

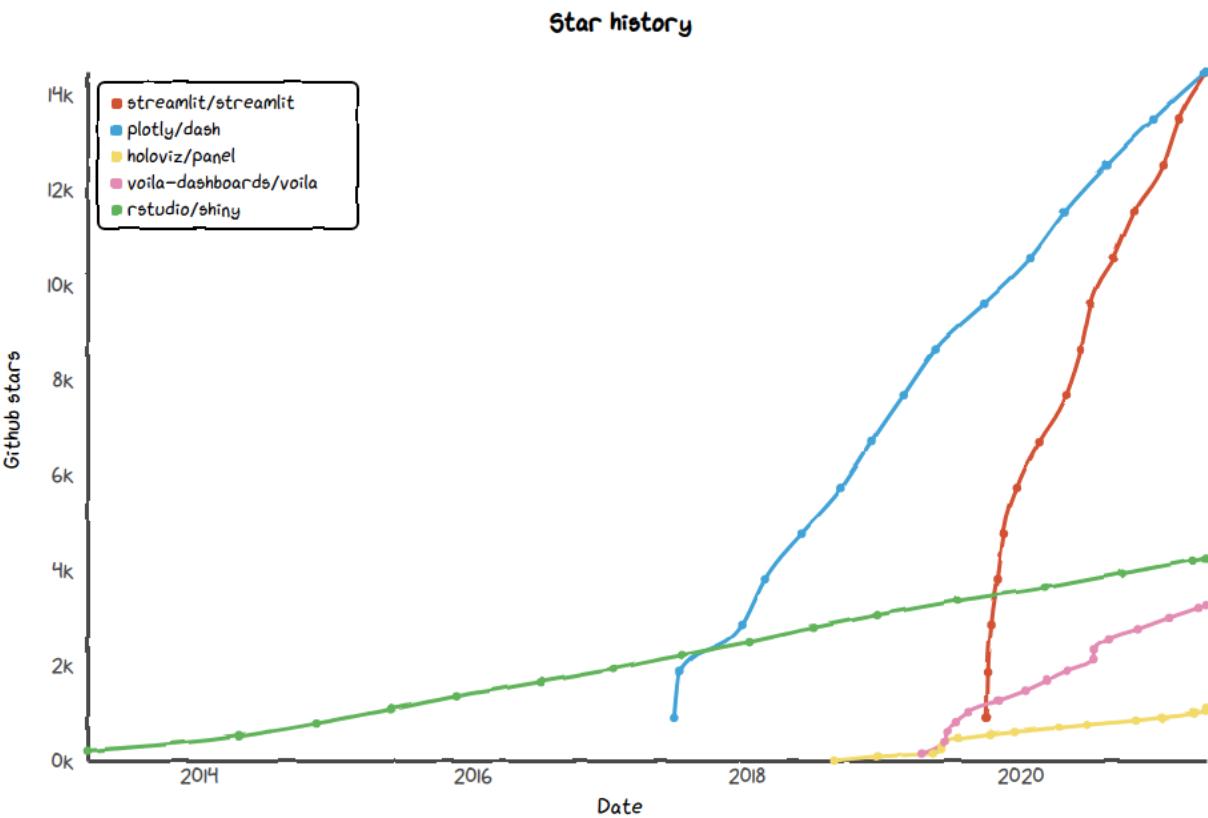
# Streamlit

Créer des applications web interactives  
avec simplicité

# C'est quoi Streamlit ?

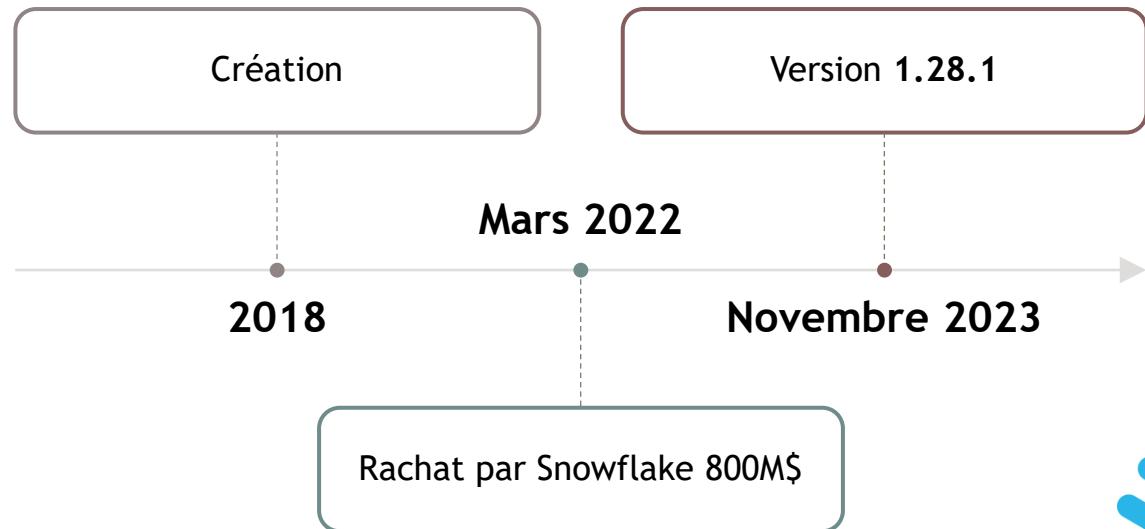
- ▶ Framework Open Source permettant de créer des applications web
- ▶ Facile à apprendre et utiliser
- ▶ 100% Python
- ▶ Pas d'expérience Front-end requise
- ▶ Une équipe de développement active: Une nouvelle version tous les mois
- ▶ Une communauté importante: ~250 modules tiers

# Popularité croissante



- ▶ 8 millions de téléchargements
- ▶ 1,5 millions d'applications créées
