



Introduction à Streamlit

Apprendre à créer et déployer une application Data sur Internet

École du Digital - Bordeaux



Plan de cours

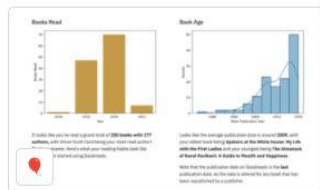
- Qu'est ce que [Streamlit](#) ?
- **Installer et démarrage** de Streamlit
- Les éléments d'interface
- Les éléments d'interaction
- L'intégration entre Streamlit et Plotly
- Déployer son application en ligne

Qu'est-ce que Streamlit ?



[Streamlit.io](#) est un service en ligne permettant de déployer des applications Data sans connaître les technologies web.

[Streamlit](#) est une librairie et une application Python qui va vous aider à construire une application web (que ce soit pour des applications Data ou non).



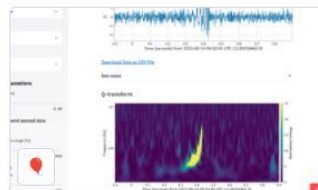
Analyzing Your Goodreads Reading Habits

An app that scrapes (and never keeps or stores) the books you've read and analyzes data with visual graphs. Then

by Tyler Richards

[View source code](#) →

[Go to app](#) →



Gravitational Wave Quickview

This app downloads and displays a few seconds of data from the Gravitational Wave Open Science Center.

by Jonah Kanner

[View source code](#) →

[Go to app](#) →



NYC Uber Ridesharing Data

A Streamlit demo to interactively visualize Uber pickups in New York City.

by Streamlit

[View source code](#) →

[Go to app](#) →

<https://streamlit.io/gallery>



Streamlit Cheat Sheet

A cheat sheet for Streamlit.

by Daniel Lewis

[View source code](#) →



Streamlit Roadmap

Check out current and upcoming features and fixes on our roadmap app.

by Streamlit

[View source code](#) →



Trainergenerator

A web app to generate template code for machine learning.

by Johannes Rieke

[View source code](#) →

Concepts principaux : Application streamlit



Vous exécuterez une application Streamlit en ligne de commandes :

```
streamlit run app.py
```

Cette instruction ne fonctionnera pas (contrairement à un script Python classique) :

```
python app.py
```

Concepts principaux : Flow de travail



Puisque l'objectif est de déployer une application web, vous allez travailler en parallèle dans votre *éditeur de code* (Visual Studio Code par exemple) ET dans un *navigateur web* pour voir le résultat s'afficher en temps réel.

L'idéal est d'avoir 2 écrans à sa disposition 🤖

Concepts principaux : streamlit as st



Toutes les fonctions de Streamlit sont liées au module Python streamlit.

Vous utiliserez un alias à l'import dans vos scripts :

```
import streamlit as st
```


st.write()

```
st.subheader("This is a subhea
```

[illegible]

Concepts principaux : Visualisation de données



Afficher des diagrammes et des cartes :

`st.line_chart()`,

`st.bar_chart()`,

`st.map()`,

...

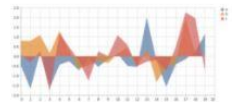
Compatible avec matplotlib, seaborn,
plotly, etc...



Simple line charts

Display a line chart.

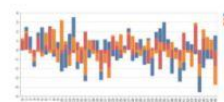
```
st.line_chart(my_data_frame)
```



Simple area charts

Display an area chart.

```
st.area_chart(my_data_frame)
```



Simple bar charts

Display a bar chart.

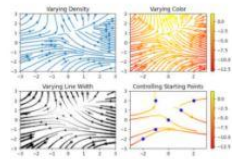
```
st.bar_chart(my_data_frame)
```



Scatterplots on maps

Display a map with points on it.

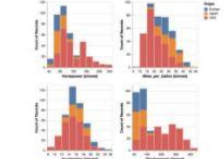
```
st.map(my_data_frame)
```



Matplotlib

Display a matplotlib.pyplot figure.

```
st.pyplot(my_mpl_figure)
```



Altair

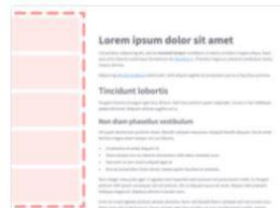
Display a chart using the Altair library.

```
st.altair_chart(my_altair_chart)
```

Concepts principaux : Organisation des contenus

Pensé pour des applications simples, sur une seule page.

1. Un contenu principal, *et un bloc de côté optionnel* (`st.sidebar()`)
2. Des colonnes de proportions configurables
3. Des containers



Sidebar

Display items in a sidebar.

```
st.sidebar.write("This lives i  
st.sidebar.button("Click me!")
```

Columns

Insert containers laid out as side-by-side columns.

```
col1, col2 = st.columns(2)  
col1.write("this is column 1")  
col2.write("this is column 2")
```

Expander

Insert a multi-element container that can be expanded/collapsed.

```
with st.expander("Open to see  
st.write("This is more conte
```

Container

Insert a multi-element container.

```
c = st.container()  
st.write("This will show last"  
c.write("This will show first"  
c.write("This will show second"
```

Empty

Insert a single-element container.

```
c = st.empty()  
st.write("This will show last"  
c.write("This will be replaced"  
c.write("This will show first"
```

Concepts principaux : Widgets



Interagir avec l'application :


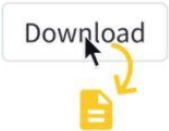




`st.button()`,

`st.checkbox()`,

`st.selectbox()`,

...

Récupérer les valeurs envoyées par les utilisateurs pour adapter le rendu de l'application.

| | | |
|---|---|---|
|  |  |  |
| Button Display a button widget. | Download button Display a download button widget. | Checkbox Display a checkbox widget. |
| <code>clicked = st.button("Click me")</code> | <code>st.download_button("Download", file_data, file_name="downloaded_file.zip")</code> | <code>selected = st.checkbox("I agree with the terms of service")</code> |
|  |  |  |
| Radio Display a radio button widget. | Selectbox Display a select widget. | Multiselect Display a multiselect widget. The multiselect widget starts as empty. |
| <code>choice = st.radio("Pick one", ["Dog", "Cat", "Goldfish"])</code> | <code>choice = st.selectbox("Pick one", ["cats", "dogs", "potatoes"])</code> | <code>choices = st.multiselect("Buy", ["Apples", "potatoes"])</code> |

Concepts principaux : Documentation de l'API




Vous verrez le *minimum vital*, mais il y a beaucoup plus !

La documentation est bien faite, et on trouve de nombreux tutoriels sur Internet.

La base de connaissances est bien organisée.

Soyeux curieux/curieuses 😊 !

