



PIP 2022 :manuel d'utilisation du site web et charte visuelle



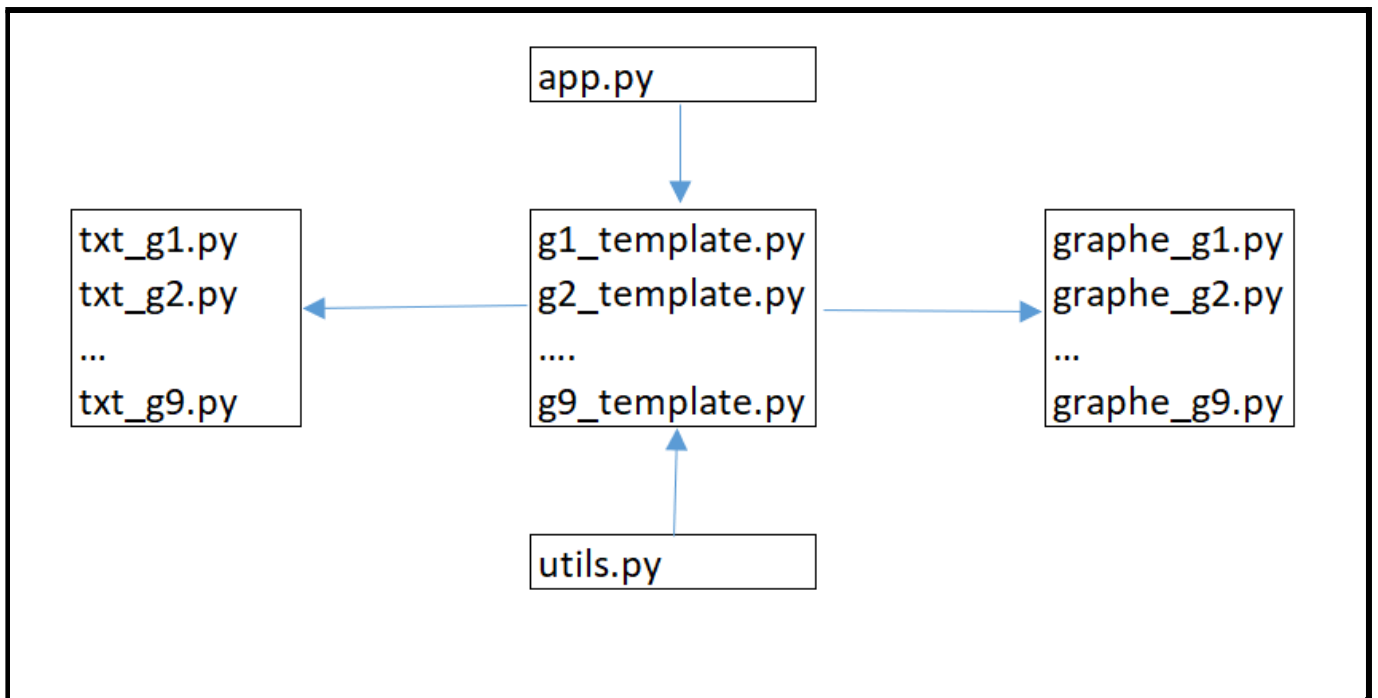
M2 SID UT3_Projet Non Alternants _ Décembre 2021
Hassan Hadda - Damien Chaligné - Amal Ghita - Emanuel Afessa

SOMMAIRE

1.Architecture	2
2.Comment créer un graphique	2
2.Charte graphique à respecter	4
2.1. Palette de couleurs	4
2.2. Règles de nommage	5
2.3. Règles de Mise en page	6
6.Liens utiles	7
7. Screenshot Dashboard	7

1.Architecture

Nous présentons une architecture où chaque groupe possède des fichiers **.py** attribué sur lequel il est tenu de faire ses développements.



