

### 3 - Mise en pratique → Exercice 1 → Démarrage d'un projet Flask

#### 3.1-option 1 : Installation "globale" et création d'un dossier pour le projet

Comme pour les autres packages Python, l'installation peut se faire en ligne de commande en utilisant l'utilitaire `pip`:

```
pip install Flask Jinja2
```

Si vous souhaitez les installer dans votre espace utilisateur vous pouvez utiliser l'argument `--user`:

```
pip install --user Flask Jinja2
```

On crée ensuite le dossier (`projet_flask`) dans lequel on va placer le code de notre projet :

```
mkdir projet_flask
```

#### 3.1-option 2 : Installation dans un environnement virtuel au sein du dossier du projet

Cette solution sera préférée si vous avez l'habitude d'utiliser des environnements virtuels Python :

```
mkdir projet_flask && cd projet_flask  
python3 -m venv env
```

- `projet_flask` représente le dossier dans lequel on va placer le code du projet
- `env` représente le nom de votre environnement virtuel, vous êtes libre d'en choisir un autre

On active ensuite l'environnement virtuel et on y installe Flask et Jinja2 :

```
source env/bin/activate  
pip install Flask Jinja2
```

#### 3.2 Structure du projet

Dans le dossier `projet_flask` vous allez créer un fichier `app.py` ainsi qu'un dossier `templates`. Dans le dossier `templates` vous décompresserez les données du jour : données TP5 - il s'agit de 2 fichiers de `templates` utilisant Jinja2.

Vous devez donc avoir la structure de projet suivante:

```
projet_flask  
├── app.py          <--- Le code de l'appli Web à réaliser
```

```
├── templates          <--- Le dossier contenant nos modèles de documents HTML
│   ├── index.html
│   └── header.html
└── env                <--- Seulement si vous avez utilisé un environnement virtuel !
```

### 3.3 Premier bloc de code et vérification du démarrage de l'application

Utilisez le premier extrait de code de ce TP (ici) pour créer votre application Flask dans le fichier `app.py`. Ajoutez le morceau de code suivant à la fin du fichier :

```
if __name__ == '__main__':
    app.run(debug=True)
```

Il vous permettra de lancer l'application en utilisant la commande :

```
python3 app.py
```

#### Si cette façon de faire ne fonctionne pas :

Rendez-vous dans le dossier dans lequel se situe votre code :

```
cd MonDossier/projet_flask
```

Il est possible d'utiliser l'outil en ligne de commande de Flask pour démarrer l'application. Dans ce cas on devra définir la variable d'environnement `FLASK_APP` pour quelle contiennent le nom du fichier à exécuter :

```
set FLASK_APP=app.py # Sous windows
# ... ou
export FLASK_APP=app.py # Sous Linux
```

L'application peut ensuite être démarrée de la manière suivante :

```
flask run
```

**Il est ensuite possible d'accéder à l'application avec un navigateur web : `http://127.0.0.1:5000/`.**

Comme vu lors du premier cours, `127.0.0.1` désigne votre **adresse IP** et `5000` désigne le **port TCP** sur lequel votre application écoute et émet.