● API reference

Write and magic 

Text elements 

Data display elements 

Chart elements 

Input widgets 

Media elements 

Layouts and containers 

Status elements 

Control flow 

Utilities 

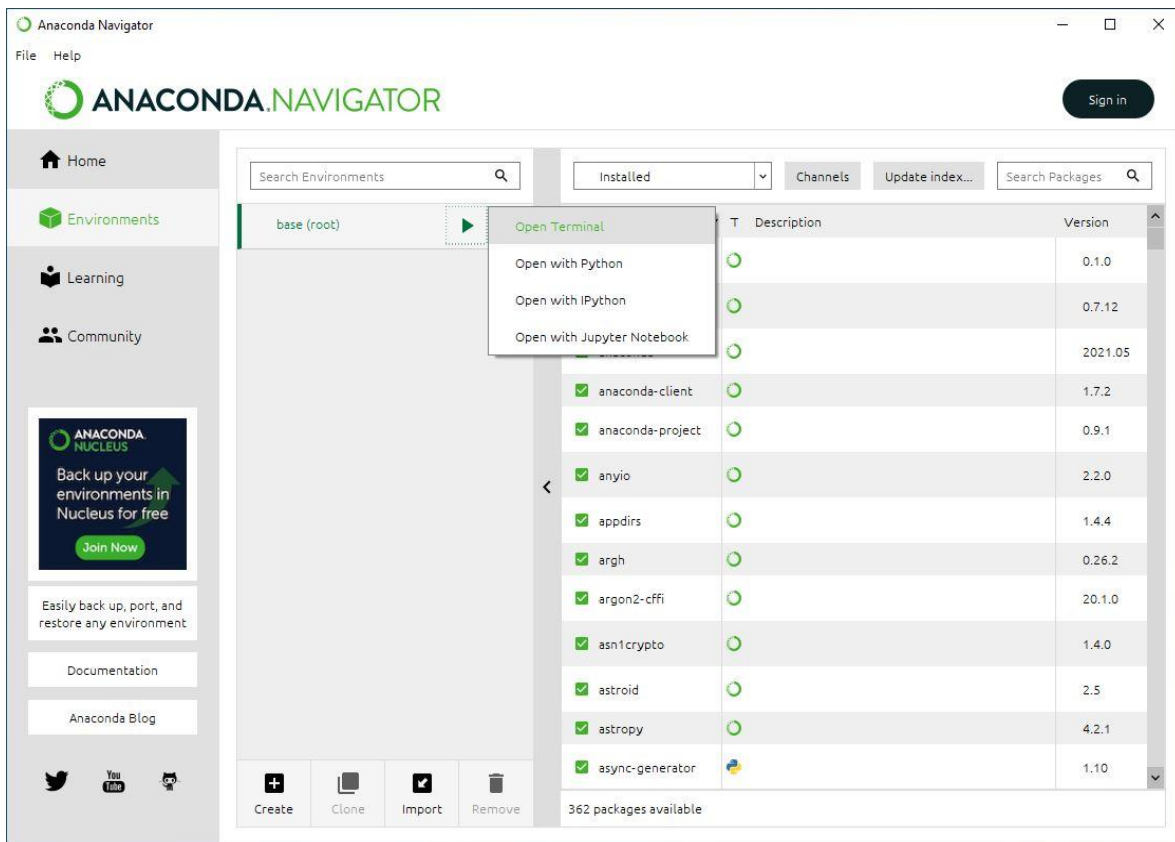
Mutate charts

State management

Performance 

Installation et démarrage de Streamlit

Installation à l'aide d'Anaconda



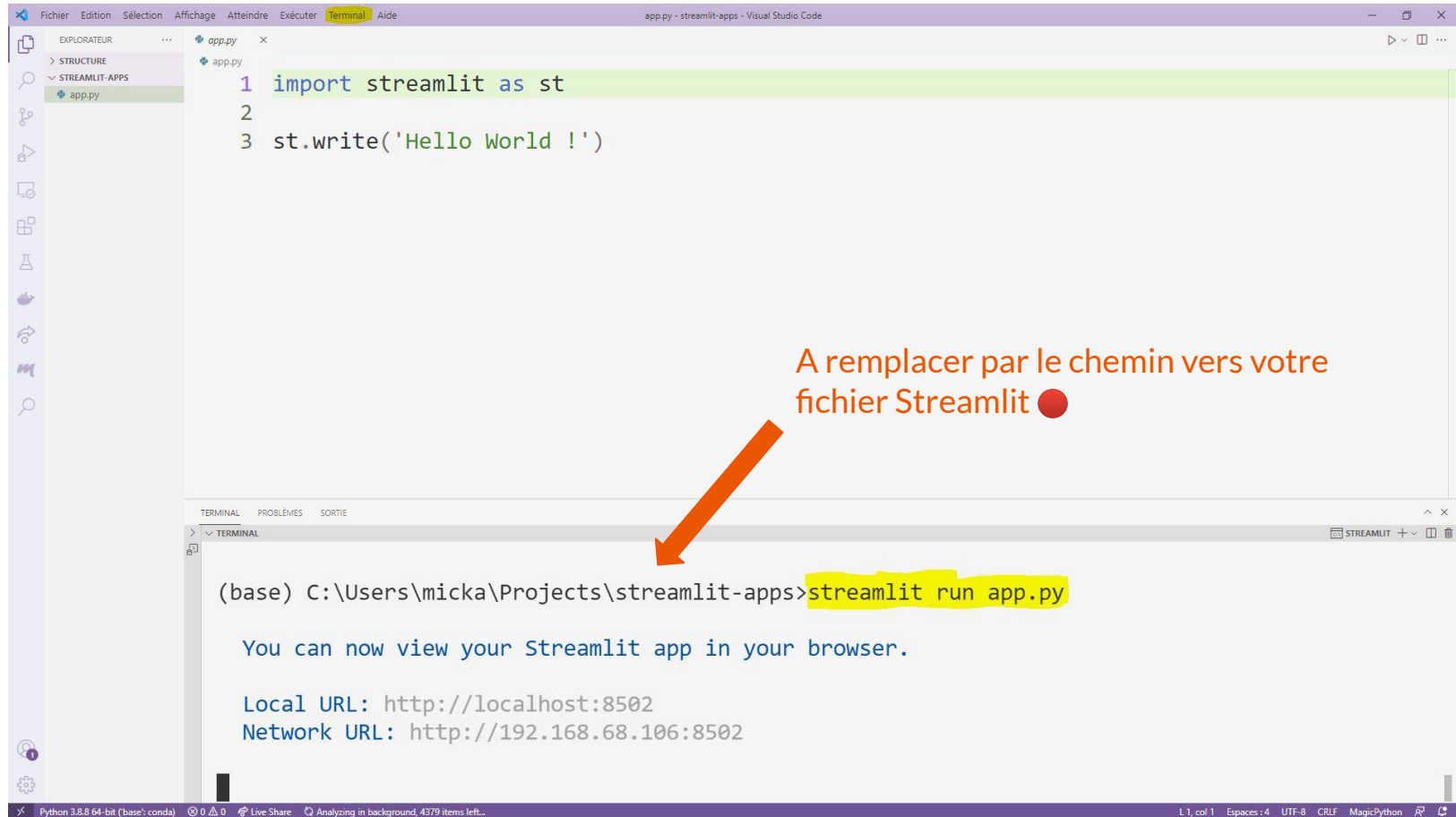
Dans l'invite de commandes :

```
pip install streamlit
```

