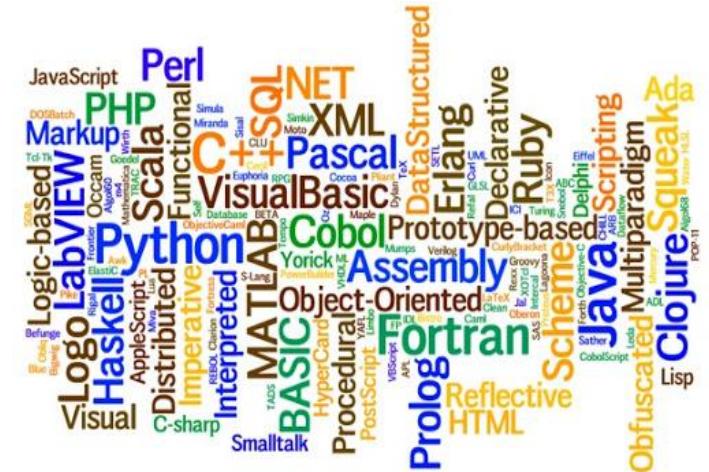




Programando Front-End com Linguagem Python

Prof. Me. Massaki de O. Igarashi
prof.massaki@gmail.com



Agenda desta oficina!

01

- Nosso objetivo!

02

- Definições importantes

03

- 5 Passos para Front-end Python ONLINE

04

- 8 Passos para Front-end Python OFF-LINE

05

- A Biblioteca ACTLIB

Nossos Objetivos de Hoje!

1º OBJETIVO:

Programar usando linguagem Python e publicar um aplicativo web simples!

Exemplo 1

MEU 1º APlicativo Web PYTHON

ACTlib Versão 0.1

Nome:

Não esqueça de preencher seu nome 0/30

2º OBJETIVO/DESAFIO:

Criar Dashboard para respostas Gforms

Pesquisa Teste - Aula Python 01

massaki.igarashi@gmail.com Mudar de conta

Não compartilhado

* Indica uma pergunta obrigatória

Sua idade *

Sua resposta

Você nasceu em Jundiaí? *

SIM, sou Jundiaiense

NÃO, sou de outra cidade e/ou Estado

Escolha uma qualidade que busca em alguém *

Companheirismo

Determinação

Disciplina

Entusiasmo

Honestidade

Paciência

Respeito

Um sonho

Sua resposta

Enviar

Limpar formulário



Um resumo sobre Python

A linguagem Python

foi criada pelo holandês Guido van Rossum por volta de 1990 e tem como principal filosofia, a simplicidade e legibilidade do código. Não obstante, é utilizada por grandes empresas como YouTube, Google, Yahoo e Microsoft.



Python é uma das linguagens mais populares hoje. Existem diversas bibliotecas para análise de dados e materiais para auxiliar o desenvolvimento de algoritmos. Além disso, é uma linguagem poderosa ... e rápida; interage bem com outras, é amigável, fácil de aprender e é de código aberto.

“ ”



Na década de 1970, a BBC tinha um programa de TV popular do qual Van Rossum era um grande fã chamado Fly Circus de Monty Python, ou apenas Monty Python para os íntimos. Assim, quando desenvolveu a linguagem, ele pensou que precisava de um nome que fosse curto, único e um pouco misterioso, e por algum motivo que só ele conhecia, decidiu chamar o projeto de 'Python'.

A linguagem foi batizada em homenagem ao programa de humor britânico “*Monty Python*” (1970).

Para mais informação:
<http://www.montypython.com>



O grupo Monty Python foi muito famoso na Inglaterra, na década de 70 e recebeu muitas críticas de conservadores por causa do seu humor ácido e irreverente.

Você pode conhecer a licença do Python e fazer o download de sua última versão no seu site oficial www.python.org.

Um resumo sobre Python

Python é uma **linguagem de scripts** que permite executar e testar um código imediatamente depois de escrevê-lo, facilitando bastante as atualizações. Em outras palavras, linguagens de script são linguagens interpretadas. O **interpretador executa o programa apenas traduzindo comandos em uma série de uma ou mais sub-rotinas** que depois são traduzidas em outras linguagens.

Um script é uma **coleção de comandos em um arquivo** projetada para ser executada como um programa e não pelo processador do computador, como acontece com linguagens compiladas. **O arquivo pode conter funções e módulos variáveis, mas a ideia central é que ele possa rodar e cumprir uma tarefa específica a partir de uma linha de comando.** Um exemplo clássico disso são as linguagens para prompts de comando, como no arquivo batch Windows.

Em geral, é mais rápido e fácil programar usando uma linguagem de script do que uma mais estruturada e compilada, como C ou C++.

Interpretadores X Compiladores?

Os **interpretadores** passam pelo programa **linha a linha e executam cada comando**. Enquanto os **Compiladores** traduzem todo o programa escrito para formato no qual o computador entenda!

Principais Interpretadores Python:



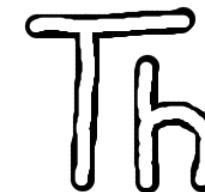
ANACONDA.



IDLE



repl.it



ONLINEGDB.COM

Algumas definições básicas importantes

A photograph of a woman with dark hair, smiling, holding a marker and writing on a whiteboard. In the foreground, there are several yellow sticky notes pinned to the board with red pushpins. One note defines 'Linguagem de Programação' as 'Instruções lógicas e símbolos escritas em um código fonte que os computadores entendam'.

Linguagem de Programação
Instruções lógicas e símbolos
escritas em um código fonte que os
computadores entendam

Lógica de programação
é a organização coesa de uma
sequência de instruções para a
resolução de problemas, ou
criação de software ou app.

Representações de algoritmo:

- Descrição Narrativa;
Descrição em linguagem natural.
- Fluxograma;
- Representação gráfica usando símbolos
geométricos.
- Pseudocódigo
Descrição em port. estruturado

Algumas definições básicas importantes

Saber como fazer comentários na programação é muito importante!

Ao usar **#** na linguagem Python você fará um comentário; isto permite ao programador **documentar a funcionalidade(s)** de cada linha **ou inserir observações importantes ao longo do código.**

Símbolo

#Comentário sem função de execução

Talvez a aplicação mais importante de saber comentar um programa é que **quando você**

desejar fazer alguma modificação no seu código mas não tem certeza de que aquilo de fato

funcionará, o que se faz é comentar o código escrito até o momento e fazer a modificação

numa outra linha de código; isto permitirá você voltar atrás caso sua modificação não

funcione!

Estrutura Básica do nosso programa

Programar é como Cozinhar!

Por isso é preciso ter em mãos:

❖ Os ingredientes certos

✓ As ferramentas certas

✓ E seguir o Passo-a-Passo!

A estrutura básica de um Programa inicia com a **importação da(s) Bibliotecas**

A Palavra **import** informa em seguida o nome da biblioteca que desejamos importar!

Ou podemos usar a palavra **from**, seguida do Nome da Biblioteca, especificando qual **função importar**

```
1 import streamlit as st
2 from ACTlib import *
3 from PIL import Image
4
5 def main():
6
7     Nessa área fica todo o código do
8     programa, junto com as funções
9     declaradas ou oriundas das
10    bibliotecas importadas.
11
12    if __name__ == '__main__':
13        main()
```

as permite especificar um **apelido** para a biblioteca importada. Isto simplifica a escrita do programa!