- ▶ 150 000 utilisateurs actifs tous les mois
- ▶ 28 600 étoiles sur Github

# Quelques dates



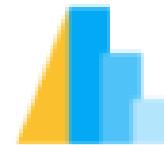
# Streamlit



# Dataviz

- ▶ Streamlit est compatible avec de nombreuses librairies Python de data visualisation

**matplotlib**



Vega-Altair



**bokeh**



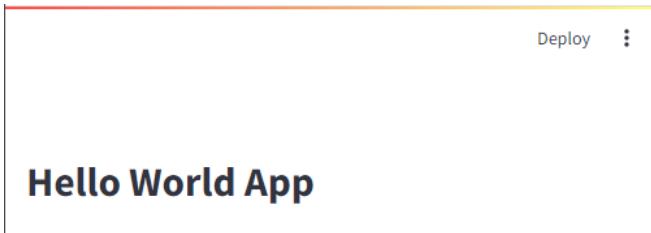
 **plotly**

 **seaborn**

# Hello world

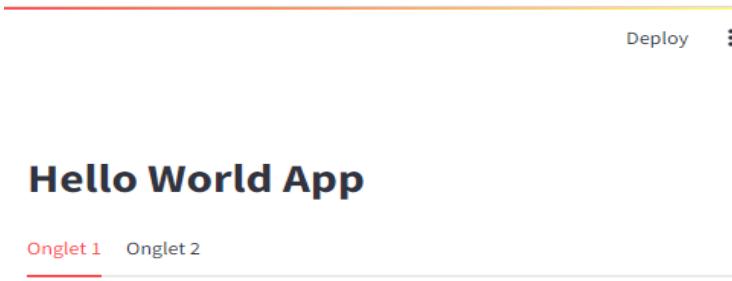
- ▶ J'ai dans mon appli ... 1 titre

```
import streamlit as st  
  
st.title('Hello World App')
```