- **app.py** : fichier de lancement
- **utils.py**: fichier de construction du header de tous les fichiers

g<num_groupe>_template.py

- **g<num_groupe>_template.py**: fichier qui appelle les textes et graphiques construits sur **txt_g<num_groupe>.py** et **graphe_g<num_groupe>.py**

2.Comment créer un graphique

L'accès se fait via l'URL suivant

https://github.com/hadda-hassan/PIP2022_PREPARATOIRE

La création de graphique se fait en deux étapes :

Etape 1 : Chaque groupe construit son graphe et son texte (description graphique) sur les 2 fichiers suivants :

- **graphe_g<num de groupe>.py** pour le graphique. Pour exemple, une capture du fichier *graphe_g1.py* :

```
### exemple graphe
df = px.data.tips()
fig3 = px.box(df, x="time", y="total_bill", points="all")
```

- **txt_g<num de groupe>.py** pour le texte. Pour exemple, une capture du fichier *txt_g1.py* :

```
#### text
cont_txt_1 = dbc.Row([
    dbc.Col(dbc.Card([dbc.CardBody([html.H4('Remarques explication sur le graphe ', className='card-title', style=CARD_TEXT_STYLE),
        html.P('Sample text bla bla bla bla bala 1.', style=CARD_TEXT_STYLE),]),md=12),
])
```

Etape 2 : Intégrer les graphes et textes créés lors de l'étape précédente dans le fichier **g<num de groupe>_template.py**. Cette intégration se fait via un appel des modèles. Pour exemple, une capture du fichier *g1_template.py* :

```
html.Div([ ### les 2graphes
    html.Div([dcc.Graph(figure=graphe_g1.fig1),],className="six columns",), ## graphe fig1
    html.Div([dcc.Graph(figure=graphe_g1.fig3),],className="six columns",), ## graphe fig3
],
    className="row ",),

html.Div([
    ### les deux cases textes.
    html.Div([txt_g1.cont_txt_1],className="six columns",), ## text 1
    html.Div([txt_g1.cont_txt_1],className="six columns",), ## text 2
],
    className="row ",),
```

Pour lancer : app.py doit être exécuté.

NB: Lors de la création de vos graphiques, vous êtes tenu de respecter la charte graphique. Voir la section correspondante dans ce document.

NB : Se référer au screenshot de dashboard à la fin de ce document pour avoir un visuel des codes indiqués plus haut.

NB: Pour plus de détails sur la syntaxe plotly et dash se référer à la section 'liens utiles' de ce document