Carregaremos o código no <https://streamlit.io>. Nosso código estará no [GitHub](#).





05 PASSOS

Para programar
Front-end online

PASSO 01: Crie seu próprio Git Hub

A screenshot of a web browser showing the GitHub sign-up page at github.com/signup?source=login. The page features a large, bold headline: "É muito simples criar um Git Hub!" with a subtitle "Crie o seu agora!". Below this is a "Welcome to GitHub!" message and a "Enter your email" input field. A red arrow points from the text "Clique!" to the "Sign up" button in the top right corner. In the bottom right corner, there is a Bitly QR code.

Join GitHub · GitHub

github.com/signup?source=login

Already have an account? Sign in Sign up

É muito simples criar um Git Hub!

Crie o seu agora!

Welcome to GitHub!
Let's begin the adventure

Enter your email

Continue

Clique!

<https://bit.ly/CrieSeuGitHub>

bitly

PASSO 02: Faça FORK neste Repositório

https://github.com/drive-actsp/ACTlib01

1º) Clique em FORK

Basta clicar em **FORK** e clonar este **repositório** para que você tenha uma cópia semelhante e funcional no seu próprio drive Git Hub. Portanto, se serve de dica: **CRIE SEU PRÓPRIO GIT HUB ANTES DE PROSSEGUIR COM ESTE TUTORIAL!**

3:00

12

Passo 03: Editar o código

massakiigarashi2 / webapp00 (Public)
forked from massakiigarashi2/webapp00

Pin Watch 0 Fork 19 Star 0

Code Pull requests Actions Projects Wiki Security Insights Settings

main webapp00 / webapp00.py < Jump to Go to file ...

massakiigarashi2 Add files via upload Latest commit 6f4a7b2 on 8 Aug History

1 contributor

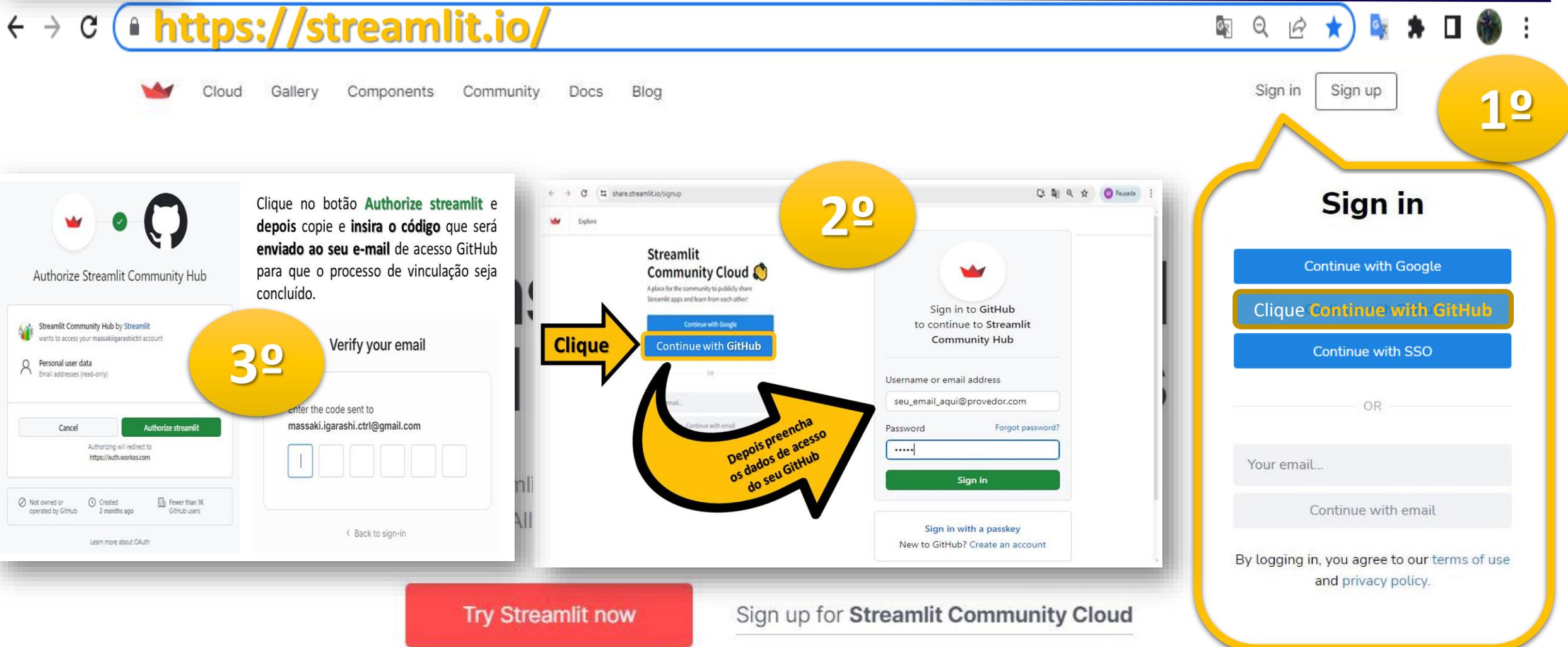
16 lines (11 sloc) | 539 Bytes

Raw Blame Edit this file

01 # myFirstStreamlitApp.py
02 #import the library
03 import streamlit as st
04
05 # Use st.title("") para adicionar um TÍTULO ao seu Web app
06 st.title("MEU 1º WEB APP STREAMLIT")
07
08 # Use st.header("") para adicionar um CABEÇALHO ao seu Web app
09 st.header("Cabeçalho")
10
11 # Use st.subheader("") para adicionar um SUB CABEÇALHO ao seu Web app
12 st.subheader("Sub Cabeçalho")
13
14 # Use st.write("") para adicionar um texto ao seu Web app
15 st.write("Como já deve ter percebido, o método st.write() é usado para escrita de texto e informações gerais!")



Passo 04: Entrando no Streamlit.io



The screenshot shows the Streamlit.io login interface. A yellow arrow points from the Streamlit Community Hub authorization screen to the Streamlit Cloud sign-in screen. A large yellow circle labeled '1º' highlights the 'Sign in' button. A yellow arrow labeled 'Clique' points to the 'Continue with GitHub' button. A yellow speech bubble labeled 'Depois preencha os dados de acesso do seu GitHub' points to the GitHub sign-in fields. A yellow circle labeled '2º' highlights the GitHub sign-in screen. A yellow circle labeled '3º' highlights the Streamlit Community Hub authorization screen.