Vérifier que ça fonctionne :

```
streamlit hello
```

Exécution dans Visual Studio Code



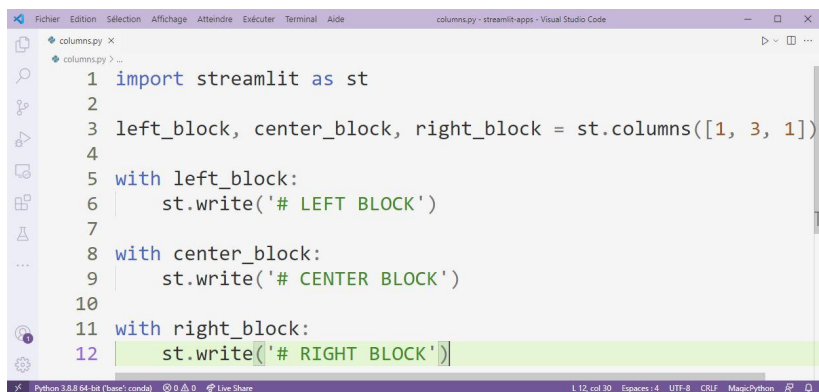
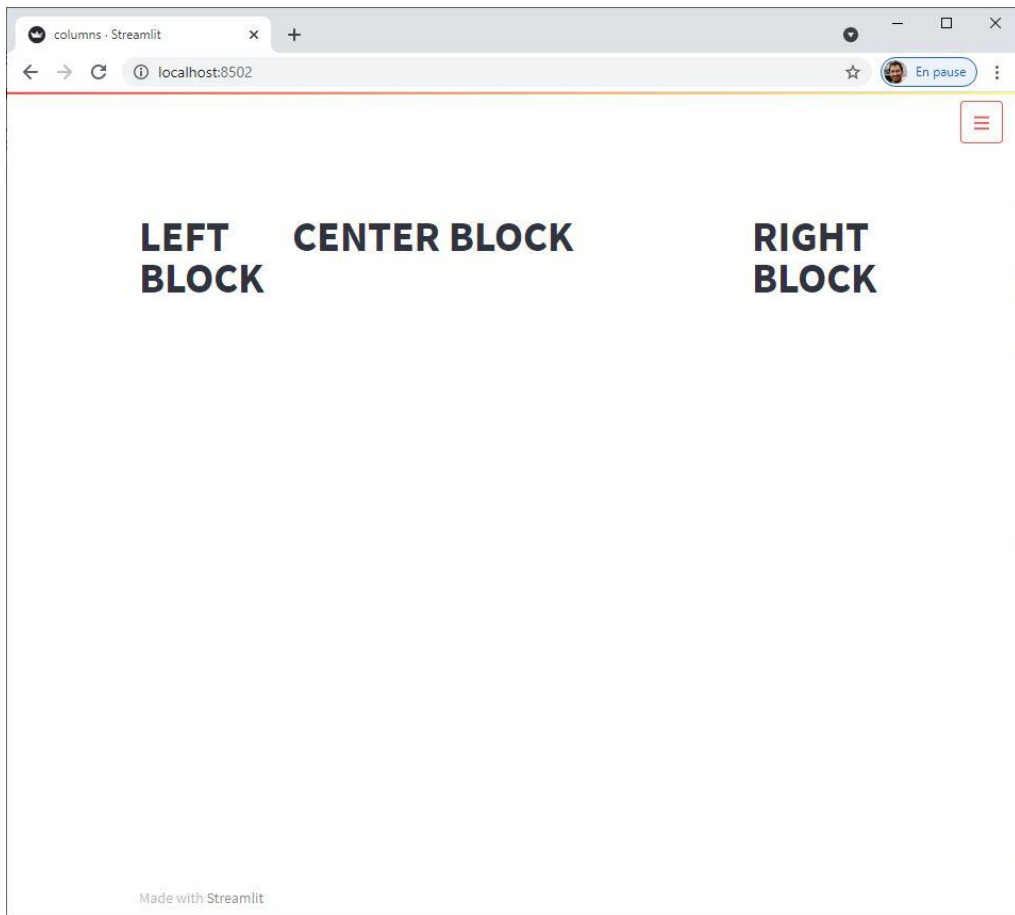
Les éléments d'interface

