

TP : Création d'une application Dash collaborative

Objectifs

- Créer une application web interactive avec Dash.
- Travailler en équipe (à 2 ou 3).
- Utiliser Git et GitHub pour la gestion de version et la collaboration.
- Livrer une application fonctionnelle avec plusieurs pages et des interactions dynamiques.
- (Bonus) Déployer l'application sur [Render](#).

Déroulement

Durée estimée : **2 à 3 heures**

Chaque groupe devra créer une application Dash structurée avec :

1. **Une page d'accueil (Home)** : introduction à l'application.
2. **Une page par membre du groupe** : chaque membre développera sa propre page contenant un import de données, un graphique, et au moins deux composantes dynamiques contrôlant le graphique via des callbacks.
3. **Une navigation fonctionnelle** entre les pages.

Livrables

1. Un dépôt public GitHub contenant le code source.
2. Une application fonctionnelle documentée dans le README.md.
3. (Bonus) Une application déployée sur Render.

Documentation

1. Support Dash : <https://univ-rennes-dash.onrender.com/>
2. Support Git : <https://github.com/LeoDtrt/univ-rennes-cours/blob/main/InitiationGit.html>
3. Support Render : <https://github.com/LeoDtrt/univ-rennes-cours/blob/main/DemoRender.html>

Liens Utiles

1. Github : <https://github.com/>
 2. Render : <https://render.com/>
-

Étapes

1. Initialisation du projet

1. **Créer un dépôt GitHub public** (nom du dépôt : tp-dash-app).

- Invitez tous les membres du groupe au dépôt.

2. **Installer les dépendances :**

- Créez un environnement virtuel
- Installez les bibliothèques

3. **Initialisez la structure du projet :**

- Créez les fichiers suivants :

```
tp-dash-app/  
|-- app.py # Point d'entrée principal  
|-- pages/  
|   |-- home.py  
|   |-- membre1.py  
|   |-- membre2.py  
|   |-- membre3.py # (si applicable)  
|-- assets/  
|   |-- style.css # Pour personnaliser le style  
|-- README.md
```

2. Création de l'application Dash

1. **Configurer app.py :**

- Importer les bibliothèques Dash et initialiser l'application.
- Configurer la navigation entre les pages.

2. **Page Home (pages/home.py) :**

- Contenir une présentation du projet et un menu de navigation entre les pages

3. **Pages Membres (pages/membre1.py) :**

- Chaque membre :
 1. Importe un jeu de données (fichier CSV, URL, ou généré dynamiquement).
 2. Crée un graphique (par ex. avec Plotly).
 3. Ajoute **au moins deux composants interactifs** (ex. : dropdown, slider) reliés au graphique via des callbacks.

4. Documentation

1. README.md :

- Description du projet.
- Instructions pour exécuter l'application localement.
- Lien vers l'application déployée (si applicable).

5. (Bonus) Déploiement sur Render

1. Configurer le déploiement et Déployer sur Render :

- Connectez votre dépôt GitHub sur Render.
 - Configurez le déploiement en vous référant au support de démonstration.
-