1º

Sign in

Sign up

2º

Clique

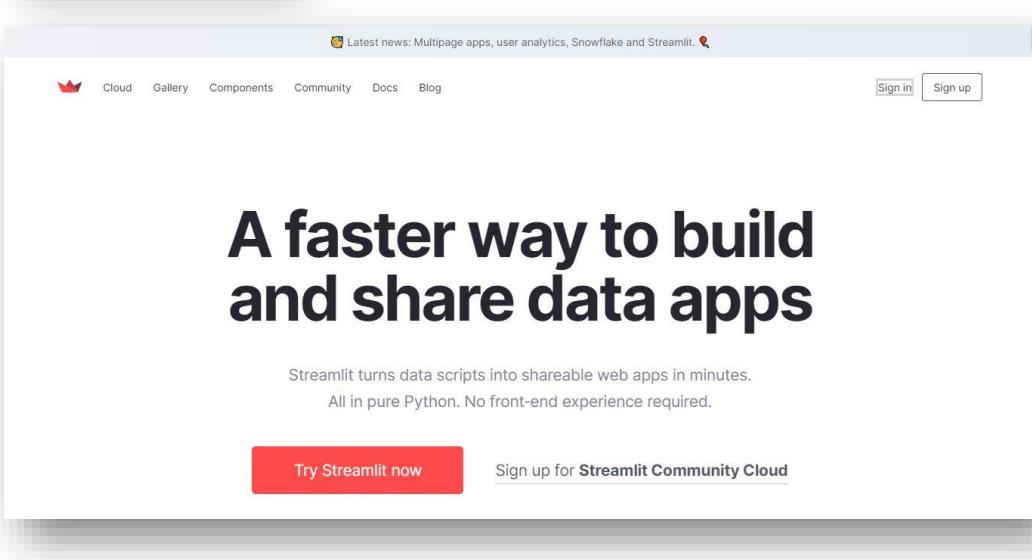
Depois preencha os dados de acesso do seu GitHub