Columns

Containers

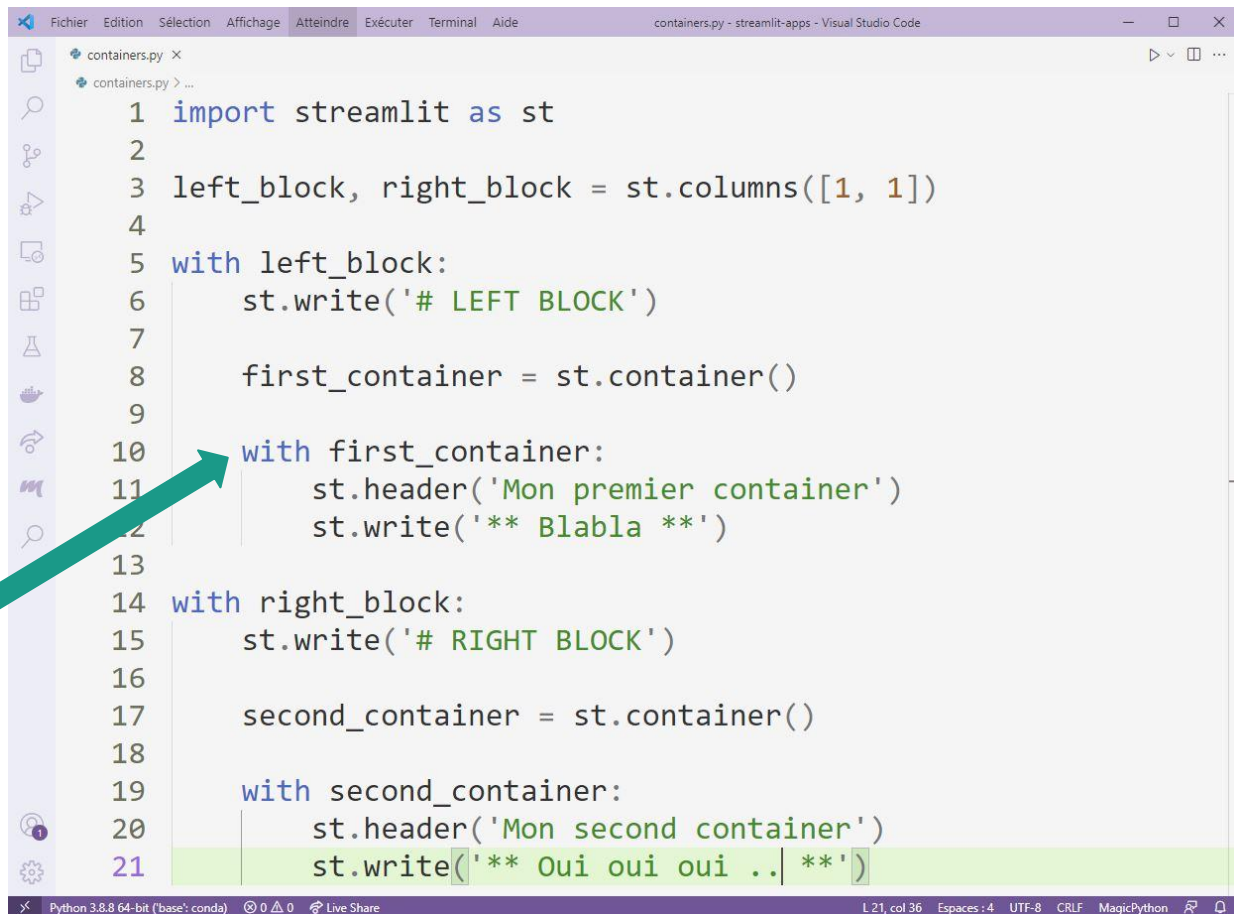
Expanders

Les Colonnes permettent d'organiser son application en blocs

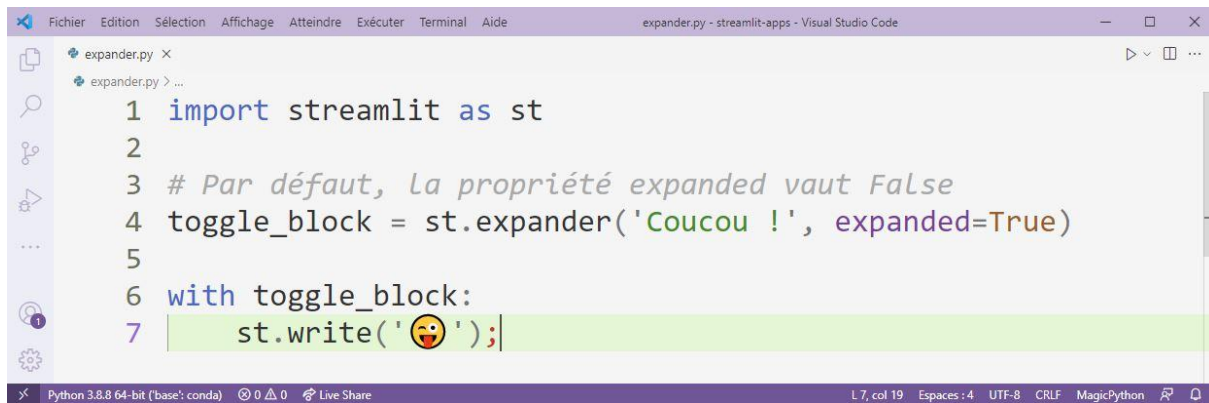
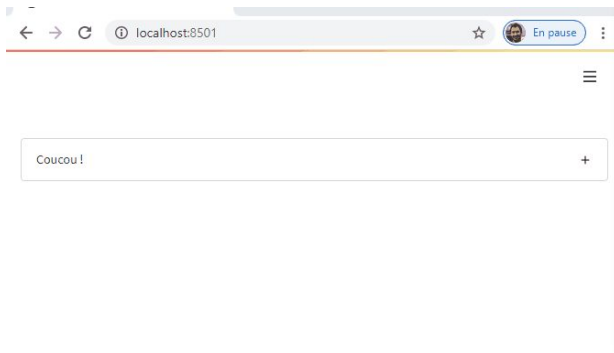


- Chaque valeur représente la **proportion d'espace** occupée par la colonne !
- On utilise **with** pour effectuer des opérations dans la colonne spécifiée.