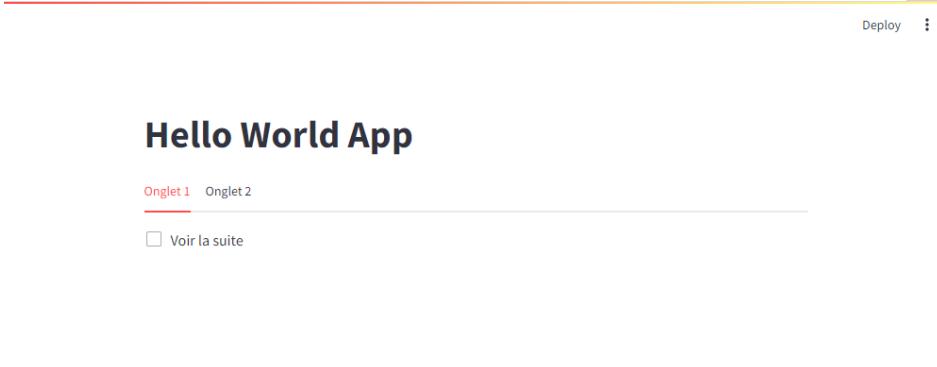
- ▶ J'ai dans mon appli ... 1 titre, 2 onglets

```
import streamlit as st  
  
st.title('Hello World App')  
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
```



- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox

```
import streamlit as st  
  
st.title('Hello World App')  
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])  
with tab1:  
    display_tab_content = st.checkbox('Voir la suite', value=False)
```

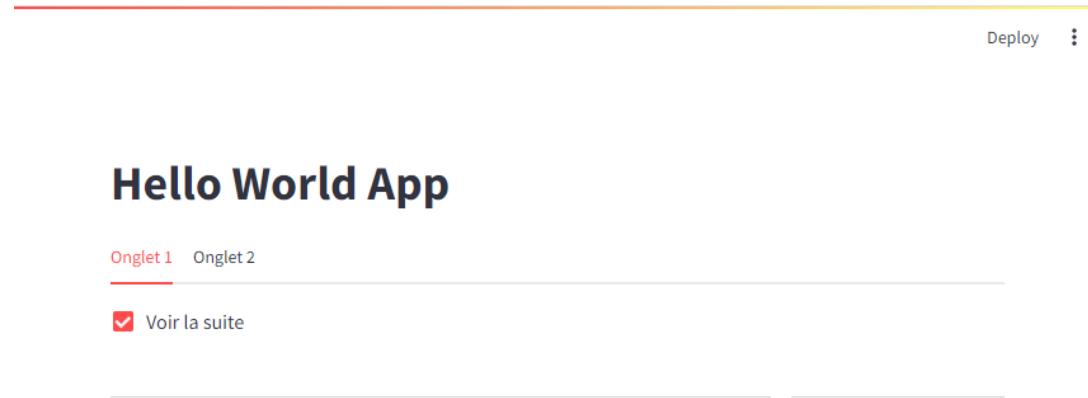


# Hello world

- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox, 2 colonnes

```
import streamlit as st

st.title('Hello World App')
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
with tab1:
    display_tab_content = st.checkbox('Voir la suite', value=False)
    if display_tab_content:
        large_column, small_column = st.columns([3, 1])
        with large_column:
            st.divider()
        with small_column:
            "___"
```