3º

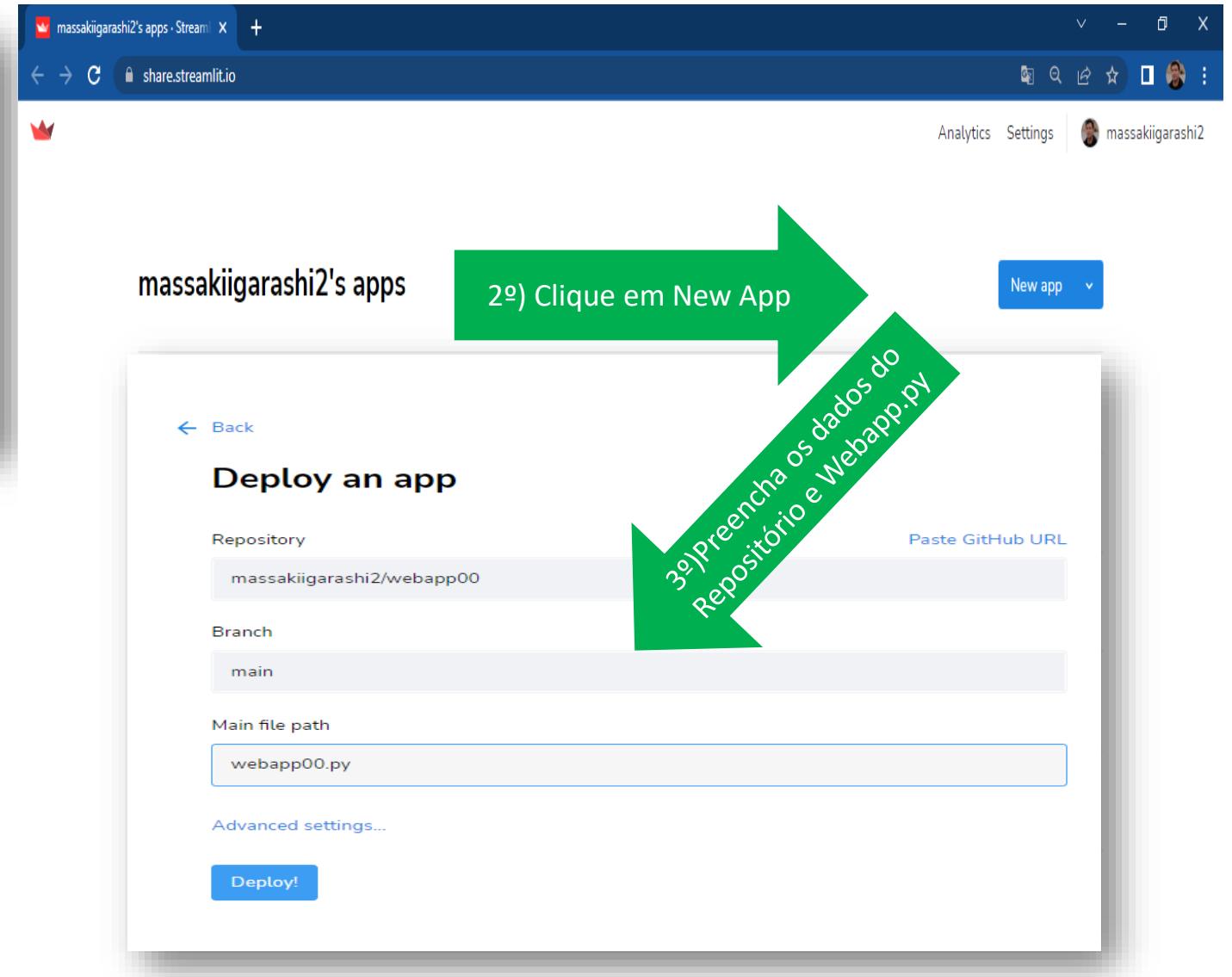
Try Streamlit now

Sign up for Streamlit Community Cloud

Passo 05: Deploy na App

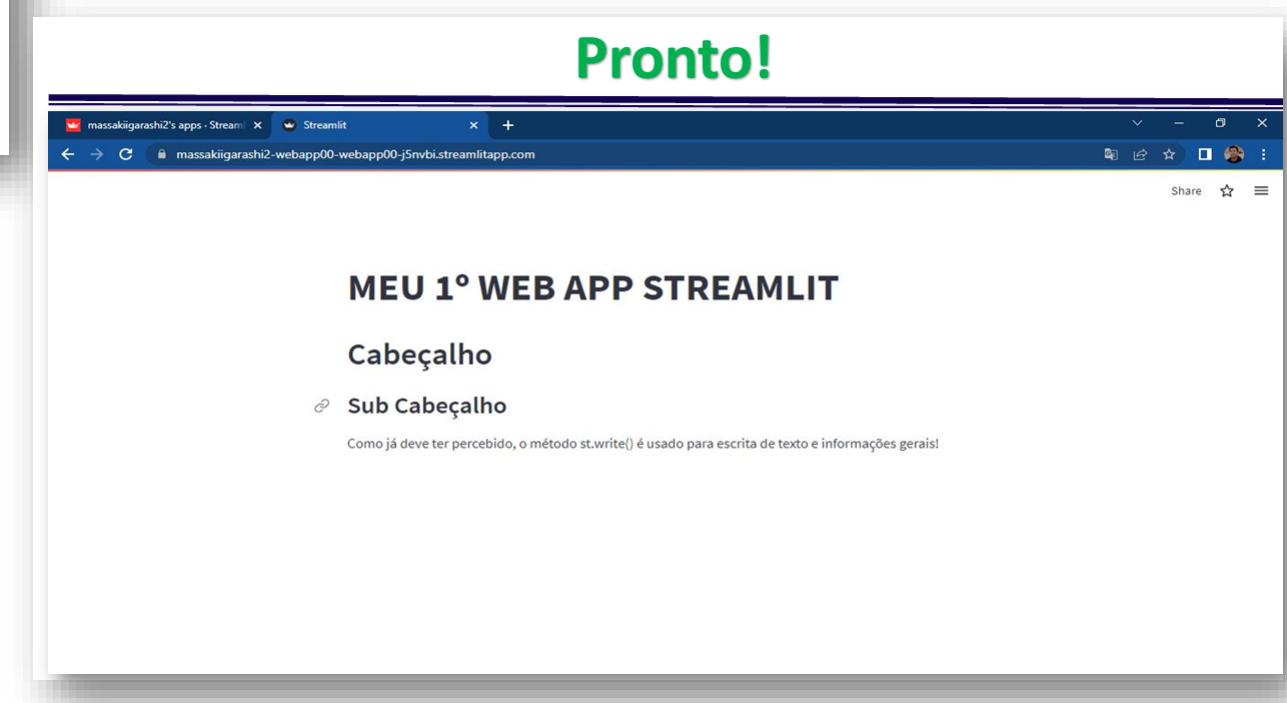


1º) Se for sua primeira vez com o framework Streamlit você precisará fazer, rapidamente, um cadastro e **informar os dados de acesso do seu Git Hub para que a plataforma Streamlit possa se comunicar com seu github e fabricar seu Web App integrando-o à ele.**



The image shows the Streamlit deployment interface. It displays a list of apps under "massakiigarashi2's apps" and a "Deploy an app" form. A green arrow points to the "New app" button with the text "2º) Clique em New App". Another green arrow points to the "Repository" field containing "massakiigarashi2/webapp00" with the text "3º) Preencha os dados do Repositório e Webapp.py".

Fabricando seu Web App



A screenshot of a Streamlit application window titled "massakiigarashi2's apps - Streamlit". The URL in the address bar is "massakiigarashi2-webapp00-webapp00-j5nvbi.streamlitapp.com". The main content area displays the text "Pronto!" in green, followed by "MEU 1º WEB APP STREAMLIT" in bold black text. Below this, there are two sections: "Cabeçalho" and "Sub Cabeçalho". A small note at the bottom states: "Como já deve ter percebido, o método st.write() é usado para escrita de texto e informações gerais!". The bottom right corner shows the number "16".



PASSOS
Para programar
Front-end OFF-LINE

Passo 01: Instale ANACONDA no Desktop

The screenshot shows a video player interface with a black background. At the top, there's a navigation bar for 'vimeo' with links like 'Gerenciar Videos', 'Recursos', 'Funções', 'Assistir', and 'Upgrade'. A search bar says 'Pesquisar vídeos, pe' with a magnifying glass icon. On the right, there are icons for a bell, a user profile (M), and a blue button labeled 'Novo vídeo'.

The main content area displays the 'Anaconda Distribution' page from anaconda.com. It features a QR code on the left. The text 'Individual Edition is now ANACONDA DISTRIBUTION' is displayed above a large green header. Below it, the text 'The world's most popular open-source Python distribution platform' is shown. A video player at the bottom left shows a video titled 'Open Source Python Distribution' with a duration of 03:39. To the right, there's a call-to-action box for 'Anaconda Distribution' with a 'Download' button and links for 'For Windows' (Python 3.9 • 64-Bit Graphical Installer • 594 MB) and 'Get Additional Installers' (links for Windows, Mac, and Linux).

1º Aponte seu celular para ter acesso ao vídeo de instrução p/ instalação da Distribuição ANACONDA PYTHON