## 0. Inscription à PythonAnywhere

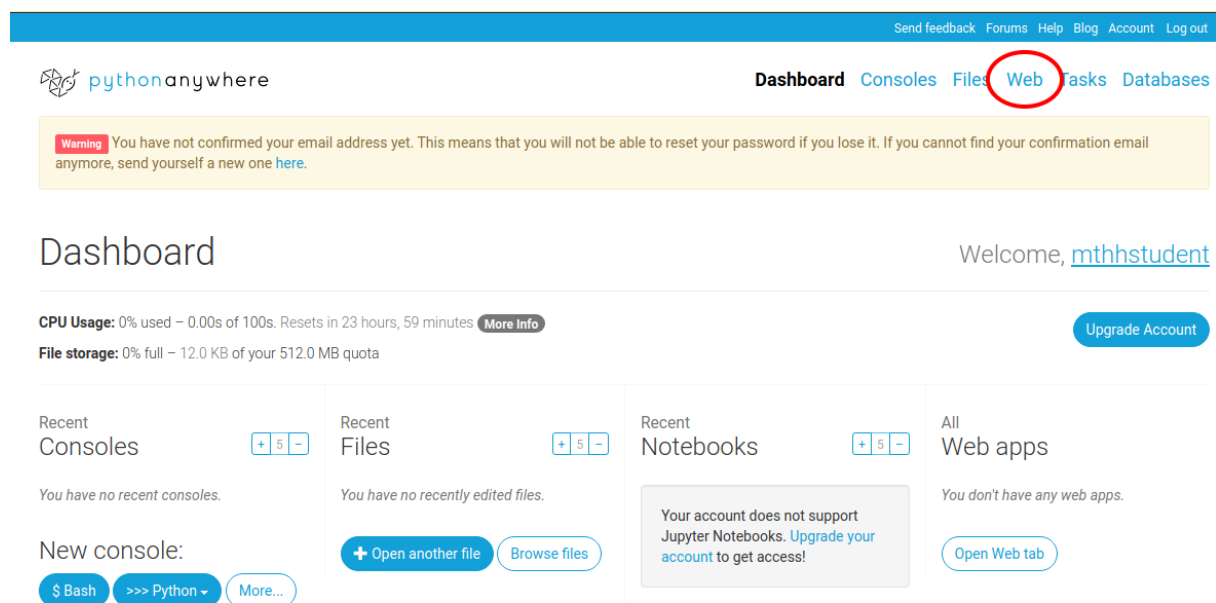
L'inscription se passe à l'URL suivante : <https://www.pythonanywhere.com/registration/register/beginner/>.

Cette plateforme va vous permettre d'exécuter du code Python sur un serveur distant. Cette solution peut être utile si vous voulez vous libérer des contraintes liées à la plateforme sur laquelle vous travaillez (système d'exploitation, droits administrateurs, etc.).

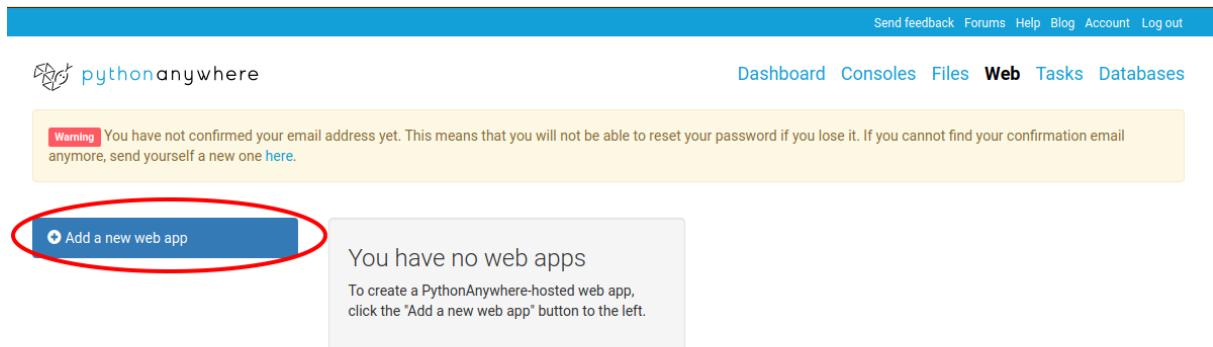
Dans notre cas, elle va être utile car elle permet d'héberger une application Web (créée avec Flask mais aussi avec Web2py ou Django) et d'y accéder à une URL de la forme **http://{pseudonyme}.pythonanyw**

Suivez les indications suivantes pour y créer une application Flask vide.

- Après vous être enregistré et loggé sur la plateforme, cliquez sur le bouton “Web” :



- Cliquez sur le bouton “Add a new web app” :



- Une fenetre s'ouvre, cliquez sur "next" :

Create new web app

## Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `mtlhstudent.pythonanywhere.com`.

Want to change that? [Upgrade now!](#)

Otherwise, just click "Next" to continue.

Cancel

« Back

Next »

- Choisissez d'utiliser le *framework* **Flask** :

Create new web app

## Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » ~~web2py~~
- » **Flask**
- » Bottle
- » **Manual configuration** (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Cancel

« Back

Next »

- Sélectionnez la version de Python avec laquelle vous avez l'habitude de travailler :

Create new web app

## Select a Python version

- » Python 2.7 (Flask 1.1.1)
- » Python 3.5 (Flask 1.1.1)
- » Python 3.6 (Flask 1.1.1)
- » Python 3.7 (Flask 1.1.1)
- » Python 3.8 (Flask 1.1.1)

**Note:** If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are [instructions here](#).

Cancel

« Back

Next »

- Ne modifiez pas le chemin du fichier `flask_app.py` et cliquez sur “next” :

Create new web app

## Quickstart new Flask project

Enter a path for a Python file you wish to use to hold your Flask app. If this file already exists, its contents will be overwritten with the new app.


