Agora vamos carregar os arquivos estáticos da aplicação. O caminho será similar ao dos arquivos *HTML*.

Na linha 61 de "setups > settings.py", informamos que há uma pasta onde manteremos todos os nossos códigos *HTML*. Faremos um processo semelhante, mas para indicar o caminho dos arquivos estáticos.

Vamos descer a página até chegarmos à linha 122, onde lemos STATIC\_URL = 'static/'. Vamos informar a pasta que receberá esses arquivos, escrevendo, na linha 124, STATICFILES\_DIRS, passando o caminho os.path.join, com os parâmetros BASE\_DIR, 'setup/static'.

Precisamos passar, também, o caminho absoluto para que o diretório consiga pegarr os arquivos estáticos. Para isso, vamos inserir a "raiz" dos caminhos com STATIC\_ROOT, com o caminho `os.path.join(BASE\_DIR, 'static'):

STATIC\_URL = 'static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'setup/static')

]

STATIC\_ROOT = os.path.join(BASE\_DIR, 'static')Copiar código

Na pasta "setup", criaremos uma pasta chamada "static". Agora criamos o diretório que servirá para a manipulação dos arquivos estáticos.

Nosso projeto já tem *HTML* e *CSS*. Vamos copiar, por isso, os arquivos das pastas "assets" e "styles" e levá-los para a pasta "setup > static". Vamos abrir o terminal, executar "Ctrl + C", para fazê-lo funcionar, e carregar a *venv* com o comando python manage.py runserver.

De volta ao navegador, se atualizarmos a página, nada mudará. Apesar disso, já temos os arquivos estáticos dentro da aplicação. Existe um comando para solicitar que o *Django* manipule os arquivos estáticos da aplicação, para que possamos visualizá-los.

Trata-se do comando collectstatic, que pode ser encontrado acessando a lista de comandos via *CLI*, passando python manage.py help. De volta ao terminal, vamos rodar o comando python manage.py collectstatic:

python manage.py collectstaticCopiar código

Depois de executarmos o comando, percebemos a criação de uma pasta chamada "static", relacionada ao STATIC\_ROOT que criamos. Agora, já podemos usar os arquivos estáticos no *Django*. Se rodarmos o servidor outra vez, com python manage.py runserver, não vamos notar alterações no projeto.

Para realizar as alterações, vamos acessar "templates/galeria > index.html". Vamos adicionar o código {% load static %}, que solicita o carregamento dos arquivos estáticos, à primeira linha do arquivo.

Vamos carregar, inicialmente, o arquivo "style.css", o caminho para encontrá-lo é "static > styles > style.css". Na linha 13 do código, vamos adicionar {% static após o href e %} após o fechamento das aspas, o que identifica um arquivo estático a ser subido.

Também vamos colocar styles/style.css entre aspas simples e levar as aspas duplas para depois do fechamento das chaves:

<link rel="stylesheet" href="{% static '/styles/style.css' %}">Copiar código

Obs: Quando usamos código *Python* junto ao *HTML*, chamamos de "embedado". Porque fizemos o *embedding* do código *Python* ao *HTML*.

De volta ao navegador, ao atualizarmos o projeto, veremos que o estilo da aplicação começou a mudar. No próximo vídeo, adicionaremos as imagens que estão faltando.