<https://vimeo.com/724995785>

Passo 02: Instale Notepad++ no Desktop

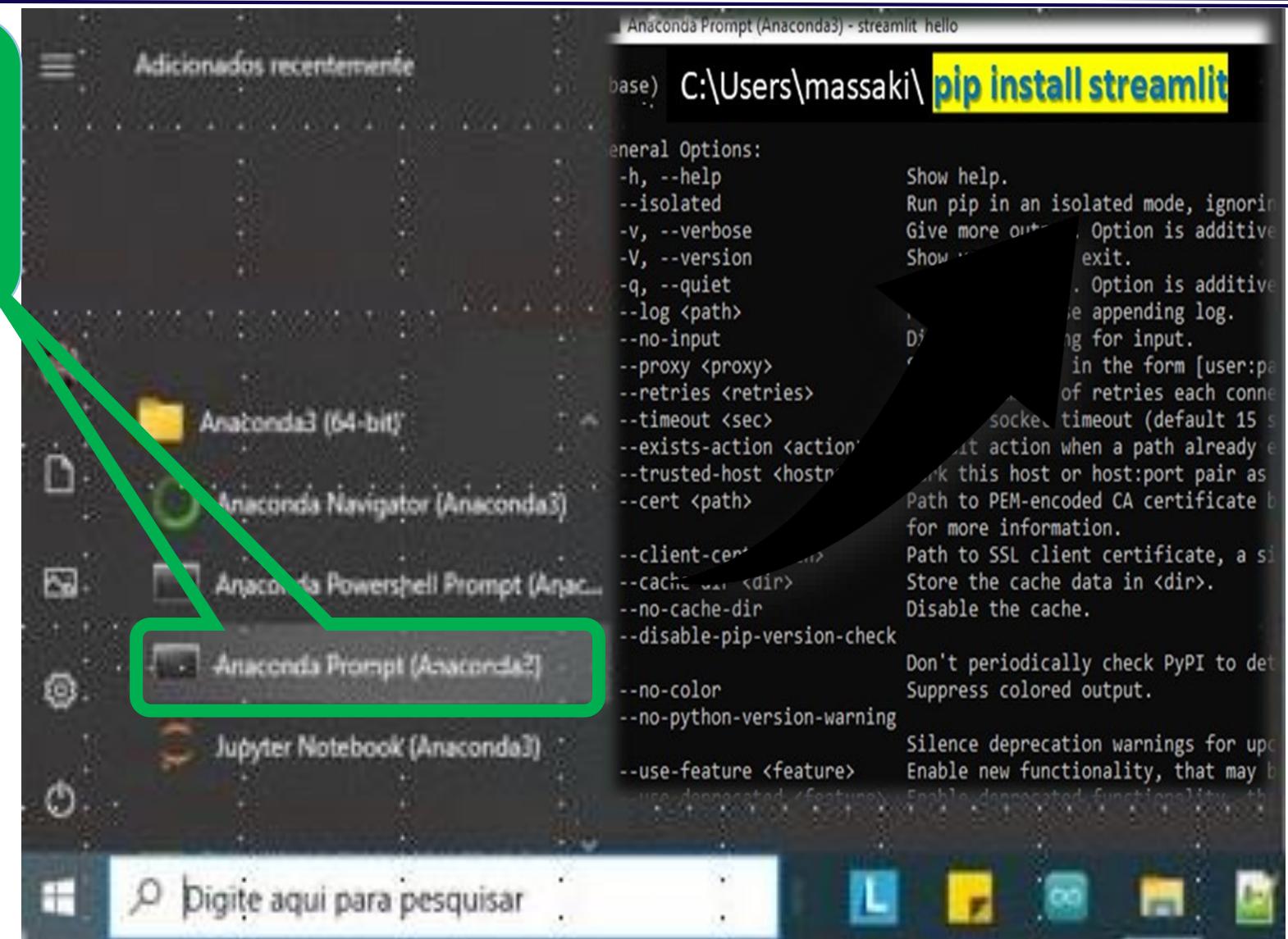
Passo 03: Instale o Framework Streamlit



2º) Após instalar o pcte Anaconda

Vá em Anaconda Prompt

e digite: **pip install streamlit**

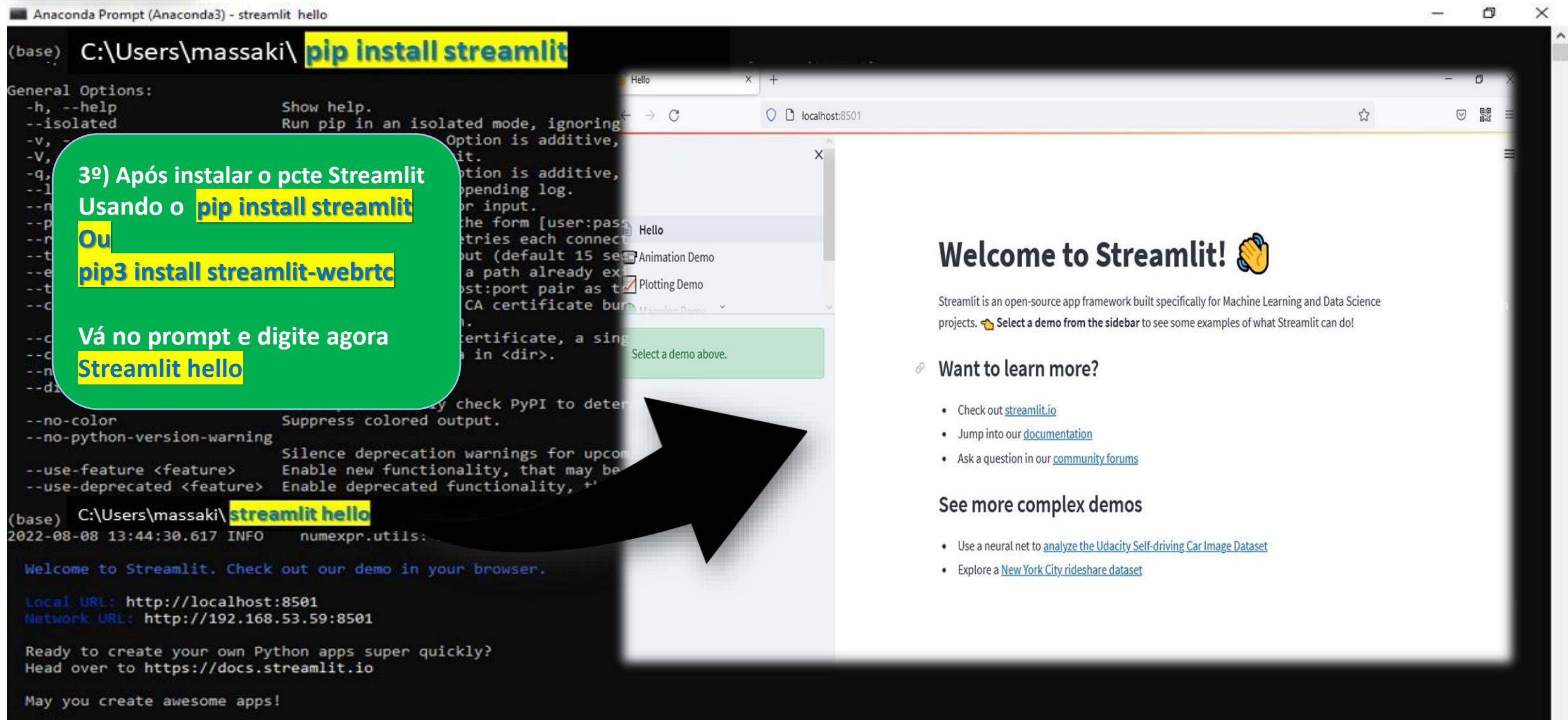


```
Anaconda Prompt (Anaconda3) - streamlit hello
base) C:\Users\massaki\ pip install streamlit

General Options:
-h, --help
--isolated
-v, --verbose
-V, --version
-q, --quiet
--log <path>
--no-input
--proxy <proxy>
--retries <retries>
--timeout <sec>
--exists-action <action>
--trusted-host <host>
--cert <path>
--client-cert <path>
--cache-dir <dir>
--no-cache-dir
--disable-pip-version-check
--no-color
--no-python-version-warning
--use-feature <feature>
--use-deprecated <feature>
--enable-deprecated-functionality <feature>
```

Digite aqui para pesquisar

Passo 04: Teste o App padrão Hello



The screenshot shows a Windows desktop with two windows open. On the left is the Anaconda Prompt window titled 'Anaconda Prompt (Anaconda3) - streamlit hello'. The command entered is '(base) C:\Users\massaki\ pip install streamlit'. A green callout box highlights this command with the text: '3º) Após instalar o pkte Streamlit Usando o pip install streamlit Ou pip3 install streamlit-webrtc'. Another green callout box highlights the text 'Vá no prompt e digite agora Streamlit hello' below it. On the right is a Streamlit application window titled 'Hello' with the URL 'localhost:8501'. The Streamlit interface displays a 'Welcome to Streamlit!' message with a hand icon, followed by instructions: 'Streamlit is an open-source app framework built specifically for Machine Learning and Data Science projects. Select a demo from the sidebar to see some examples of what Streamlit can do!'. It also has sections for 'Want to learn more?' with links to the official website, documentation, and forums, and 'See more complex demos' with links to neural net and NYC rideshare datasets.

General Options:

- h, --help Show help.
- isolated Run pip in an isolated mode, ignoring system site packages.
- v, --version Print the version number.
- q, --quiet Option is additive, so quiet mode is additive.
- log-level Log level (default: info).
- nolog Don't log anything.
- path Path to the directory containing the requirements file.
- prefix Prefix to add to every package name.
- no-deps Don't check PyPI to determine dependencies.
- no-color Suppress colored output.
- no-pyversion-warning Silence deprecation warnings for upcoming Python versions.
- use-feature <feature> Enable new functionality, that may be experimental.
- use-deprecated <feature> Enable deprecated functionality, that may be removed.

(base) C:\Users\massaki\ **streamlit hello**

2022-08-08 13:44:30.617 INFO numexpr.utils: Welcome to Streamlit. Check out our demo in your browser.

Local URL: <http://localhost:8501>
Network URL: <http://192.168.53.59:8501>

Ready to create your own Python apps super quickly?
Head over to <https://docs.streamlit.io>

May you create awesome apps!

Welcome to Streamlit! 

Streamlit is an open-source app framework built specifically for Machine Learning and Data Science projects.  Select a demo from the sidebar to see some examples of what Streamlit can do!

Want to learn more?

- Check out [streamlit.io](#)
- Jump into our [documentation](#)
- Ask a question in our [community forums](#)

See more complex demos

- Use a neural net to [analyze the Udacity Self-driving Car Image Dataset](#)
- Explore a [New York City rideshare dataset](#)

PASSO 05: Crie uma pasta no seu Desktop c/ os arquivos

The screenshot shows a GitHub repository named 'ACTlib01'. The repository is public and contains several files and folders: 'imgs', 'pages', '1stDashboard.png', 'Earth_Clock.jpg', 'Escolha_Qualidade.jpg', 'Exemplo1.py', 'Procfile.txt', and 'README.md'. A green arrow labeled '2º) Download ZIP' points to the 'Download ZIP' button in the bottom right corner of the repository page. Another green arrow labeled '1º Clique Code' points to the 'Code' dropdown menu at the top right of the page. The URL in the browser's address bar is <https://github.com/drive-actsp/ACTlib01>.