Path

/home/mthhstudent/mysite/flask\_app.py

Cancel

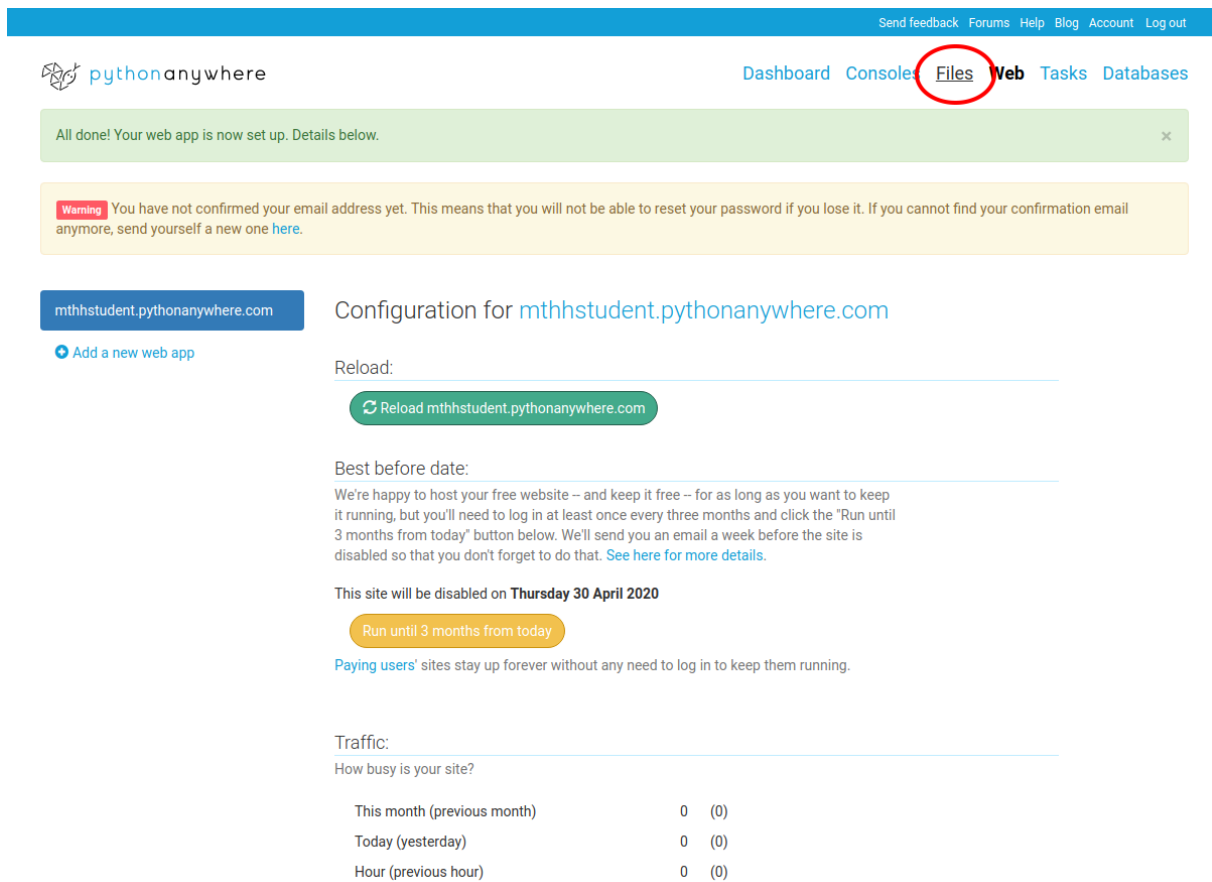
« Back

Next »



- L'application est créée et accessible à l'URL **`https://{pseudonyme}.pythonanywhere.com`**. Cliquez sur le bouton "Files" pour éditer le code de votre serveur :





Send feedback Forums Help Blog Account Log out

pythonanywhere Dashboard Consoles **Files** Web Tasks Databases

All done! Your web app is now set up. Details below.

**Warning** You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

mthhstudent.pythonanywhere.com

➕ Add a new web app

Configuration for mthhstudent.pythonanywhere.com

Reload:

↻ Reload mthhstudent.pythonanywhere.com

Best before date:

We're happy to host your free website -- and keep it free -- for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details](#).