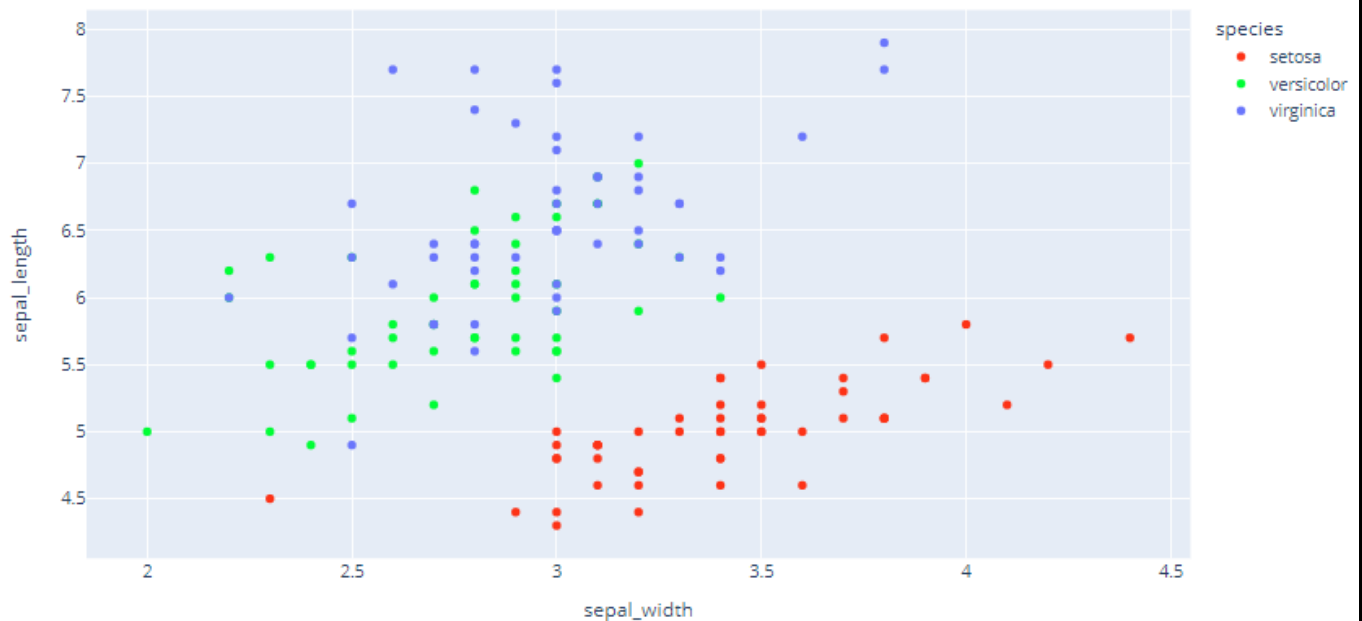
2. Charte graphique à respecter

Lors de création des graphiques veuillez respecter les conditions suivantes afin d'obtenir une plateforme inter-groupes homogène.

2.1. Palette de couleurs

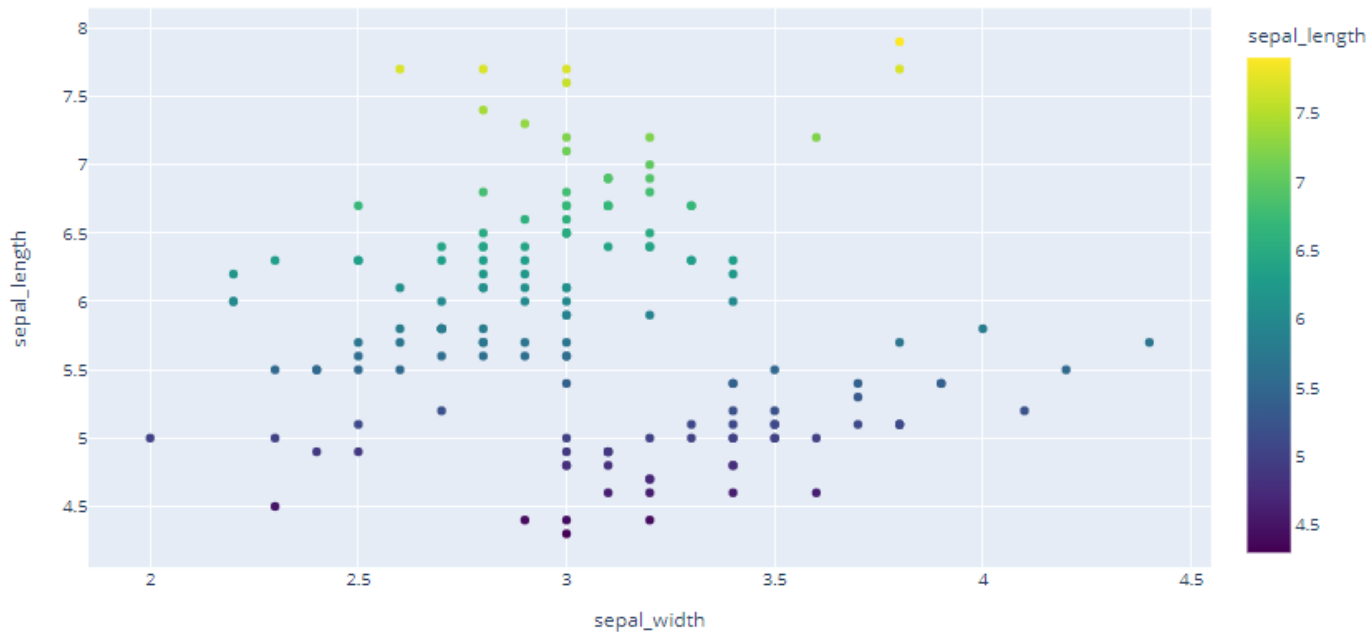
Pour les variables qualitatives le 'Light 24' est à favoriser. Voici un exemple de code et de rendu pour cette palette (Pour plus de détails sur la syntaxe Dash Plotly se référer à la section liens utiles de ce document) :