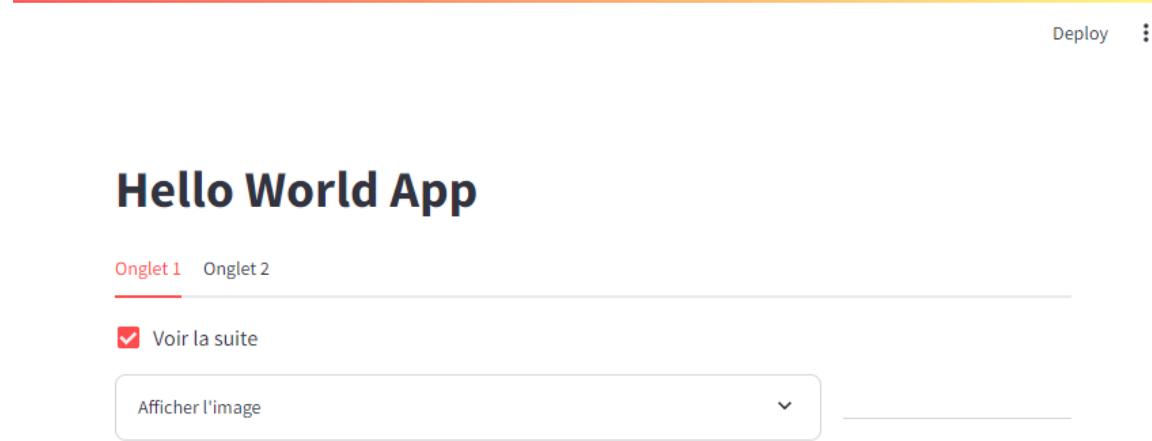


# Hello world

- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox, 2 colonnes, 1 expander

```
import streamlit as st

st.title('Hello World App')
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
with tab1:
    display_tab_content = st.checkbox('Voir la suite', value=False)
    if display_tab_content:
        large_column, small_column = st.columns([3, 1])
        with large_column:
            with st.expander("Afficher l'image", expanded=False):
                "___"
        with small_column:
            "___"
```

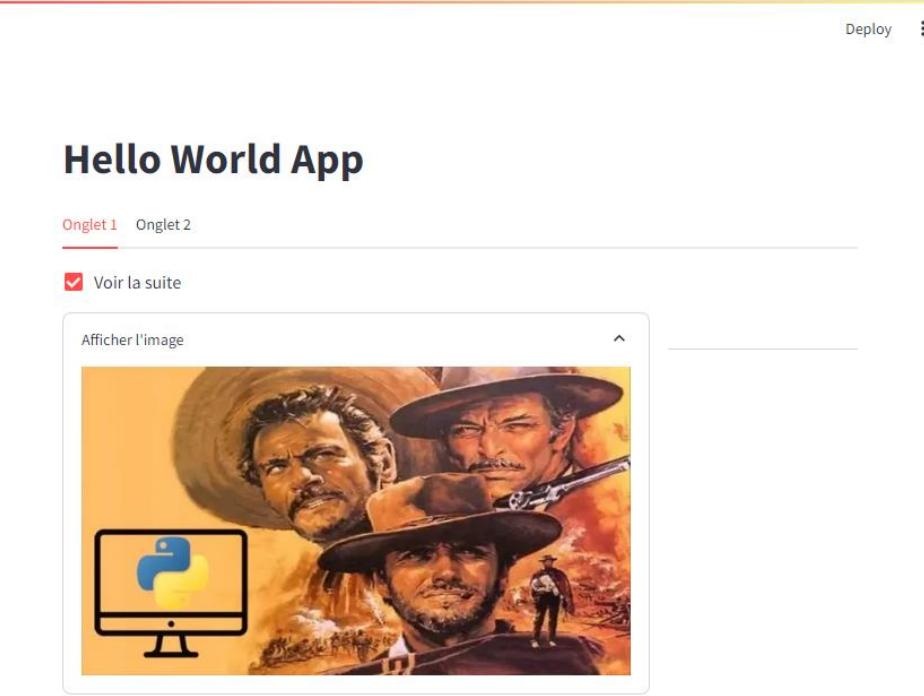


# Hello world

- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox, 2 colonnes, 1 expander, 1 image

```
import streamlit as st

st.title('Hello World App')
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
with tab1:
    display_tab_content = st.checkbox('Voir la suite', value=False)
    if display_tab_content:
        large_column, small_column = st.columns([3, 1])
        with large_column:
            with st.expander("Afficher l'image", expanded=False):
                st.image('meetup.png')
        with small_column:
            "___"
```

