

TP2 Flask: templates

Rappel TP1 :

tpflask.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/home")
def home():
    return "home page"

@app.route("/about")
def about():
    return "about page"
```

serveur.py

```
from tpflask import app
app.run(debug=True)
```

dans le terminal tapez python 'serveur.py'

Créez un nouveau répertoire 'templates' dans le répertoire du projet

Créez un nouveau répertoire 'static' dans le répertoire du projet

Dans le répertoire 'templates' Créez trois fichiers HTML, about.html, home.html

Initialisez les fichiers HTML par les balises HTML de base.

Modifiez la ligne du import dans le fichier tpflask1.py

```
from flask import Flask, render_template
```

La fonction render_template permet la redirection vers une page HTML dans le répertoire templates

Modifiez le return de la fonction home()

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

Envoyer des variables avec la fonction render_template vers la page home.html

```
return render_template('home.html', a=somme)
```

Il faut initialiser la variable somme avec une valeur.

Pour la récupération et l'affichage de la variable 'a' il faut utiliser la syntaxe de Jinja2 : {{a}} dans une balise HTML inclus dans la balise body

Créez une page layout.html dans le répertoire templates.

Ajoutez ce block dans la balise body du fichier layout.html

```
{% block content%} {% endblock %}
```

Ensuite supprimer le contenu de la page home.html about.html

Ajoutez ce code dans ces deux pages.

```
{% extends "layout.html"%}
{% block content%}
<h1>Météo Essaouira</h1>

{% endblock content%}
```

Passons aux choses sérieux !!!!!!!!!!!!!!!

Conseil : utilisez googlecolab pour tester vos codes.

Maintenant, nous allons créer une application qui permet de récupérer les prévisions météorologiques de la ville d'Essaouira de la date d'aujourd'hui via une API de Open-Meteo.com.

Vérifiez le contenu de ce lien dans le navigateur

https://api.open-meteo.com/v1/forecast?latitude=31,51&longitude=-9,77&hourly=temperature_2m&start_date=2024-03-20&end_date=2024-03-20

Consultez la documentation de l'API pour pouvoir modifier les paramètres de cet URL

<https://open-meteo.com/en/docs>

NB : Latitude et longitude de la ville d'Essaouira est : 31.506327, -9.754354.

Installez le module requests en utilisant pip

Ajoutez ce code dans le fichier tpflask1.py

```
import requests
import json
url="https://api.open-meteo.com/v1/forecast?latitude=31,51&longitude=-9,77&hourly=temperature_2m&start_date=2024-03-19&end_date=2024-03-19"

response=requests.get(url)
response=requests.get(url).content.decode('utf-8')
data = json.loads(response)
print(data)
```

Notre objectif est d'afficher les prévisions météo d'aujourd'hui de la ville d'ESSAOUIRA, ri un peu similaire à cet affichage dans la page home.html associer à l'URL /home.



Le résultat stocké dans le variable data est un dictionnaire (ou liste de deux dictionnaire dans ce cas on utilise data[0])

On doit extraire les données de chaque 3h

Par exemple :

```
print(data[0]["hourly"]["temperature_2m"])
```

Affiche les températures de chaque heure (24 valeurs)

[14.9, 14.4, 13.9, 14.8, 14.8, 14.2, 14.3, 16.6, 21.0, 23.3, 25.3, 26.5, 26.8, 27.1, 27.0, 26.7, 25.5, 23.1, 20.8, 17.6, 15.8, 15.0, 14.9]

Ajouter ce code pour récupérer la date d'aujourd'hui automatiquement

```
from datetime import datetime
dateLyouma=datetime.today().strftime("%Y-%m-%d")
print(dateLyouma)
```

Modifier l'URL

```
url="https://api.open-meteo.com/v1/forecast?latitude=31,51&longitude=-9,77&hourly=temperature_2m&start_date="+dateLyouma+"&end_date="+dateLyouma
```

Pour afficher le nom du jour en anglais et le traduire en français (la variables 'jours' est de type dictionnaire):

```
nomLyouma=datetime.today().strftime("%A")
print(nomLyouma)
jours={'Monday':'Lundi', 'Tuesday':'Mardi', 'Wednesday':'Mercredi',
'Thursday':'jeudi', 'Friday':'Vendredi', 'Saturday':''}
print(jours[nomLyouma])
```

pour récupérer les précipitation, la vitesse du vent etc. Visiter la documentation de l'API

<https://open-meteo.com/en/docs>

Après avoir tester le code sur googlecolab intégrez le dans le fichier tpflask1.py

En suite envoyer seulement les informations à afficher vers la page avec la fonction render_template

Par exemple pour envoyer la liste des températures (vous devez ajouter les autres infos {le vent, précipitation ...} dans la fonction render_template)

```
listeTemperatures=data[0]["hourly"]["temperature_2m"]
blabla
render_template('home.html', lesTemperatures= listeTemperatures)
```

Dans la page home.html vous pouvez afficher les infos envoyées par la fonction render_template

```
{% extends "layout.html"%}
{% block content%}
<h1>Météo Essaouira</h1>

{% endblock content%}
```

Utilisez la syntaxe de Jinja2 : <https://jinja.palletsprojects.com/en/3.1.x/templates/>

Par exemple :

```
{% for item in navigation %}
    <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
{% endfor %}

{% if True %}
    Bla bla
{% endif %}
```

Une Aide!!!

```
windsliste=data[0]["hourly"]["windspeed_10m"]
precipitationliste=data[0]["hourly"]["precipitation"]
cloud_cover=data[0]["hourly"]["cloud_cover"]
```