This site will be disabled on **Thursday 30 April 2020**

Run until 3 months from today

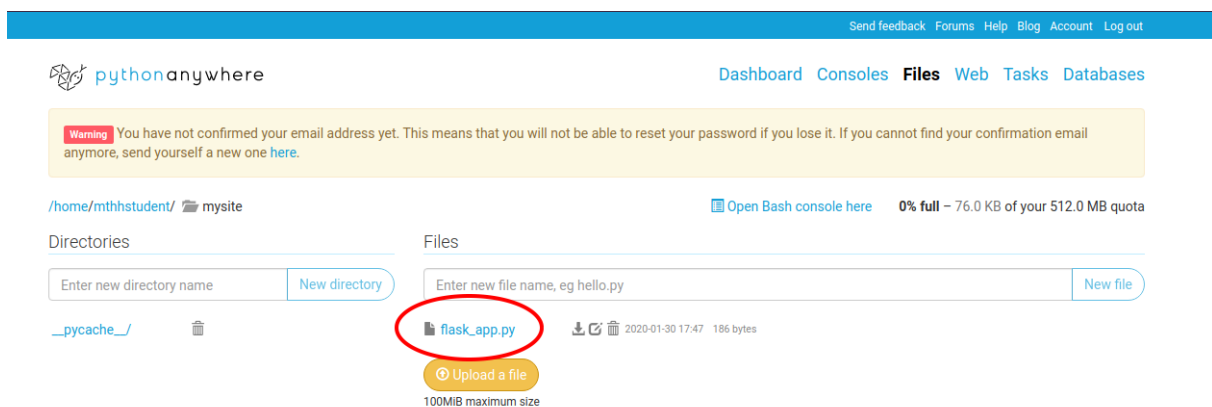
Paying users' sites stay up forever without any need to log in to keep them running.

Traffic:

How busy is your site?

This month (previous month)	0	(0)
Today (yesterday)	0	(0)
Hour (previous hour)	0	(0)

- Naviguez dans l'arborescence jusqu'à trouver ce fichier. Cliquez pour l'ouvrir :



