

BLOG

[Início](#) » [Blog](#) » Criar Site em Python com Flask – Criação e Publicação de Site

Postado em [Python](#), em 4 de novembro de 2021

CRIAR SITE EM PYTHON COM FLASK – CRIAÇÃO E PUBLICAÇÃO DE SITE

Na aula de hoje eu vou te mostrar como criar site em Python com Flask e vou te ensinar a fazer a publicação do seu site utilizando o heroku!

Caso prefira esse conteúdo no formato de vídeo-aula, assista ao vídeo abaixo ou acesse o nosso [canal do YouTube](#)!



Para receber por e-mail o(s) arquivo(s) utilizados na aula, preencha:

Enviar

Como Criar e Publicar um Site em Python

Hoje eu quero te ensinar como criar site em Python e como publicar site em Python, então além de criar vamos publicar esse site!

Está preparado para criar e publicar site em Python pela primeira vez? Então vamos a aula que temos bastante conteúdo!

Criar Site em Python com Flask

O primeiro passo como você já deve ter adivinhado é a instalação da biblioteca Flask, então vamos escrever no terminal **pip install Flask**.

OBS: Lembrando que para esse projeto nós vamos utilizar o PyCharm, então se você ainda não tem instalado ou não sabe como instalar basta [clique aqui](#).

Caso você queira um material bem mais detalhado e explicando tudo sobre o flask você pode [clique aqui](#) ou procurar no Google por Flask Python.

Procure o que você quiser sobre Excel, VBA, Power BI ou Python:

Quer aprender tudo de Excel para se tornar o destaque de qualquer empresa?

Preencha com o seu e-mail*

QUERO ME CADASTRAR

CATEGORIAS

[Dicas da Hashtag](#)

[Excel Avançado](#)

[Excel Básico](#)

[Excel Intermediário](#)

[Modelos de Planilhas](#)

[Power BI](#)

[PowerPoint](#)

[Python](#)

[SQL](#)

[VBA](#)

POSTS RECENTES

[Curso de Power Apps Aula 1 – O que é Power Apps](#)

[Portfólio de Projetos no Power BI – Aprenda a Criar o Seu Portfólio](#)

[Gravador de Macros no Excel – Como Criar Macros sem Programar?](#)

[Aprenda a Programar do Jeito Certo – Como Aprender Python?](#)

[Fórmula SOMASES: como somar com condições no Excel](#)

Dessa forma você vai ter acesso a toda documentação do Flask para verificar como funciona e todos os detalhes que precisa.



Vou mostrar o código que nós vamos utilizar aqui, mas vale lembrar que ele está disponível para download!



Código para a criação do site em Python

Vamos agora a explicação do código para que depois você possa fazer suas alterações de acordo com sua necessidade.

Biblioteca e criação do site

Nessa parte inicial nós temos a importação da biblioteca Flask e logo abaixo já temos o comando para a criação do site, que é **`app = Flask(__name__)`**.

A documentação do próprio Flask já orienta deixarmos o nome dessa variável como app mesmo, então estamos apenas seguindo o padrão.

Essa parte do `__name__` e do if que vamos visualizar no final do código nós explicamos um pouco mais detalhada em um outro post aqui do canal, caso tenha interesse basta [clique aqui](#) para visualizar esse conteúdo.

Após a criação da página nós temos 3 elementos: route, que é o caminho do nosso domínio. Se você não colocar nada após a barra vai significar que essa é a página inicial. Então vamos ter qual link vai abrir qual página do site.

O segundo elemento é o que vai aparecer dentro daquela página, então é de fato o que será mostrado ao usuário.

E o terceiro elemento que é o template, que é necessário por conta do framework para facilitar a criação das nossas páginas utilizando o html ao invés de escrever tudo isso diretamente dentro do código em Python.

Abaixo disso nós temos algumas funções que foram definidas utilizando o **`@app.route`**.

Definindo as páginas do site

O **app** que estamos utilizando é por conta do nome do nosso site e o **.route** é porque estamos indicando o caminho daquela página como já vimos.

O primeiro por exemplo temos apenas a **/**, que como comentei anteriormente vai ser a nossa página inicial.