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length",
                color="species", color_discrete_sequence=px.colors.qualitative.Light24)
fig.show()
```



Pour les variables quantitatives le 'Viridis' est à favoriser. Voici un exemple de code et de rendu pour cette palette :

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length",
                color="sepal_length", color_continuous_scale=px.colors.sequential.Viridis)
fig.show()
```

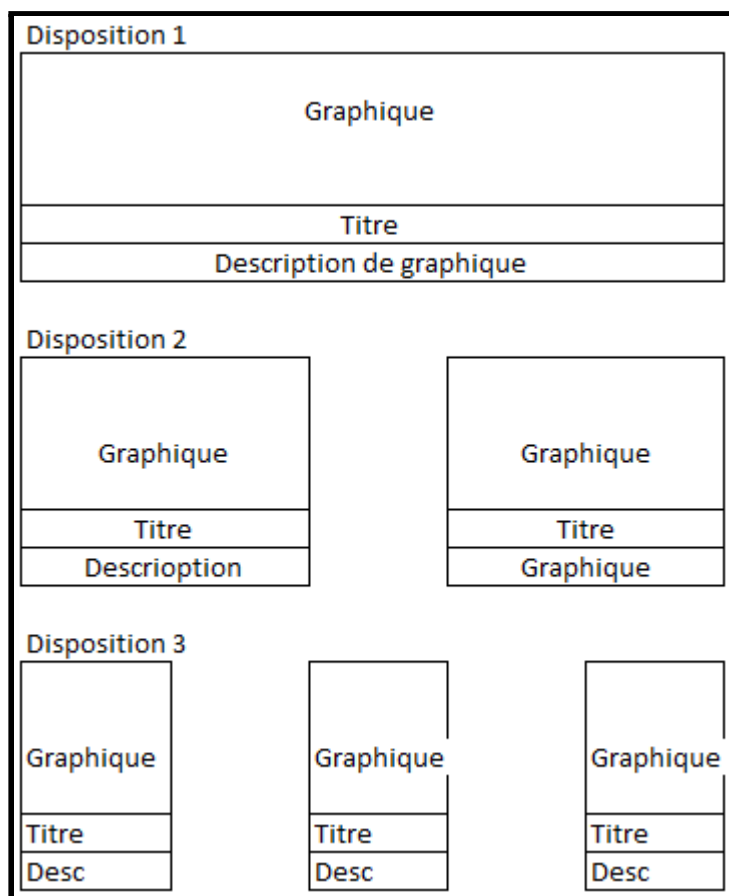


2.2. Règles de nommage

Les x et y axis sont à rédiger en minuscules et séparés par underscore si composé. L'encapsulation du titre, graphique et description de graphique est prédéfinie (se référer à la sous-section suivante). En effet, la taille, disposition et autres aspects de ces éléments sont déjà paramétrés.

2.3. Règles de Mise en page

Lors de créations de graphique, favorisez une de ces trois dispositions :



Un code permettant de reproduire chaque type de disposition est proposé sur chaque fichier **g<num de groupe>_template.py** :

Par exemple pour encapsuler votre graphique dans la Disposition 1 : se référer à ligne 47-55.

```
html.Div([
    html.Div([html.Br([]),
              html.Strong("Titre comm 2",style={"color": "#3a3a3a"}, ),
              dcc.Graph(figure=graphe_g1.fig2),],className="twelve columns",), ##fig 2
    ], className="row ",),

html.Div([
    html.Div([txt_g1.cont_txt_1],className="twelve columns",), ## casetext
], className="row ",),
```

6. Liens utiles

Liens vers documentation tutoriel plotly et Dash

- <https://plotly.com/python/>
- <https://plotly.com/dash/>

Liens vers questions spécifiques :

Personnaliser les couleurs :

- <https://stackoverflow.com/questions/60962274/plotly-how-to-change-the-colorscheme-of-a-plotly-express-scatterplot>
- https://developer.mozilla.org/en-US/docs/Web/CSS/color_value

7. Screenshot Dashboard