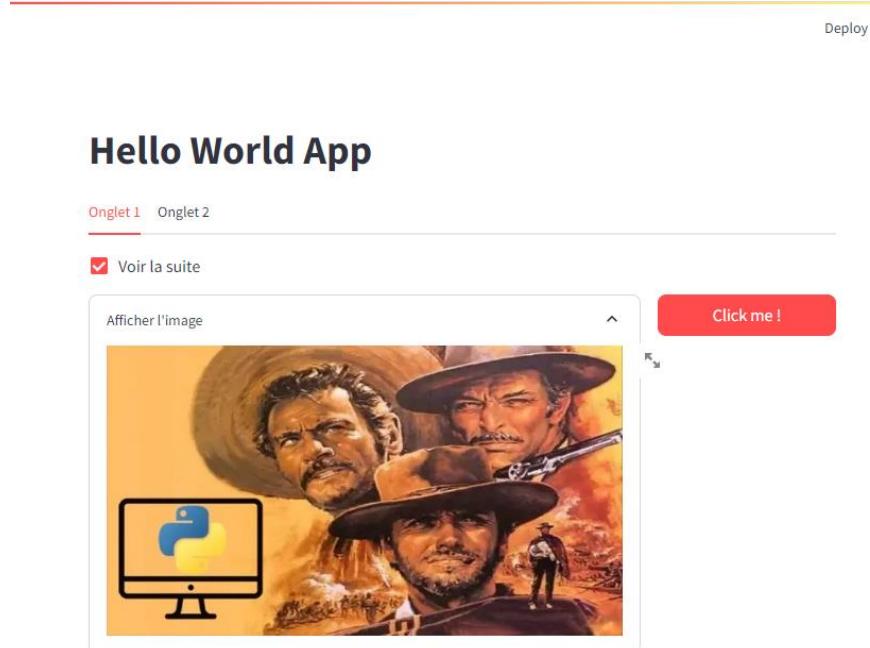


# Hello world

- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox, 2 colonnes, 1 expander, 1 image, 1 bouton

```
import streamlit as st

st.title('Hello World App')
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
with tab1:
    display_tab_content = st.checkbox('Voir la suite', value=False)
    if display_tab_content:
        large_column, small_column = st.columns([3, 1])
        with large_column:
            with st.expander("Afficher l'image", expanded=False):
                st.image('meetup.png')
        with small_column:
            if st.button('Click me !', type='primary', use_container_width=True):
                '___'
```

