

Introduction à Flask

Pour commencer, prenez le temps nécessaire pour **lire**, pour **bien comprendre** et pour **manipuler** le notebook Intro_Flask.

Une fois que **c'est fait** et que **c'est clair** vous pouvez attaquer la suite.

Les questions 1, 2 et 3 reprennent exactement ce qui est fait dans le notebook donc faites l'effort de le refaire par vous-même et pas seulement copier/coller une solution.

La question 4 consiste à modifier et adapter ce qui est fait dans la fin du notebook.

Enfin, à partir de la question 5, y a de la nouveauté.

1. Faire avec Flask une page internet sur laquelle est écrit "Hello World!"

2. Faire avec Flask une page internet sur laquelle est écrit "Hello World!" en modifiant le style (pour cela il faut donc passer par une page html et fichier css pour le style)

3. Faire avec Flask un site web contenant 2 pages (une page d'accueil et une seconde page) avec bien sûr la possibilité de passer de l'une à l'autre.

4. Faire avec Flask une page /test-formulaire qui contient un formulaire permettant de rentrer le prénom, le nom, le sexe de l'utilisateur ainsi qu'un pseudo. Une fois les informations renseignées, une nouvelle page doit afficher :

"Bonjour Mr {prénom} {nom}, votre nom d'utilisateur est {pseudo}" si c'est un homme et "Bonjour Mme..." sinon.

5. Créer une base de données SQL contenant une table users avec comme colonnes : prenom, nom, sexe, pseudo.

Modifier /test-formulaire pour que désormais à chaque fois qu'un utilisateur rentre ses informations, celles-ci soient ajoutées dans la base de donnée. Pour cela il faudra utiliser un client sql python pour se connecter à la base de donnée.

Il faudra par ailleurs gérer le cas d'un pseudo déjà existant...

6. Faire une page /utilisateurs-inscrits qui permet de lister tous les noms d'utilisateurs présents dans la base de donnée.

Pour aller plus loin:

7. Faire une page avec Flask qui affiche les statistiques de base d'un dataframe que l'utilisateur aurait chargé grâce à un formulaire.

8. Charger un modèle entraîné sur MNIST et afficher la prédiction du modèle lorsque le user charge une image via un formulaire.