Abre o ficheiro settings.py do projeto 'meusite'.

Procura o dicionário de nome TEMPLATES e altera o valor da chave 'DIRS', indicado ao django onde procurar os ficheiros de templates. Adiciona o código seguinte no valor da chave: BASE_DIR / 'PAGINAS', que significa o path da raiz do projeto /caminho até paginas.

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [BASE_DIR / 'PAGINAS'],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```

- ▷ Cria um ficheiro de nome 'base.html', dentro da pasta 'paginas'
- ▷ Dentro do ficheiro criar um template para html5 (escrever no ficheiro 'html' e seleccionar html 5)
- ▷ No template procede às seguintes alterações:
 - Altera a linguagem de 'eng' para 'pt'



O Django tem uma linguagem própria para renderizar o código html. Neste caso iremos criar alguns **blocos** para poder inserir conteúdo de html provindo de outros ficheiros html que estão no projeto.

Criação de blocos em tags html (sintaxe):

```
{% block 'nome_do_bloco' %}{%endblock%}
```

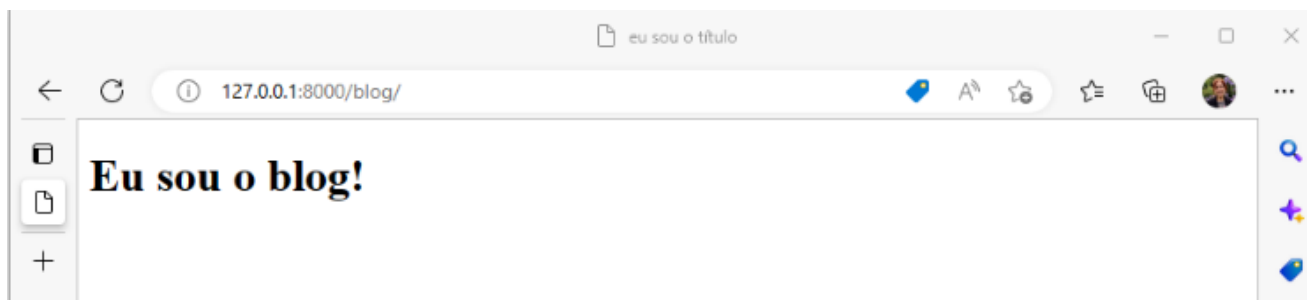
- ▷ Criar no ficheiro base.html os blocos 'titulo' e 'conteudo', respetivamente, nas tag title e body

```
meusite > paginas > <> base.html > ...
1  <!DOCTYPE html>
2  <html lang="pt">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>{%block 'titulo' %}{%endblock%}</title>
8  </head>
9  <body>
10     {%block 'conteudo' %}{%endblock%}
11 </body>
12 </html>
```

- ▷ No ficheiro index.html 'herdar' o template existente em base.html:

```
index.html - meu_site - Visual Studio Code
index.html x  <> base.html  settings.py
django > templates > blog > <> index.html > ...
1  {%extends 'base.html'%}
2
3  {%block 'titulo'%}eu sou o título{%endblock%}
4
5  {%block 'conteudo'%}
6  <h1>Eu sou o blog</h1>
7  {%endblock%}
```

- ▷ Executa o servidor para visualizares o resultado



- ▷ Observa o código-fonte da página (ctrl+u):

```
view-source:127.0.0.1:8000/blog/

Moldar linhas ☐

1 <!DOCTYPE html>
2 <html lang="pt">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>eu sou o título</title>
8 </head>
9 <body>
10
11   <h1>Eu sou o blog!</h1>
12
13 </body>
14 </html>
15
```

- ▷ Acrescenta agora, no ficheiro de base (base.html) do template, um estilo ao corpo da página:

```
ginas > <> base.html > html > body
1 <!DOCTYPE html>
2 <html lang="pt">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>{%block 'titulo'%}{%endblock%}</title>
8   <style>
9     body{
10       font-size: 40px;
11       background-color: coral;
12       color: blue;
13     }
14   </style>
```

Este código corresponde à mistura de código de dous ficheiros – o ficheiro base.html (ficheiro principal) e o ficheiro index.html, que adiciona conteúdo ao primeiro.

▷ Executa o servidor e observa o resultado:



Método – criar uma app para homepage:

Criar uma nova app de nome 'home';

Registá-la em settings e na url (este último a fazer de modo direto)

```
settings.py  urls.py ficha8 1 x  base.html blog...  index.html ...templates
ficha8 > urls.py > ...
9  Class-based views
10  1. Add an import: from other_app.views import Home
11  2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12  Including another URLconf
13  1. Import the include() function: from django.urls import include, path
14  2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15  """
16  from django.contrib import admin
17  from django.urls import path, include
18  from home import views
19
20  urlpatterns = [
21      path('', views.index),
22      path("admin/", admin.site.urls),
23      path("blog/", include("blog.urls")),
24      # path("home/", include("home.urls")),
25  ]
```

Ir para o ficheiro views.py da app criada :

```
views.py - ficha8_10794 - Visual Studio Code
home > views.py > ...
1  from django.shortcuts import render
2
3  def index(request):
4      return render(request, 'home/index.html')
5
```

Criar uma pasta templates da app e dentro dela a pasta home e ainda dentro da pasta home o ficheiro index.html.

Não é necessário no ficheiro urls.py da app criada, proceder à chamada do método *index*, porque foi feita uma invocação direta a partir do views.py

Executar o servidor:

>>python manage.py runserver

Bibliografia

<https://www.w3schools.com/django/index.php>

<https://stackoverflow.com/questions/63259560/extends-base-html-doesnt-loaded-in-django>

https://www.w3schools.com/django/django_master_template.php

Curso de Python 3 do básico ou Avançado - Udemy