Mas podemos criar outras páginas, como é o caso de contatos por exemplo.

Nessa parte dentro do **return** você vai poder escrever normalmente, no entanto, nós temos a opção de colocar um **html** para que você não tenha que fazer tudo dentro dessa parte.

Por esse motivo é que vamos colocar esse **render_template** com o nome do html, só que para isso funcione nós vamos ter que criar uma pasta chamada **templates**, na mesma pasta onde está o seu código.

Pasta templates com os arquivos em html

E dentro dessa pasta chamada templates é que nós vamos criar os arquivos html para poder de fato escrever o código de cada uma das páginas.

Tendo isso você já vai conseguir criar e personalizar suas páginas de acordo com a sua necessidade.

Na última parte de usuário por exemplo, você vai poder criar uma página para cada usuário com suas devidas informações, da mesma forma que uma rede social faz.

A primeira vez que você rodar o código ele vai te retornar um “código”, mas é um site, você pode tanto clicar nele quanto copiar e colar no seu navegador.

Página inicial do site

Então aqui você já vai poder visualizar o que fez no seu site, tanto na página inicial quanto nas outras páginas.

Para as outras páginas você só precisa colocar a barra (**/**) + o nome da página, como por exemplo **/contatos**.

Agora nós vamos para a última parte do código que é bem útil e importante para quando você for fazer suas alterações.

Parte final do código para rodar o site

Essa parte final dentro do if, que deixamos o link lá em cima caso queira saber o que de fato significa essa estrutura, vai permitir com que você possa rodar o seu site e que você possa utilizar o debug.

Como assim? Ele vai permitir com que o site atualize a medida em que você vai editando seu código, assim não vai precisar sempre ficar rodando o código inúmeras vezes.

OBS: É possível que tenha algum erro no código, mas nesses casos basta executar novamente que vai funcionar.

Criado o seu site nós vamos precisar publicar ele para que ele tenha um domínio sem aqueles números, neste caso nós vamos utilizar o servidor [heroku](#).

Lá você vai criar sua conta seguindo o passo a passo do site e depois fazer o login na sua conta.

Criando o app no site heroku

Feito isso você vai ter uma tela parecida com essa e não deve ter nada, pois a conta é nova, mas basta clicar em **New** e depois em **Create new app**.

Configurando o app

Em seguida você vai ser levado para a página onde vai criar o nome do seu app, que é a **url** que vai utilizar para acessá-lo, depois basta clicar em **Create app**.

Assim que criar o seu app vai ser direcionado a uma página onde temos alguns passos a seguir para fazer o deploy do site.

Passo a passo para instalação do Heroku CLI e publicação do site

Dentro do site você vai visualizar isso normalmente, veja que você vai ter que instalar o **Heroku CLI**, então já vai poder clicar no link para poder instalar esse recurso.

Nessa parte de instalação ele já mostra os passos tanto Mac, Windows e Linux, então é bem tranquilo.

IMPORTANTE: Você vai notar um aviso informando que você precisa do **Git** instalado para poder continuar, então terá que fazer essa instalação antes de instalar o CLI. Não se preocupe que na página de download do **Heroku** ele já te dá o link de instalação do Git também!

Após a instalação dos dois será necessário fechar o PyCharm e abrir novamente para que ele reconheça tanto o Git quanto o Heroku.

Para garantir que os dois foram reconhecidos pelo PyCharm você pode escrever tanto Git quanto Heroku no terminal que vai conseguir visualizar os comandos de cada um deles.

OBS: Nesse caso se não funcionar verifique se de fato fez a instalação correta dos dois e reiniciar o computador pode ajudar também!

Já estamos quase terminando os preparativos para poder rodar tudo de forma correta, só que antes vamos precisar criar outro arquivo.

Na mesma pasta onde temos o nosso código, temos que criar um arquivo chamado **Procfile** sem extensão mesmo e vamos abrir ele no formato de texto.

IMPORTANTE: Esse arquivo não vai ter extensão, então se você não estiver usando o PyCharm pode criar como arquivo txt para fazer o passo e depois deletar o .txt para excluir a extensão.

