

# Activité : utiliser Flask avec le site repl.it

## J'accède au site repl.it

- J'accède au site à l'adresse <https://repl.it/> pour créer un nouveau repl avec le bouton **New repl** utilisant le langage **Python** :

Le site de repl.it du projet se présente en 3 parties :

1. à **gauche l'arborescence** de fichiers ;
2. au **centre l'éditeur** de fichier ;
3. à **droite la console** montrant l'exécution du programme.

## Préparation du projet

- dans le dossier du projet le fichier principal python de l'application est **main.py** ;
- dans le dossier , je crée un sous-dossier **templates** ;
- dans le sous-dossier **templates** je crée une page **index.html**

Le sous-dossier **templates** va contenir toutes les pages html du projet

## Le résultat à obtenir

## Le fichier python de l'application

- Il est important, pour le lancement de l'application, d'ajouter en paramètre l'adresse 0.0.0.0 pour le serveur. Cela donne le contenu suivant dans le fichier **app.py** : `<file python app.py> #!/usr/bin/python # -- coding:utf-8 --`

```
from flask import Flask, render_template, request app = Flask(name_)
```

```
@app.route('/') def index():
```

```
    return render_template('index.html')
```

```
if name == 'main':
```

```
    app.run(host='0.0.0.0', debug=True)
```

```
</file>
```

## Explications des instructions

- l'instruction `from` permet d'importer
  - le module **Flask** qui gère le serveur HTTP ;
  - le module **render\_template** qui permet d'utiliser les pages HTML comme modèle ou **template** ;
  - le module **request** pour gérer les **envois** de données avec la **méthode GET ou POST** ;
  - la création de la variable **app** est fondamental car il s'agit de l'application Web ;
  - **app.run(debug=True)** permet de **lancer** l'application et donc le serveur Web. Le paramètre **debug** lance l'application dans ce mode car cela est **utile lors du développement** de mon application pour **détecter les erreurs** et permettre de **rafraichir** les pages web.
  - la fonction **index** correspond à une **vue**. Elle peut porter le nom que je veux :
    - elle permet au serveur de **répondre** à une requête à l'adresse, **la route**, indiquée à la ligne précédente
    - cette ligne située juste avant la fonction **index** est un décorateur (**@app.route**) qui indique :
      - quelle **route** (adresse) permet d'exécuter la fonction **index** et donc au serveur de **renvoyer sa réponse** ;
      - de **préparer la bonne exécution** de la fonction **index**
    - la fonction **index** va utiliser le **moteur de template** (modèle) de Flask qui s'appelle **Jinja2**, pour utiliser la page passée en paramètre comme modèle de réponse et, comme je verrai plus tard, permettre de personnaliser la page avant son envoi au navigateur.

Une fonction **décorée** par **@app.route** pour renvoyer une page Web est appelée **vue**.

**ATTENTION :**

Mon application **ne doit pas avoir** deux routes identiques, ou deux vues portant le même nom !

## Le fichier HTML de la page Web

- Ecrivez le contenu suivant dans le fichier **index.html** : `<file html index.html> <!DOCTYPE html> <html lang="fr"`

```
|
```

```
>
```

```
<
```

```
head> <meta charset="UTF-8"
```

```
|
```

```
>
```

```
<title>Application Flask</title>
```

```
</head>
```

```
<
```

```
body> bonjour
```

```
</body> </html> </file>
```

## Retour à l'activité : Utiliser le framework Flask pour créer un site Web dynamique

- [Activité : Utiliser le framework Flask pour créer un site Web dynamique](#)

From:

/ - **Les cours du BTS SIO**

Permanent link:

[/doku.php/isn/flask\\_replit?rev=1575995784](/doku.php/isn/flask_replit?rev=1575995784)

Last update: **2019/12/10 17:36**