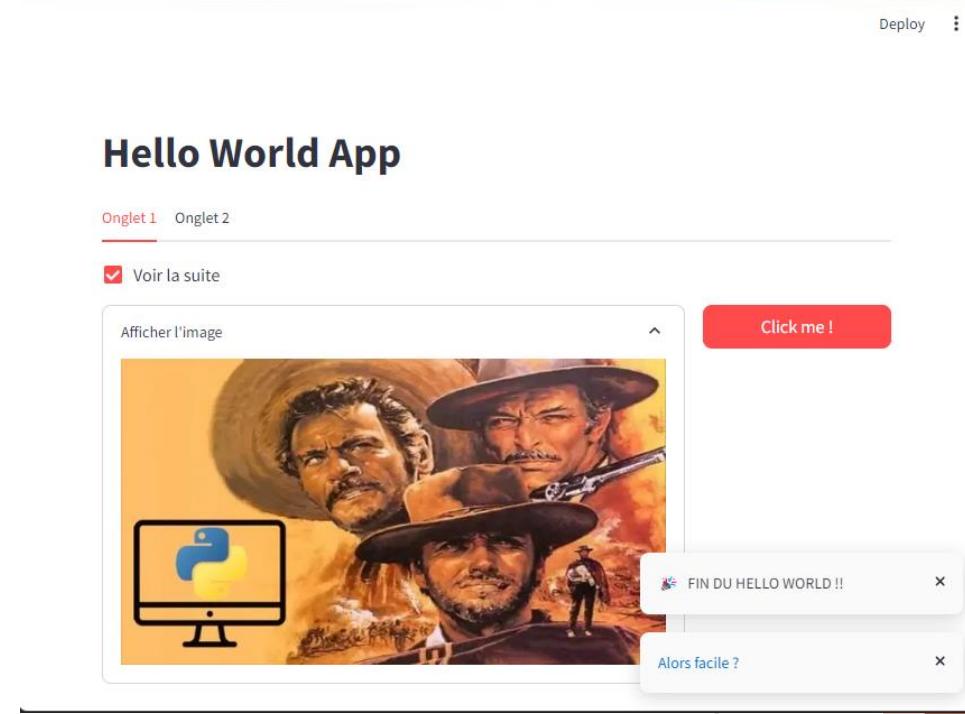


# Hello world

- ▶ J'ai dans mon appli ... 1 titre, 2 onglets, 1 checkbox, 2 colonnes, 1 expander, 1 image, 1 bouton, 2 toasts ...

```
import streamlit as st

st.title('Hello World App')
tab1, tab2 = st.tabs(['Onglet 1', 'Onglet 2'])
with tab1:
    display_tab_content = st.checkbox('Voir la suite', value=False)
    if display_tab_content:
        large_column, small_column = st.columns([3, 1])
        with large_column:
            with st.expander("Afficher l'image", expanded=False):
                st.image('meetup.png')
        with small_column:
            if st.button('Click me !', type='primary', use_container_width=True):
                st.toast("FIN DU HELLO WORLD !!", icon='🎉')
                st.toast(":blue[Alors facile ?]")
```



# Les principaux groupes de composants

- ▶ Text elements
- ▶ Data elements
- ▶ Chart éléments
- ▶ Input widgets
- ▶ Media elements
- ▶ Layout and containers
- ▶ Chat elements
- ▶ Status elements

# Démo

```
streamlit run hello_world.py
```

```
streamlit run widgets.py
```

```
streamlit hello
```

# Quelques composants notables

- ▶ st.set\_page\_config
- ▶ st.sidebar
- ▶ st.session\_state
- ▶ st.cache
- ▶ st.connection

# st.set\_page\_config

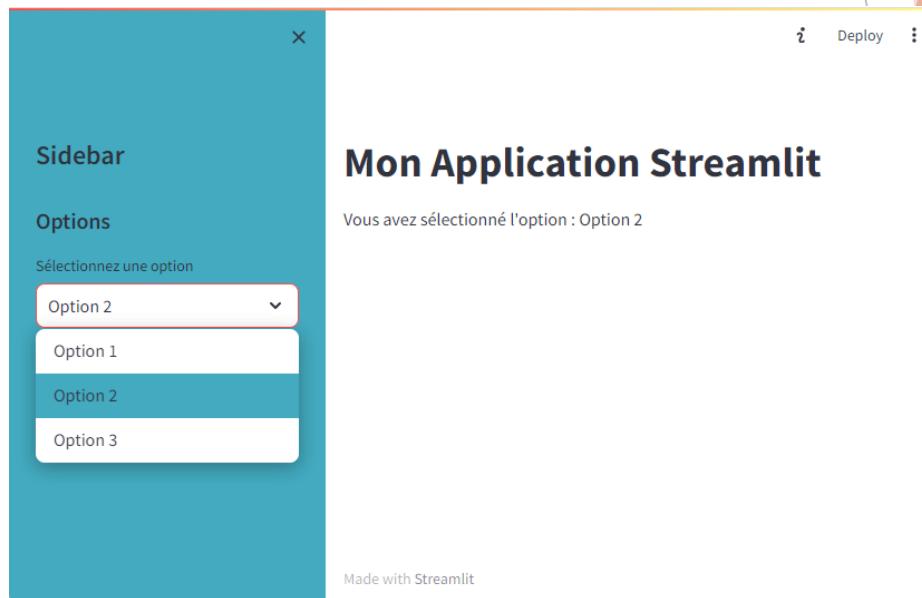
- ▶ Permet de configurer l'application
  - ▶ Titre de la page
  - ▶ Icône de la page (favicon)
  - ▶ Taille de l'écran (centré ou large)
  - ▶ Lien des menus (Help, Report a bug, About)
  - ▶ Affichage ou non de la sidebar
- ▶ Cette commande doit être la première commande Streamlit utilisée sur une page d'application, et ne doit être défini qu'une seule fois par page.
- ▶ Exemple:

```
st.set_page_config(  
    page_title="Ex-stream-ly Cool App",  
    page_icon="💡",  
    layout="wide",  
    initial_sidebar_state="expanded",  
    menu_items={  
        'Get Help': 'https://www.extremelycoolapp.com/help',  
        'Report a bug': "https://www.extremelycoolapp.com/bug",  
        'About': "# This is a header. This is an *extremely* cool app!"  
    }  
)
```