Dentro desse arquivo você vai escrever o seguinte texto:

web: gunicorn nome_do_seu_arquivo:app

Lembrando que onde está esse **nome_do_seu_arquivo** é de fato para você colocar o nome do arquivo que criou tirando o **.py**, então só o nome do arquivo mesmo.

Depois de escrever esse texto dentro do arquivo nós vamos voltar ao nosso código e instalar o gunicorn no terminal, para isso basta escrever **pip install gunicorn**.

Como já fizemos a instalação do gunicorn nós já podemos criar o nosso arquivo **requirements.txt**.

Para essa criação nós vamos utilizar o seguinte código no terminal: **pip freeze > requirements.txt**.

O pip freeze vai criar uma lista do que você tem instalado no seu projeto e vai escrever dentro desse arquivo.

Lista do que está instalado dentro do projeto

Ao abrir o arquivo você vai poder visualizar tudo que está instalado nesse projeto, incluindo o gunicorn que acabamos de instalar!

Agora nós terminamos as configurações e só vamos precisar levar o nosso projeto para o online, para isso nós vamos seguir aquele passo a passo do Heroku.

Prosseguindo com a publicação do site

Basta ir seguindo o passo a passo dos códigos escrevendo-os no terminal do PyCharm.

IMPORTANTE: Na parte do **git init** se você nunca tiver executado nada do git no seu computador você vai ter que fazer uma configuração com os dois códigos que vou mostrar para configura um e-mail e um nome de usuário. O primeiro código é: **git config --global user.email "seu e-mail"**. O segundo código é: **git config --global user.name "seu nome de usuário"**.

O passo após o **git init** depois do **-a** você vai ter que digitar o nome do seu site, no caso da imagem está aparecendo o nome do site que eu utilizei, então você precisa colocar o seu para que funcione.

Todos os códigos e procedimentos até o momento só serão necessários na primeira vez que você for fazer o deploy do seu site.

Nas próximas vezes você só vai precisar garantir que fez o login no heroku e seguir as últimas 3 linhas de código do passo a passo!

OBS: Dentro do código de commit nós geralmente colocamos entre aspas o que foi alterado em cada uma das versões para facilitar.

Após terminar os códigos você já vai ter o link para acessar o site que você criou.

Site publicado e com link para acesso

Como estamos utilizando a versão gratuita do heroku o seu site vai ter o nome que você colocou **.herokuapp**.

Agora basta clicar no link para que você possa acessar o seu site! E com o site criado você pode alterar entre os routes que criamos lá no início.

Isso quer dizer que agora o seu site foi publicado e você pode acessá-lo de outra máquina ou passar o link para que outra pessoa possa acessar!

Conclusão de Como Criar Site em Python com Flask

Essa foi aula longa, mas a ideia é que você consiga fazer a criar de site em Python com Flask e claro, que você possa fazer suas alterações e ajustes para atender o que precisa!

É importante que inicialmente você siga o passo a passo que foi mostrado e depois vá fazendo as suas alterações.

São alguns passos chatos e demorados, mas depois que fizer a primeira vez você só vai precisar logar no heroku e escrever as últimas 3 linhas de código!

Agora deixo com você a tarefa de criar o seu próprio site em Python com Flask, depois você pode mandar para os amigos para mostrar o seu primeiro site em Python!

Hashtag Treinamentos

Para acessar outras publicações de Python, [clique aqui!](#)

Quer aprender mais sobre Python com um minicurso gratuito?

Preencha com o seu e-mail*

QUERO ME CADASTRAR



Fale Conosco

Cursos Online

Carol Coe

livstavi@hashtagtreinamentos.com

☎ (21) 99865-8852

(WhatsApp)

Endereços

Espaço Coworking Rio, Rua Uruguai, nº 94 – 18º andar – Edifício Livreiro / CEP: 20.050-091 – Rio de Janeiro/RJ

Proteção de Dados

[Política de Privacidade](#)

[Código de Conduta](#)

Newsletter

Seu primeiro nome*

Seu melhor e-mail*

ENVIAR