Chaque colonne peut contenir un ou plusieurs containers



Un expander est un bloc qui s'ouvre/ferme au clic

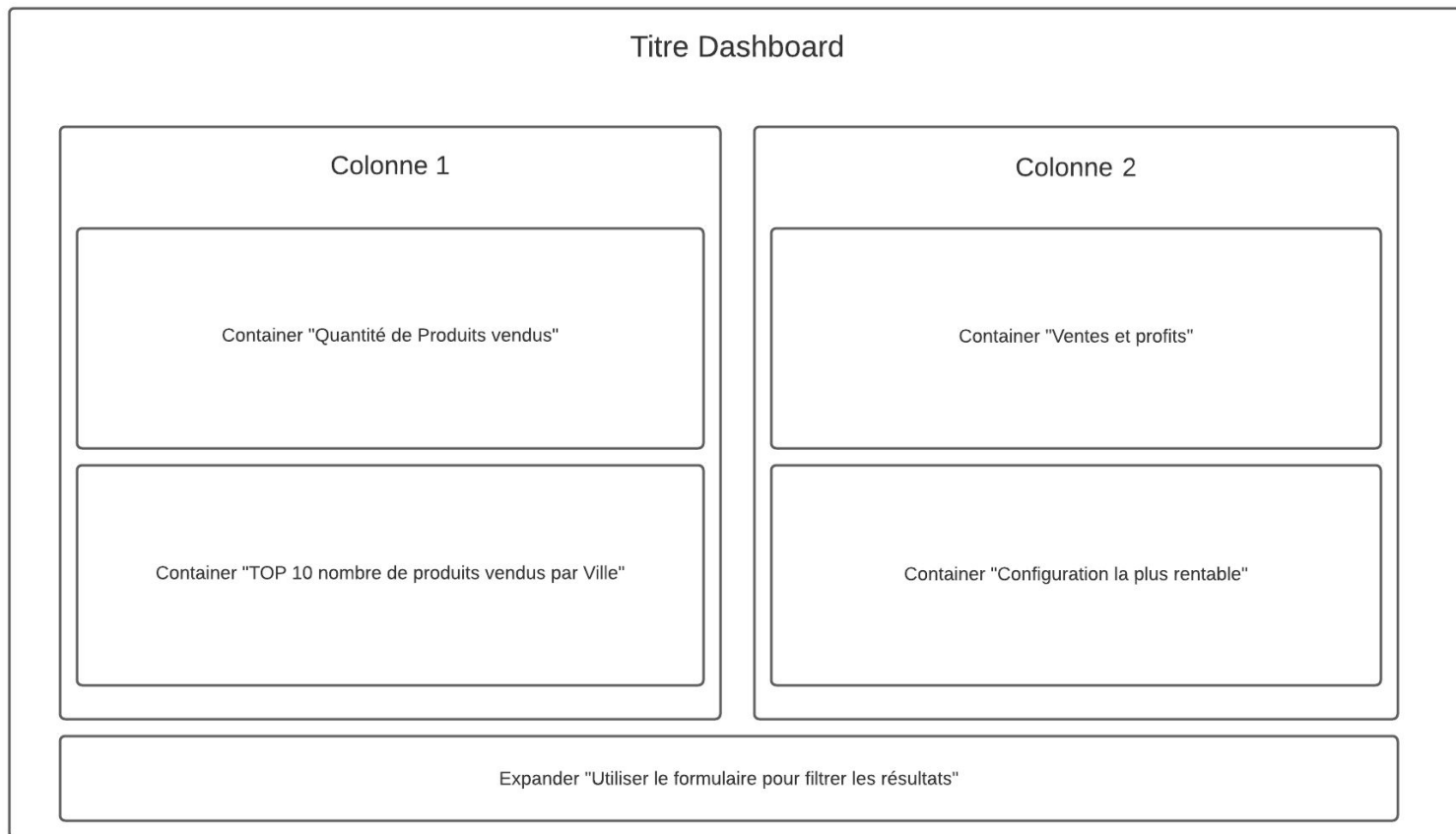


Pratique pour :

- Les instructions, compléments d'information
- Les formulaires de tri/sélection de données

Pour ouvrir le bloc par défaut, on passe la propriété **expanded** à **True**.

Préparez votre application sur papier/LucidChart



Les éléments d'interaction

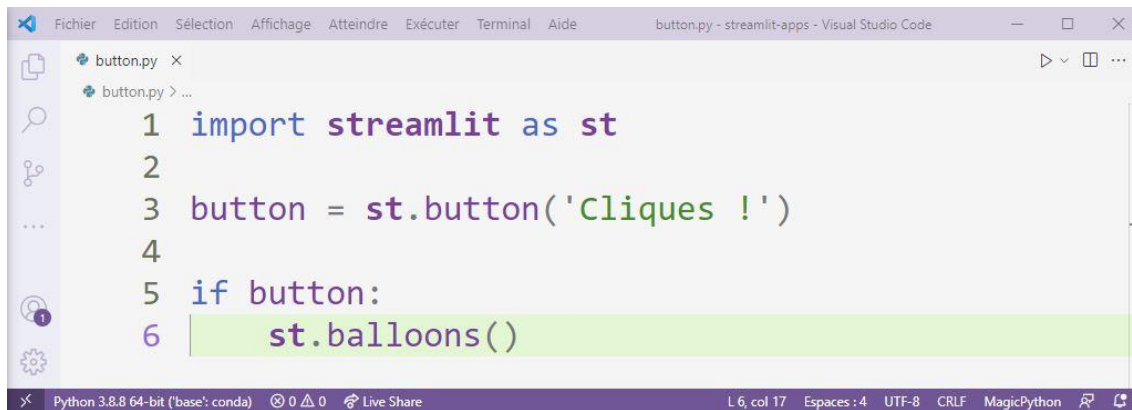
Buttons

Sélection simple

Sélection multiple

Plus difficile : récolter une plage de dates

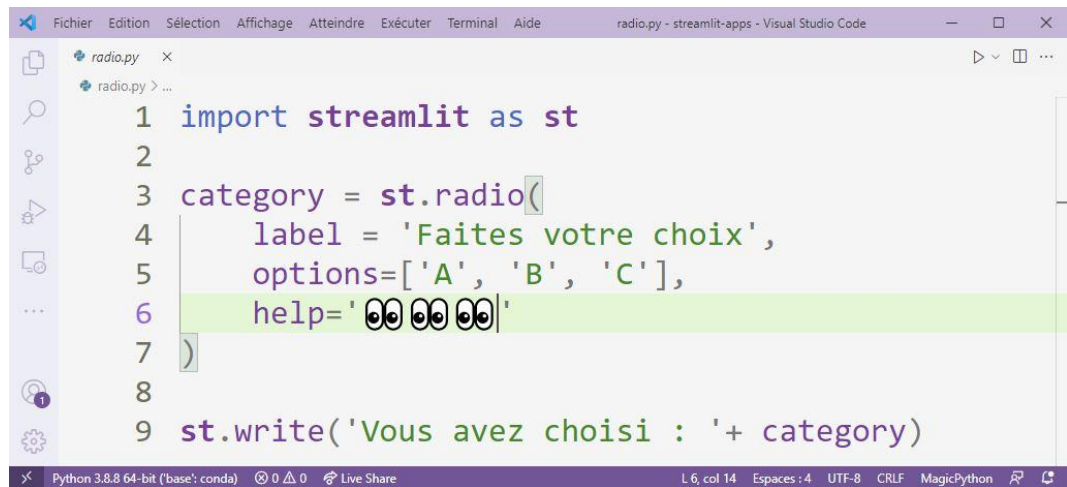
Les boutons ont un label et des options



Le bouton est le widget le plus simple, vous pouvez déclencher des actions lors du clic à l'aide d'une simple condition Python.