# st.sidebar

- ▶ Barre latérale
- ▶ Peut servir de menu
- ▶ Peut contenir n'importe quel composant
- ▶ Peut être masquer ou afficher
- ▶ Redimensionnable

```
with st.sidebar:  
    st.title('Sidebar')  
    st.header("Options")  
    option = st.selectbox("Sélectionnez une option", ["Option 1", "Option 2", "Option 3"])  
  
    st.title("Mon Application Streamlit")  
    st.write(f"You avez sélectionné l'option : {option}")
```



# st.session\_state

- ▶ A chaque interaction entre l'utilisateur et un composant Streamlit, l'ensemble du code est réexécuté entièrement.
- ▶ `st.session_state` permet d'enregistrer et stocker des valeurs entre les réexécutions
- ▶ S'utilise comme un dictionnaire (clé - valeur)
- ▶ Exemple: Compteur de click

```
# Initialisation
if 'compteur' not in st.session_state:
    st.session_state['compteur'] = 0

if st.button('Click here'):
    st.session_state['compteur'] += 1
st.text(f"Nombre de click effectués = {st.session_state['compteur']}")
```

# st.cache

- ▶ S'utilise comme un décorateur python (`@st.cache`)
- ▶ Permet de mettre le résultat de fonctions Python dans le cache
- ▶ Si lors d'une exécution, le programme rencontre une fonction qu'il a déjà exécutée **avec les mêmes paramètres** alors la fonction n'est pas réexécutée. Streamlit va chercher les résultats qui sont enregistrés dans le cache
- ▶ Très pratique pour des actions qui prennent du temps, comme le chargement de données ou l'entraînement d'un modèle de machine learning
- ▶ `st.cache_data` et `st.cache_resource` selon le type d'objet retourné par la fonction.

# st.connection

- ▶ Permet de créer et gérer des connexions à des sources de données ou des API dans une application Streamlit.
- ▶ Connection à la source de données paramétrable via un fichier de configuration ou directement en paramètre de la fonction
- ▶ Retourne un objet de connection du type spécifié

# Modules tiers

- ▶ Développé par la communauté
- ▶ **st-aggrid:** Tableau de données personnalisés
- ▶ **streamlit-keycloak:** Authentification keycloak
- ▶ **st-folium:** Cartographie avancée
- ▶ Et beaucoup d'autres...
- ▶ Component gallery:  
<https://streamlit.io/components>