<https://bit.ly/ACTlib01>

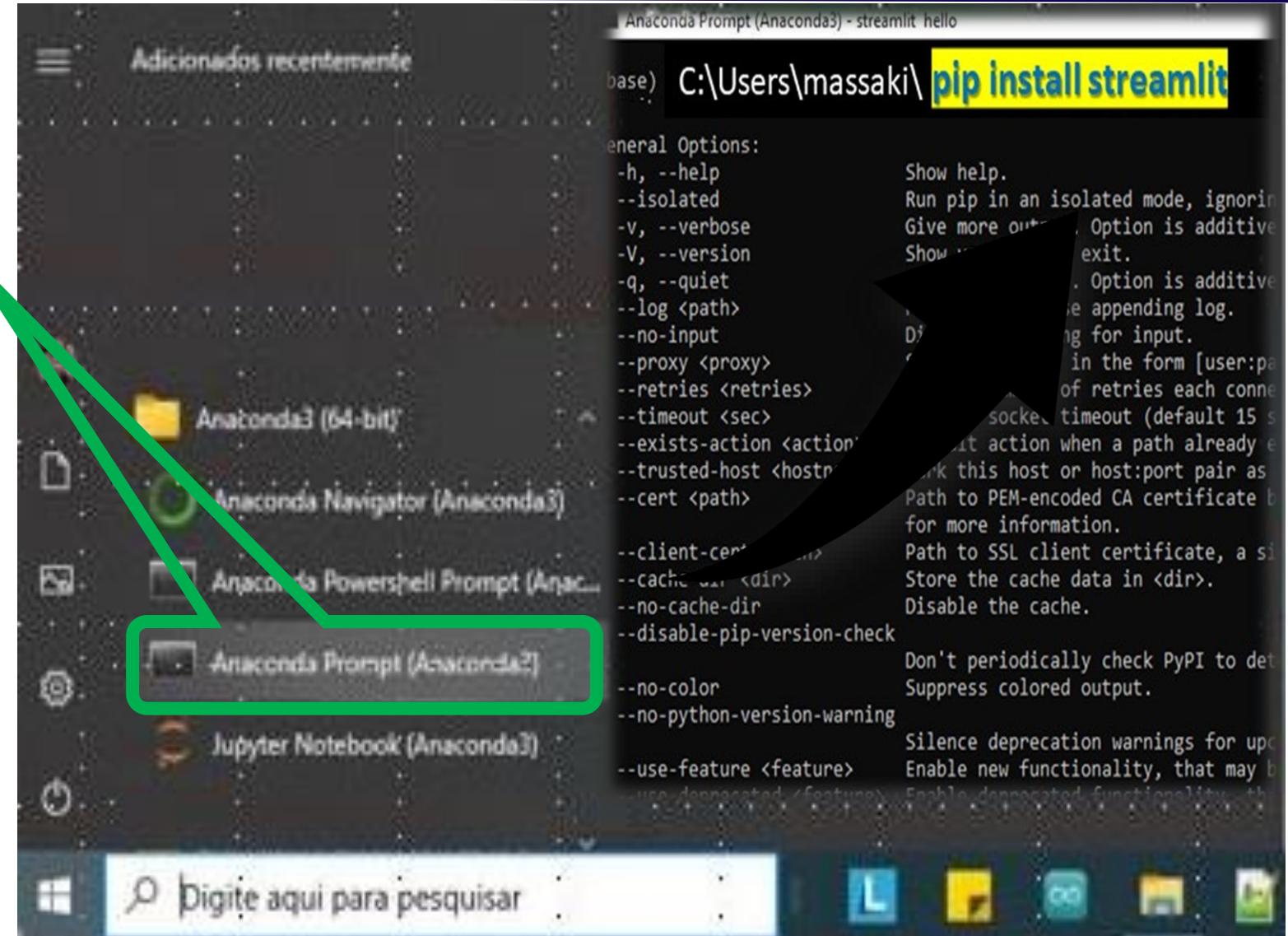
No packages published

PASSO 06: Instale as bibliotecas necessárias

pip3 install -r requirements.txt

ou

python -m pip install -r requirements.txt

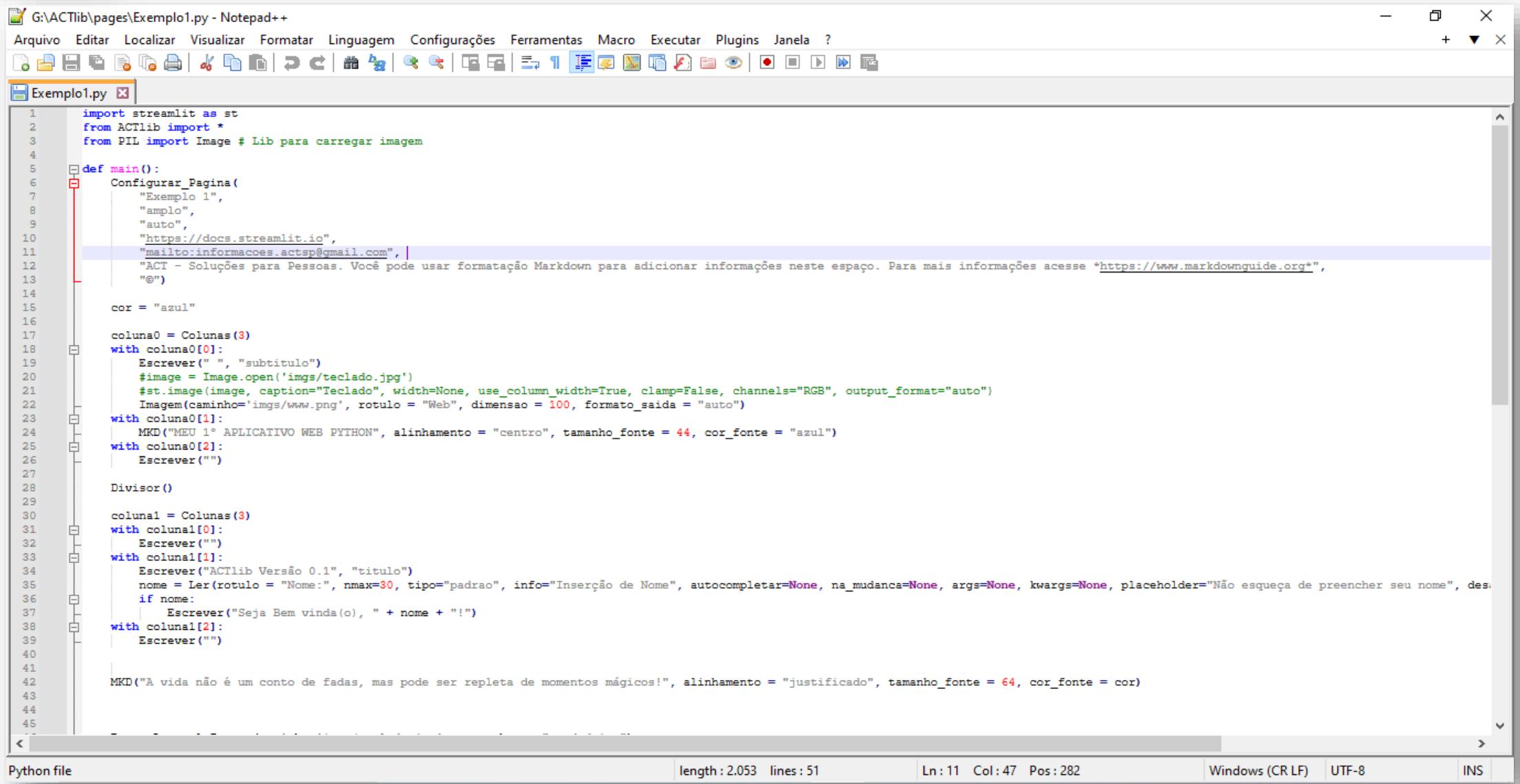


```
Anaconda Prompt (Anaconda3) - streamlit hello
base) C:\Users\massaki\ pip install streamlit

General Options:
-h, --help
--isolated
-v, --verbose
-V, --version
-q, --quiet
--log <path>
--no-input
--proxy <proxy>
--retries <retries>
--timeout <sec>
--exists-action <action>
--trusted-host <host>
--cert <path>
--client-cert <path>
--cache-dir <dir>
--no-cache-dir
--disable-pip-version-check
--no-color
--no-python-version-warning
--use-feature <feature>
--use-deprecated <feature>
--enable-deprecated-functionality <feature>

Show help.
Run pip in an isolated mode, ignoring
Give more output. Option is additive.
Show version information.
Option is additive.
Show quiet mode.
Append log file.
D...
Show help.
Run pip in an isolated mode, ignorin
Give more output. Option is additive
Show version information.
Option is additive.
Append log file.
D...
in the form [user:pa
of retries each conn
socket timeout (default 15 s
t action when a path already e
mark this host or host:port pair as
Path to PEM-encoded CA certificate b
for more information.
Path to SSL client certificate, a si
Store the cache data in <dir>.
Disable the cache.
Don't periodically check PyPI to det
Suppress colored output.
Silence deprecation warnings for upc
Enable new functionality, that may b
Enable deprecated functionality, th
```