[st.button\(\)](#)
[st.download_button\(\)](#)

Les radio boutons pour récupérer la sélection d'une utilisatrice

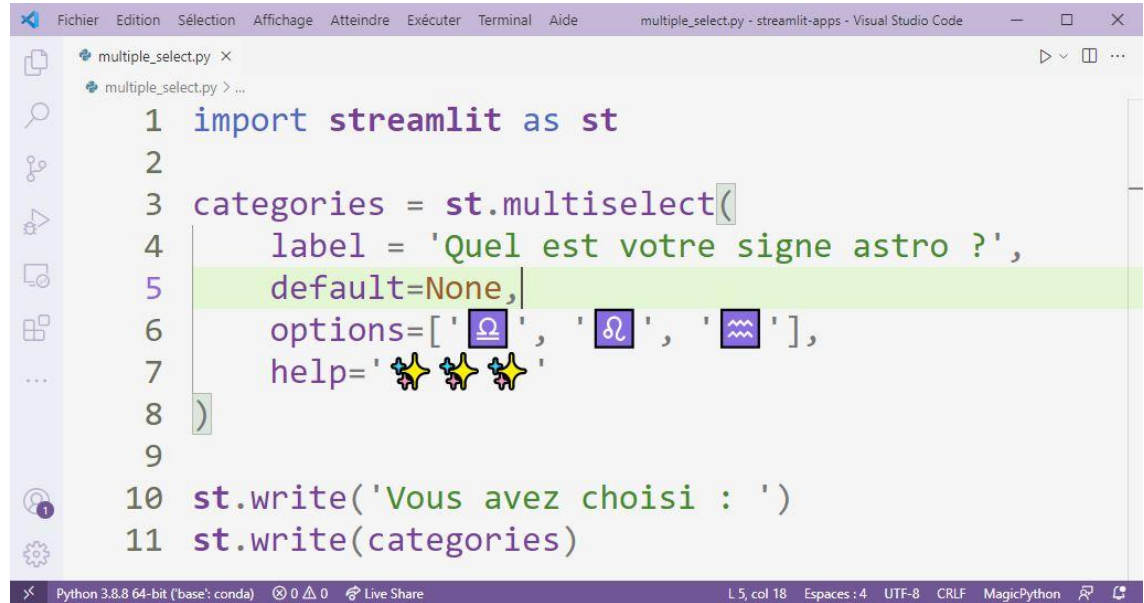


Pour valider une autorisation, privilégiez [st.checkbox\(\)](#).

[st.selectbox\(\)](#)
[st.radio\(\)](#)
[st.slider\(\)](#)
[st.date_input\(\)](#)

...

Les listes à choix pour récupérer les choix des utilisateurs



Made with Streamlit

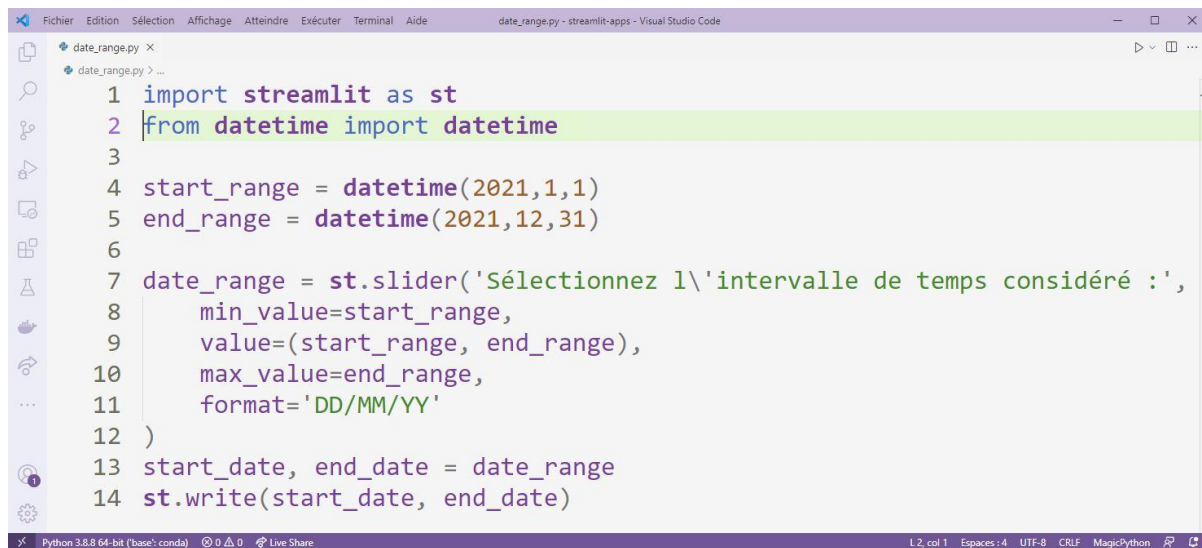
Avec la propriété **default**, on peut sélectionner les valeurs initiales. Vous passerez une liste des valeurs **parmi la liste des options disponibles**.

Malheureusement, les plages de dates ne sont pas disponibles dans Streamlit.

Hors, c'est une des fonctionnalités les plus utiles dans un Dashboard Data 😞😞

Relax, suivez le guide ! 😎

Utilisation d'un slider pour récupérer un intervalle de dates



Passez un tuple (2 valeurs) dans la propriété **value** pour récupérer deux dates.

La propriété **format** permet de définir le format de date affiché.

Passage d'une date pandas à un datetime Python

```
Fichier  Edition  Sélection  Affichage  Atteindre  Exécuter  Terminal  Aide  date_range_pd.py - streamlit-apps - Visual Studio Code

date_range_pd.py x
date_range_pd.py > ...

1 import pandas as pd
2 import streamlit as st
3
4 # Dans l'idéal, vous aurez utilisé parse_dates
5 df = pd.DataFrame([[ '2021-01-01' ], [ '2021-12-31' ]], columns=[ 'date' ])
6 df[ 'date' ] = pd.to_datetime(df[ 'date' ])
7
8 # Conversion en objets datetime Python
9 start_range = df[ 'date' ][0].to_pydatetime() # transforme en datetime Python
10 end_range = df[ 'date' ][1].to_pydatetime()
11
12 date_range = st.slider('Sélectionnez l\'intervalle de temps considéré :',
13     min_value=start_range,
14     value=(start_range, end_range),
15     max_value=end_range,
16     format='DD/MM/YY'
17 )
18 start_date, end_date = date_range
19 st.write(start_date, end_date)
```

Python 3.8.8 64-bit ('base': conda) 0 0 0 Live Share L 2, col 23 Espaces : 4 UTF-8 CRLF MagicPython

Intégration de Plotly et Streamlit

Une seule fonction à retenir : st.plotly_chart()

```
Fichier  Edition  Sélection  Affichage  Atteindre  Exécuter  Terminal  Aide  plotly_integration.py - streamlit-apps - Visual Studio Code
plotly_integration.py x
plotly_integration.py > ...
1 import pandas as pd
2 import plotly.express as px
3 import streamlit as st
4
5 super_store = pd.read_csv('Data/super_store.csv', parse_dates=['Order Date', 'Ship Date'])
6 sales_per_region = super_store[['Sales', 'Region']] \
7     .groupby('Region') \
8     .agg('sum') \
9     .round(2)
10
11 sales_per_region.reset_index(inplace=True)
12
13 fig = px.bar(sales_per_region,
14             x='Sales',
15             y='Region',
16             text='Sales',
17             title='Ventes effectuées par Région',
18             orientation='h')
19
20 st.plotly_chart(fig)
```

Python 3.8.8 64-bit ('base': conda) 0 0 0 Live Share L8, col 18 Espaces : 4 UTF-8 CRLF MagicPython