Send feedback Forums Help Blog Account Log out

pythonanywhere Dashboard Consoles **Files** Web Tasks Databases

**Warning** You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

/home/mthhstudent/ mysite

Open Bash console here 0% full - 76.0 KB of your 512.0 MB quota

Directories

Enter new directory name New directory

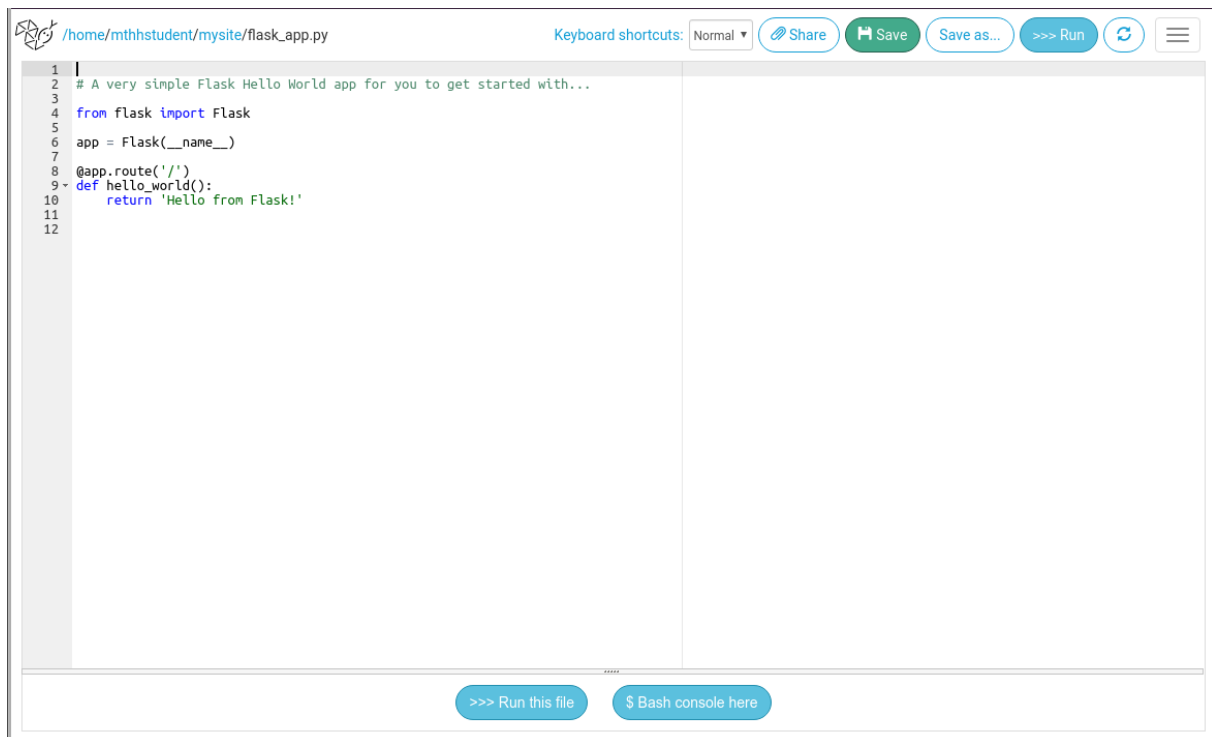
Files

Enter new file name, eg hello.py New file

flask\_app.py 2020-01-30 17:47 186 bytes

Upload a file 100MiB maximum size

- Vous pouvez l'éditer (et le sauvegarder en faisant ctrl + s par exemple).

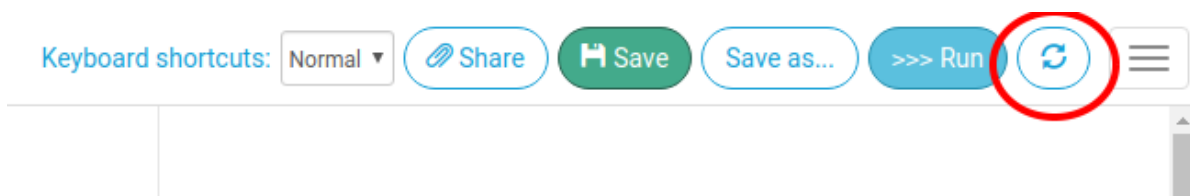


The screenshot shows a code editor interface with a file path `/home/mthhstudent/mysite/flask_app.py` at the top. The code is as follows:

```
1 |
2 | # A very simple Flask Hello World app for you to get started with...
3 |
4 | from flask import Flask
5 |
6 | app = Flask(__name__)
7 |
8 | @app.route('/')
9 | def hello_world():
10 |     return 'Hello from Flask!'
11 |
12 |
```

At the bottom of the editor, there are two buttons: `>>> Run this file` and `$ Bash console here`.

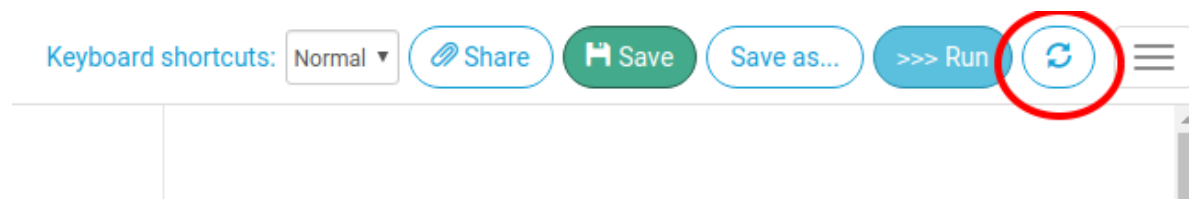
- Après l'avoir édité, vous devez recharger votre *web app*. Cela peut se faire directement depuis la page de l'éditeur de code (image ci-dessous) ou depuis la section **Web** de votre espace *PythonAnywhere*.



→ Copiez le code de l'application `app.py` créée dans le TP précédent dans le fichier `flask_app.py` dans votre espace personnel sur *PythonAnywhere*. Les étudiants qui ont eu des difficultés avec l'installation de `Flask` lors du dernier TP peuvent récupérer les fichiers attendus : `TP5_correction.zip`

→ Créez un dossier `templates` et ajoutez-y les 2 fichiers de *template* : `index.html` et `header.html`.

→ Rechargez votre application et vérifiez que tout fonctionne correctement.



**Remarque:** Vous pouvez consulter les logs d'erreur de l'application à l'adresse <https://www.pythonanywhere.com/>