Passo 07: Editar o código usando Notepad++



The screenshot shows the Notepad++ interface with the file `G:\ACTlib\pages\Exemplo1.py` open. The code is a Streamlit application named `Exemplo 1`. It includes imports for `streamlit`, `ACTlib`, and `PIL`. The `main()` function configures the page, adds a header with a logo, and displays a text block with a link to a Markdown guide. The code then creates two columns: one with a title, a logo placeholder, and a welcome message; and another with a title, a name input field, and a message. Finally, it adds a footer message.

```
1 import streamlit as st
2 from ACTlib import *
3 from PIL import Image # Lib para carregar imagem
4
5 def main():
6     Configurar_Pagina(
7         "Exemplo 1",
8         "amplo",
9         "auto",
10        "https://docs.streamlit.io",
11        "mailto:informacoes.actsp@gmail.com", |
12        "ACT - Soluções para Pessoas. Você pode usar formatação Markdown para adicionar informações neste espaço. Para mais informações acesse https://www.markdownguide.org",
13        "@")
14
15     cor = "azul"
16
17     coluna0 = Colunas(3)
18     with coluna0[0]:
19         Escrever(" ", "subtitulo")
20         #image = Image.open('imgs/teclado.jpg')
21         #st.image(image, caption="Teclado", width=None, use_column_width=True, clamp=False, channels="RGB", output_format="auto")
22         Imagem(caminho='imgs/www.png', rotulo = "Web", dimensao = 100, formato_saida = "auto")
23     with coluna0[1]:
24         MKD("MEU 1º APPLICATIVO WEB PYTHON", alinhamento = "centro", tamanho_fonte = 44, cor_fonte = "azul")
25     with coluna0[2]:
26         Escrever("")
27
28     Divisor()
29
30     coluna1 = Colunas(3)
31     with coluna1[0]:
32         Escrever("")
33     with coluna1[1]:
34         Escrever("ACTlib Versão 0.1", "titulo")
35         nome = Ler(rotulo = "Nome:", nmax=30, tipo="padrao", info="Inserção de Nome", autocompletar=None, na_mudanca=None, args=None, kwargs=None, placeholder="Não esqueça de preencher seu nome", desfazer=None)
36         if nome:
37             Escrever("Seja Bem vinda(o), " + nome + "!")
38     with coluna1[2]:
39         Escrever("")
40
41
42 MKD("A vida não é um conto de fadas, mas pode ser repleta de momentos mágicos!", alinhamento = "justificado", tamanho_fonte = 64, cor_fonte = cor)
```

Python file length : 2.053 lines : 51 Ln : 11 Col : 47 Pos : 282 Windows (CR LF) UTF-8 INS

Passo 08: Deploy na App no Desktop

```
Anaconda Prompt (Anaconda3) - streamlit run webapp00.py

(base) G:\MCK2.1 - LP22.2\Streamlit.io>streamlit run webapp00.py
2022-08-08 15:19:58.731 INFO    numexpr.utils: NumExpr defaulting to 4 threads.

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.53.59:8501
```

1º) Entre na pasta onde se encontra seu arquivo usando o comando CD do Prompt e em seguida digite **streamlit run Webapp00.py**



MEU 1º WEB APP STREAMLIT

Cabeçalho

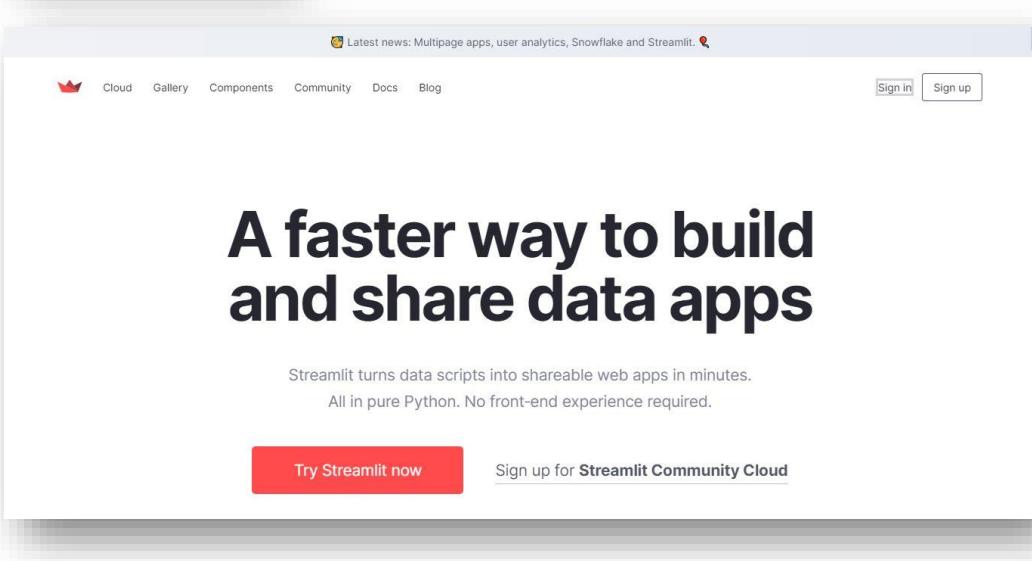
Sub Cabeçalho

Como já deve ter percebido, o método st.write() é usado para escrita de texto e informações gerais!