## CATEGORIES

All

LLMs

Widgets

Charts

Authentication

Connections

Images & video

Text

Maps

Dataframes

Graphs

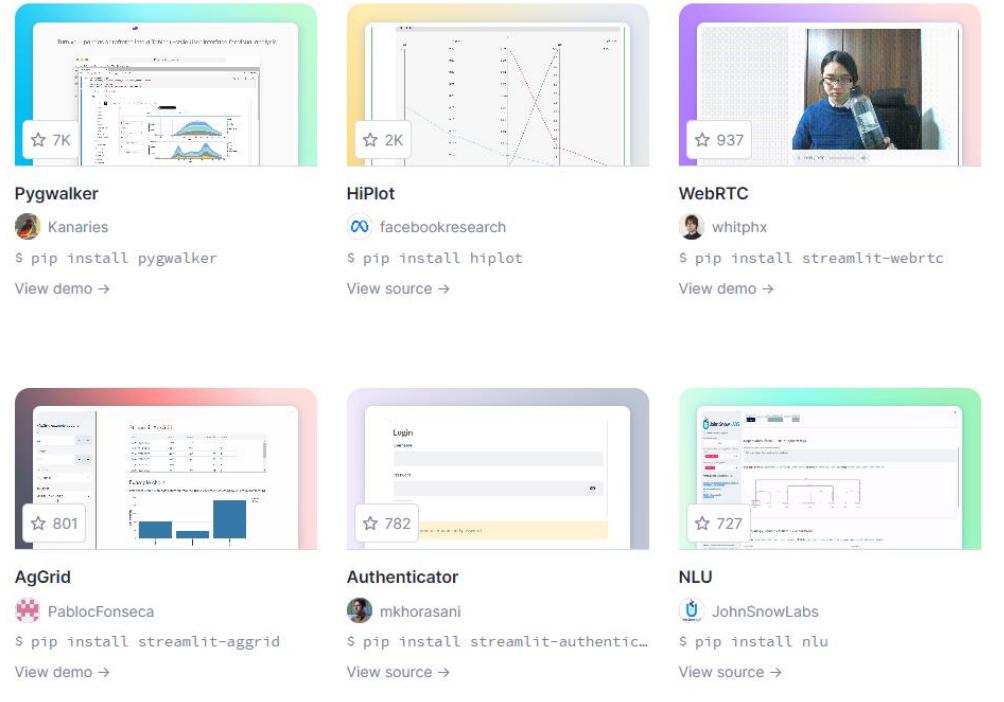
Molecules & genes

Code editors

Page navigation

Developer tools

Integrations



The image shows a grid of nine cards, each representing a different third-party module for Streamlit. The cards are arranged in three rows of three. Each card includes a preview image, the module name, the developer's profile picture and name, the number of stars, and links to view the demo and source code.

Module	Developer	Stars	Action Links
Pygwalker	Kanaries	7K	\$ pip install pygwalker View demo →
HiPlot	facebookresearch	2K	\$ pip install hiplot View source →
WebRTC	whitphx	937	\$ pip install streamlit-webrtc View demo →
AgGrid	PabloFonseca	801	\$ pip install streamlit-aggrid View demo →
Authenticator	mkhorasani	782	\$ pip install streamlit-authentic... View source →
NLU	JohnSnowLabs	727	\$ pip install nlu View source →

# Déployer son application

- ▶ Il est possible de rendre publique gratuitement son application en la déployant « Streamlit Community Cloud »
1. Créer son workspace <https://share.streamlit.io>
  2. Pousser son code sur Github
  3. Déployer son app

Deploy an app

Repository  Paste GitHub URL

Branch

Main file path

Custom subdomain (optional)  .streamlit.app

Domain is available

[Advanced settings...](#)

[Deploy!](#)

# Pour aller plus loin



Site officiel de Streamlit

- ▶ App Gallery: <https://streamlit.io/gallery>
- ▶ Blog: <https://blog.streamlit.io/>
- ▶ Forum: <https://discuss.streamlit.io/>



Chaine Youtube de Fanilo Andrianasolo:

<https://www.youtube.com/@andfanilo>



**Fanilo Andrianasolo**

@andfanilo · 6,24 k abonnés · 55 vidéos

💡 Craft Data Science That Gets you Noticed >

[andfanilo.com](#) et 3 autres liens

Abonné ▾



Mon LinkedIn:



**Bruno Quéguiner**

Consultant Data chez Seenovate

Bretagne, France · [Coordonnées](#)

231 relations

Merci de votre attention !  
Des questions ?