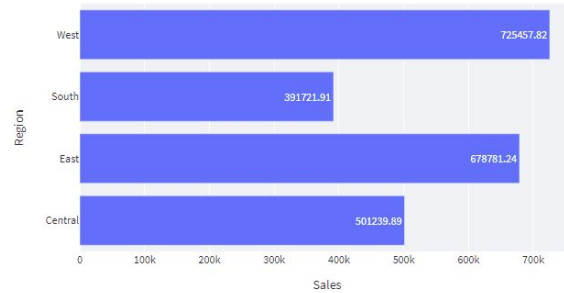
Une seule fonction à retenir : st.plotly_chart()

← → ↻ localhost:8501

☆ En pause ⋮



Ventes effectuées par Région



Les paramètres de la fonction `st.plotly_chart()`

```
st.plotly_chart(  
    fig,  
    use_container_width=True, # S'adapte à l'espace du Container  
    config={'displayModeBar': False} # Cache la barre d'options de Plotly  
)
```

Ventes effectuées par Région



```
# Exemple avec matplotlib ✈  
import matplotlib.pyplot as plt  
import streamlit as st  
import numpy as np
```

```
arr = np.random.normal(1, 1, size=100)  
fig, ax = plt.subplots()  
ax.hist(arr, bins=20)
```

```
st.pyplot(fig)
```


Adapter la requête en fonction des widgets

```
regions = st.multiselect(  
    label = 'Choisissez les régions',  
    options = sales_per_region['Region'].unique(),  
    default = ['West']  
)  
  
sales_per_region = sales_per_region.query('Region in @regions')  
  
fig = px.bar(sales_per_region, x='Region', y='Sales')  
st.plotly_chart(fig)
```

La fonction [query\(\)](#) de pandas adapte vos agrégations à partir de la sélection des utilisateurs en temps réel.

Déployer son application en ligne

Facultatif : maîtrise de git/github requise

Step 1 : création de compte **gratuit** sur [Streamlit Cloud](#)



 Settings

Workspaces

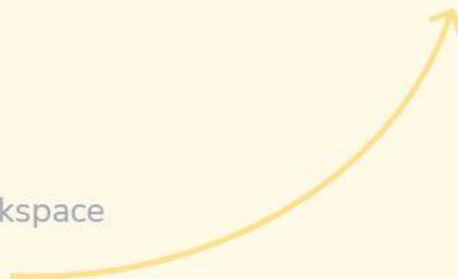
Your apps

New app



No apps to show in this workspace

Deploy one now



Step 2 : décrire les dépendances de votre projet



Créer dans le dossier de votre application un fichier appelé **requirements.txt**
Mettre dedans les lignes suivantes :

```
plotly==5.3  
pandas==1.3  
streamlit==1.1
```

Ce fichier devra être versionné sinon streamlit n'installera pas pandas et plotly 😞

Step 3 : Versionner son application sur GitHub

The screenshot shows the GitHub interface for the repository 'mickaelandrieu / learn-streamlit'. The repository is public. The navigation bar includes links for Code, Pull requests, Actions, Security, Insights, and Settings. Below the navigation bar, there are buttons for 'main' branch, '1 branch', and '0 tags'. There are also buttons for 'Go to file', 'Add file', and 'Code'. The commit history shows a commit by mickaelandrieu titled 'Dashboard is now dynamic !' with a commit hash of 94f817f, made 2 days ago, and 4 commits in total. The file list includes 'Data', 'README.md', 'app.py', and 'requirements.txt'. The README.md file is expanded, showing the title 'learn-streamlit'.

mickaelandrieu / **learn-streamlit** Public

<> Code Pull requests Actions Security Insights Settings

main 1 branch 0 tags Go to file Add file <> Code

mickaelandrieu Dashboard is now dynamic ! 94f817f 2 days ago 4 commits

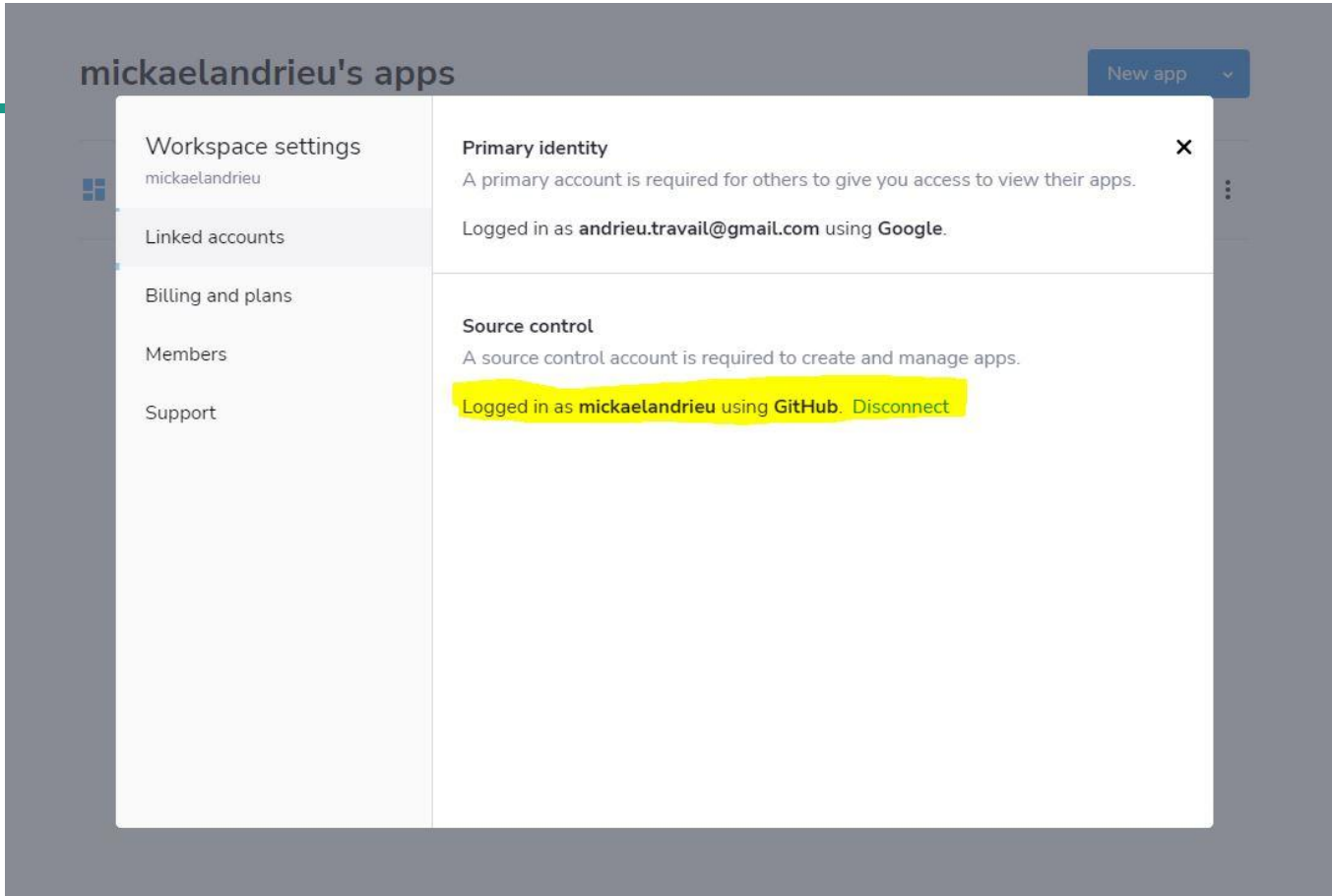
| | | |
|------------------|--------------------------------------|------------|
| Data | Added Dashboard and Super Store Data | 2 days ago |
| README.md | first commit | 2 days ago |
| app.py | Dashboard is now dynamic ! | 2 days ago |
| requirements.txt | Dashboard is now dynamic ! | 2 days ago |