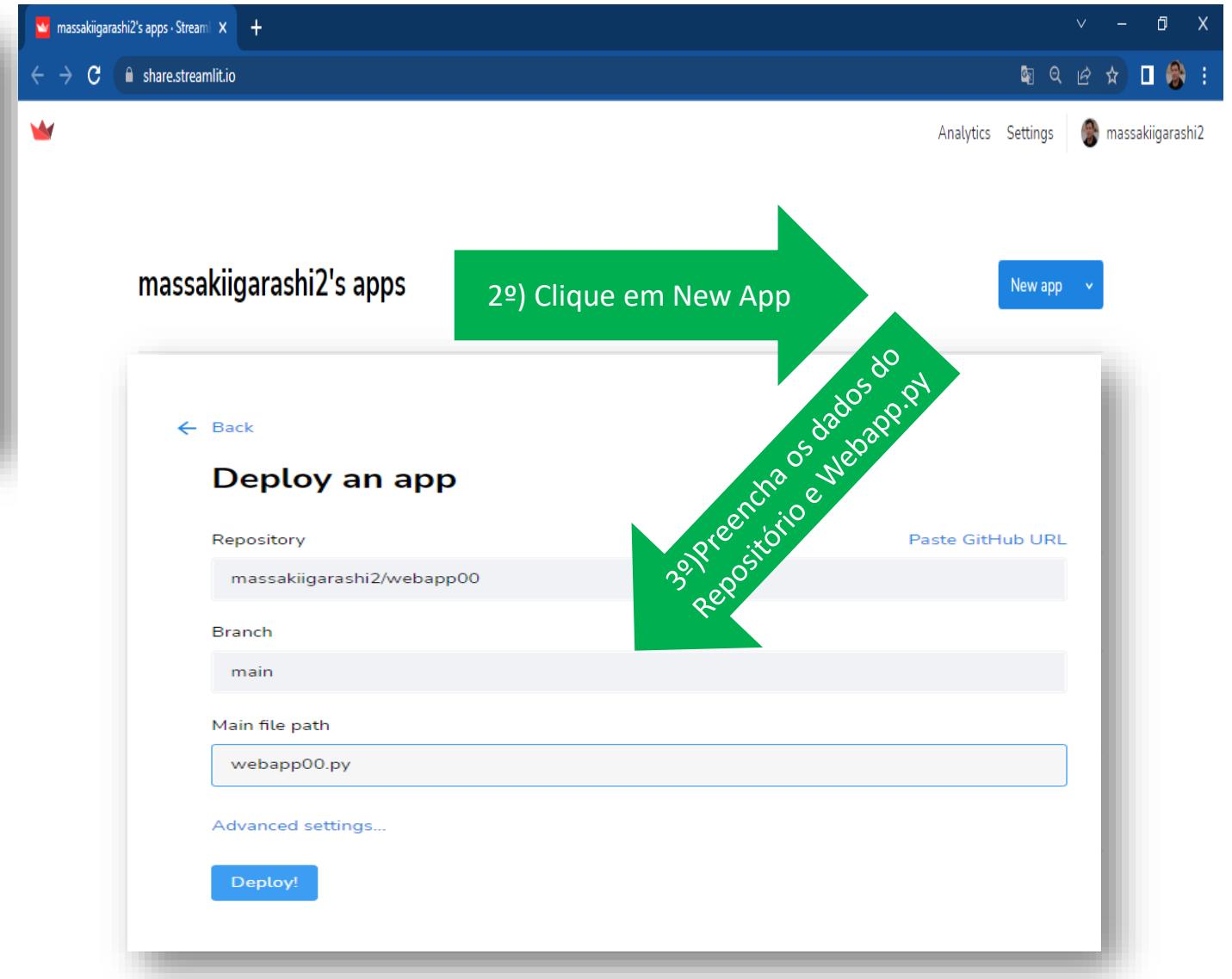
Made with Streamlit

Passo 09: Copie conteúdo do Desktop p/ GitHub

Passo 10: Deploy na App



1º) Se for sua primeira vez com o framework Streamlit você precisará fazer, rapidamente, um cadastro e **informar os dados de acesso do seu Git Hub para que a plataforma Streamlit possa se comunicar com seu github e fabricar seu Web App integrando-o à ele.**



The screenshot shows the Streamlit deployment interface. It displays the user's apps with the title "massakiigarashi2's apps". A green arrow points to the "New app" button with the text "2º) Clique em New App". Another green arrow points to the GitHub repository input field with the text "3º) Preencha os dados do Repositório e Webapp.py". The deployment form includes fields for Repository (massakiigarashi2/webapp00), Branch (main), Main file path (webapp00.py), and a Deploy! button.

Fabricando seu Web App

Após Clicar no botão Deploy
é só aguardar!



Your app is in the oven



Pronto!



MEU 1º WEB APP STREAMLIT

Cabeçalho

Sub Cabeçalho

Como já deve ter percebido, o método st.write() é usado para escrita de texto e informações gerais!

Saiba os Operadores Aritméticos e Relacionais

Operadores aritméticos: são símbolos especiais que realizam operações matemáticas básicas

Operadores Aritméticos

OPERAÇÃO	SÍMBOLO	CÓDIGO EM PYTHON	CÓDIGO EM C++	DESCRIÇÃO
ADIÇÃO	+	$x + y$	$x + y$	Efetua a soma $x + y$
SUBTRAÇÃO	-	$x - y$	$x - y$	Efetua a subtração $x - y$
MULTIPLICAÇÃO	*	$x * y$	$x * y$	Efetua a multiplicação $x * y$
DIVISÃO	/	x / y	x / y	Efetua a divisão x / y
DIVISÃO INTEIRA	//	$x // y$	Operador inexistente	Efetua a divisão truncada $x // y$ Ou seja, $5 // 2 = 2$, pois trunca o ponto flutuante 0.5
POTÊNCIA	**	$x ** y$	Operador inexistente	Eleva um número x a um expoente y (x^y)
RESTO DA DIVISÃO	%	$x \% y$	$x \% y$	Obtém o resto da divisão entre dois inteiros



Operadores Relacionais

OPERAÇÃO	SÍMBOLO	CÓDIGO EM PYTHON	CÓDIGO EM C++	DESCRIÇÃO
Menor que	<	$x < y$	$x < y$	Verdadeiro se x Menor y
Maior que	>	$x > y$	$x > y$	Verdadeiro se x Maior y
Igual	==	$x == y$	$x == y$	Verdadeiro se x Igual a y
ATRIBUIÇÃO	=	$X = 2$	$X = 2$	Atribui um valor ou resultado de cálculo à uma constante
Diferente	!=	$x != y$	$x != y$	Verdadeiro se x Diferente de y
Menor igual	<=	$x <= y$	$x <= y$	Verdadeiro se x Menor ou Igual a y
Maior igual	>=	$x >= y$	$x >= y$	Verdadeiro se x Maior ou Igual a y

O símbolo **=** é considerado operador de atribuição; usado para atribuir valores para constantes OU atribuir cálculos

Exemplo:
 $X = 2$
 $Y = 3$
 $soma = X + Y$

