# **ITÉRATION 2**

# Création d’un serveur Flask

## Modalités

* Travail en binôme ou trinôme
* 1 jour

#### Cahier des charges

* Mise en place d’un serveur Flask retournant les pages HTML que vous avez généré dans l’itération précédente.

## Livrable

Votre serveur web retournant les pages web que vous avez créées précédemment.

| 1.1 — Environnement virtuel de notre projet |  |
| --- | --- |

Les environnements virtuels donnent la possibilité de **développer des projets Python en les isolant du reste du système.** En particulier isoler le projet du Python utilisé par le système d’exploitation, ainsi que des autres environnements virtuels existants.

Cela vous permet d’avoir un bien meilleur contrôle de votre projet :

* Votre projet est **reproductible** : l’ensemble des packages nécessaires - et leurs versions - sont spécifiés.
* Votre projet est **stable dans le temps** : votre projet continuera à fonctionner même si vous mettez à jour le Python de votre système par exemple.
* Votre projet est **plus facile à mettre en place** : vous êtes moins susceptible d’avoir des conflits de dépendances.

##### CONSIGNE

* Créer un environnement virtuel **conda** pour votre projet. Soit directement avec VSCode, soit en lignes de commande.

##### RESSOURCES

* Real Python [ Pourquoi utiliser un environnement virtuel ] <https://realpython.com/python-virtual-environments-a-primer/#why-do-you-need-virtual-environments>
* Option 1 : Comment créer un environnement virtuel avec VSCode

<https://code.visualstudio.com/docs/python/environments#_create-a-virtual-environment-in-the-terminal>

* Option 2 : Comment créer un environnement virtuel en ligne de commandes (CLI)

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-with-commands>

| 1.2 — Mise en place de l’architecture de base du serveur |  |
| --- | --- |

##### CONSIGNES ET RESSOURCES

* Hello World. Quoi de plus classique ?! Servez-vous de la documentation de **Flask** pour créer une application minimale qui renvoie simplement un string “Hello World”

[https://flask.palletsprojects.com/en/2.2x/quickstart/#a-minimal-application](https://flask.palletsprojects.com/en/2.2.x/quickstart/#a-minimal-application)

* Créer un fichier **environment.yml**

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#sharing-an-environment>

* Vérifier que ce fichier correspond bien à ce qui est présent dans votre environnement

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#viewing-a-list-of-the-packages-in-an-environment>

##### COMPÉTENCE ASSOCIÉE

* Mettre en place un serveur Flask avec Python.

| 1.3 — Servir du contenu HTML |  |
| --- | --- |

##### CONSIGNES ET RESSOURCES

* Une fois votre application HelloWorld mise en place, modifiez le code pour servir du contenu HTML/ CSS. Cette application comportera deux routes : une pour chaque page HTML de votre site.
  + Retourner du HTML : <https://flask.palletsprojects.com/en/2.2.x/quickstart/#rendering-templates>
  + Tutoriel : utiliser des images en Flask : <https://sentry.io/answers/serve-static-files-flask/>
  + Doc officielle : utiliser des images en Flask : <https://flask.palletsprojects.com/en/2.2.x/quickstart/#static-files>
  + Tutoriel Flask général : <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>

| 1.4 — Création d’une page HTML dynamique pour l’affichage de la galerie |
| --- |

Nous allons maintenant nous occuper de l’affichage de la partie **Galerie** du site. L’idée générale est de demander à notre serveur de générer un page HTML affichant l’ensemble des images contenues dans un répertoire. Sans connaître à l’avance ni le nombre, ni le nom des images à afficher.

##### CONSIGNES ET RESSOURCES

* Sauvegardez plusieurs images de votre choix au sein du dossier **/static/images**

**Conseil :** ne vous faites pas du mal pour rien, prenez des images de même taille.

* Utilisez le module **os** de python pour récupérer la liste des images dans le répertoire.

<https://docs.python.org/3/library/os.html>

* Utilisez le moteur de template Jinja2 pour créer une page HTML dans laquelle toutes les images seront affichées.

<https://www.rithmschool.com/courses/flask-fundamentals/templating-with-jinja2>