README.md

learn-streamlit

Hors programme de ce cours 😞, suivre [Partie 1 & 2 de ce cours](#) gratuit.

Step 4 : Connecter Streamlit à Github



The screenshot displays the 'mickaelandrieu's apps' settings page in Streamlit. On the left, a sidebar menu contains the following items: 'Workspace settings' (with the username 'mickaelandrieu' below it), 'Linked accounts', 'Billing and plans', 'Members', and 'Support'. The 'Linked accounts' section is currently selected. The main content area is divided into two panels. The top panel, titled 'Primary identity' with a close button (X) in the top right, contains the text 'A primary account is required for others to give you access to view their apps.' and 'Logged in as andrieu.travail@gmail.com using Google.'. The bottom panel, titled 'Source control', contains the text 'A source control account is required to create and manage apps.' and 'Logged in as mickaelandrieu using GitHub. Disconnect'. The text 'Logged in as mickaelandrieu using GitHub. Disconnect' is highlighted with a yellow background.

mickaelandrieu's apps New app ▾

- Workspace settings
mickaelandrieu
- Linked accounts**
- Billing and plans
- Members
- Support

Primary identity ✕

A primary account is required for others to give you access to view their apps.

Logged in as **andrieu.travail@gmail.com** using **Google**.

Source control

A source control account is required to create and manage apps.

Logged in as **mickaelandrieu** using **GitHub**. **Disconnect**

Step 5 : Sélectionner le dépôt dans la liste

Deploy an app

Apps are deployed directly from their GitHub repo. Enter the location of your app below.

Repository

[Paste GitHub URL](#)

mickaelandrieu/learn-streamlit

Branch

main

Main file path

app.py

[Advanced settings...](#)

Deploy!

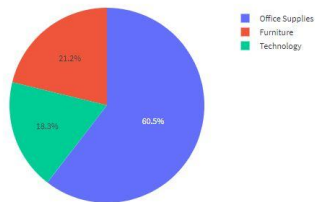
1. Cliquer sur “New app”
2. La partie en **rouge** est votre nom de compte Github
3. La partie en **bleu** est le nom du dépôt Github de votre app
4. La branche est soit **main**, soit **master**
5. Le “Main file path” est le nom de votre fichier Python
6. Cliquer sur Deploy ! 🚀💉

Step 6 : Premier déploiement

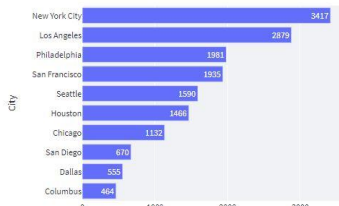
Votre premier dashboard avec Plotly et Streamlit

Quantité de produits vendus

Nombre de produits vendus par catégorie



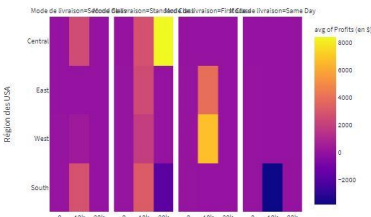
TOP 10 : Nombre de produits vendus par Ville



Ventes et Profits



Quelle est la configuration la plus rentable ?



```
Successfully uninstalled pydeck-0.7.0
Attempting uninstall: pyarrow
Found existing installation: pyarrow 5.0.0
Uninstalling pyarrow-5.0.0:
Successfully uninstalled pyarrow-5.0.0
Attempting uninstall: protobuf
Found existing installation: protobuf 3.18.1
Uninstalling protobuf-3.18.1:
Successfully uninstalled protobuf-3.18.1
Attempting uninstall: pillow
Found existing installation: Pillow 8.3.2
Uninstalling Pillow-8.3.2:
Successfully uninstalled Pillow-8.3.2
Attempting uninstall: gitpython
Found existing installation: GitPython 3.1.24
Uninstalling GitPython-3.1.24:
Successfully uninstalled GitPython-3.1.24
Attempting uninstall: click
Found existing installation: click 7.1.2
Uninstalling click-7.1.2:
Successfully uninstalled click-7.1.2
Attempting uninstall: cachetools
Found existing installation: cachetools 4.2.4
Uninstalling cachetools-4.2.4:
Successfully uninstalled cachetools-4.2.4
Attempting uninstall: blinker
Found existing installation: blinker 1.4
Uninstalling blinker-1.4:
Successfully uninstalled blinker-1.4
Attempting uninstall: base58
Found existing installation: base58 2.1.0
Uninstalling base58-2.1.0:
Successfully uninstalled base58-2.1.0
Attempting uninstall: astor
Found existing installation: astor 0.8.1
Uninstalling astor-0.8.1:
Successfully uninstalled astor-0.8.1
Attempting uninstall: altair
Found existing installation: altair 4.1.0
Uninstalling altair-4.1.0:
Successfully uninstalled altair-4.1.0
Attempting uninstall: streamlit
Found existing installation: streamlit 1.0.0
Uninstalling streamlit-1.0.0:
Successfully uninstalled streamlit-1.0.0
Successfully installed MarkupSafe-2.0.1 Send2Trash-1.8.0 altair-4.1.0 arg
WARNING: You are using pip version 21.1.1; however, version 21.3.1 is available.
You should consider upgrading via the '/home/appuser/venv/bin/python -m pip
[manager] Python dependencies were installed from /app/learn-streamlit/req
[manager] Processed dependencies!
```

A chaque nouveau “push” sur GitHub, l’application sera mise à jour sur Streamlit 🕶️

Questions ?