



Sommaire

TP1 - Exploration de l'API.....	2
TP2 - Exploitation des données.....	2
TP3 - Traitement de données.....	3
TP4 - Flask – Exploration.....	4
TP5 - Flask – Ajout d'une route simple.....	4
TP6 - Flask – Ajout d'une route dynamique.....	5
TP7 - Flask – Template.....	5
TP8 - Flask – Ajout d'une route.....	6
TP9 - Flask – Ajout d'un template.....	6

TP1 - Exploration de l'API

- Sur votre compte, créer un dossier **TP-Chapitre** dans lequel on va enregistrer les TP de ce chapitre. Récupérer le fichier **API_meteo.py** et l'enregistrer dans le dossier **TP-Chapitre**.
- Lire les commentaires du fichier et récupérer le fichier pdf qui documente l'API et l'ajouter le à votre répertoire de travail.
- Installer la bibliothèque **request**.
- Lancer les différentes fonctions test et chercher à comprendre leur fonctionnement
- Quelle est la clé du dictionnaire **dico_reponse** qui permet d'avoir les informations sur la ville ?

- Quelle est la clé du dictionnaire **dico_reponse["current_condition"]** qui permet de connaître la vitesse du vent actuelle ?

- Ajouter une fonction **tp1()** qui affiche dans la console, la température actuelle.

```
Console x
>>> %Run 'api_meteo - niveau 1.py'
>>> tp1()
température actuelle : 7
```



Lever la main pour valider ce TP.

TP2 - Exploitation des données

- Toujours dans le fichier **meteo_api.py**, ajouter une fonction **tp2()** qui affiche l'heure du lever ainsi que l'heure du coucher du soleil.

```
>>> %Run 'api_meteo - niveau 1.py'
>>> tp2()
heure du lever du soleil : 06:46
heure du coucher du soleil : 18:38
```



Lever la main pour valider ce TP.

TP3 - Traitement de données

- Toujours dans le fichier **meteo_api.py**, ajouter une fonction **tp3()** qui affiche la liste des prévisions de températures du jour.

```
>>> tp3()
prévisions températures
3.7 3.9 3.6 3.3 4.5 2.5 2.6 3.3 3 4.7 5 5.3 6.1 7.3 8.3 8.4 9.3 9.9 8.5 6.3 8.7 6.1 6.5 7.1
```

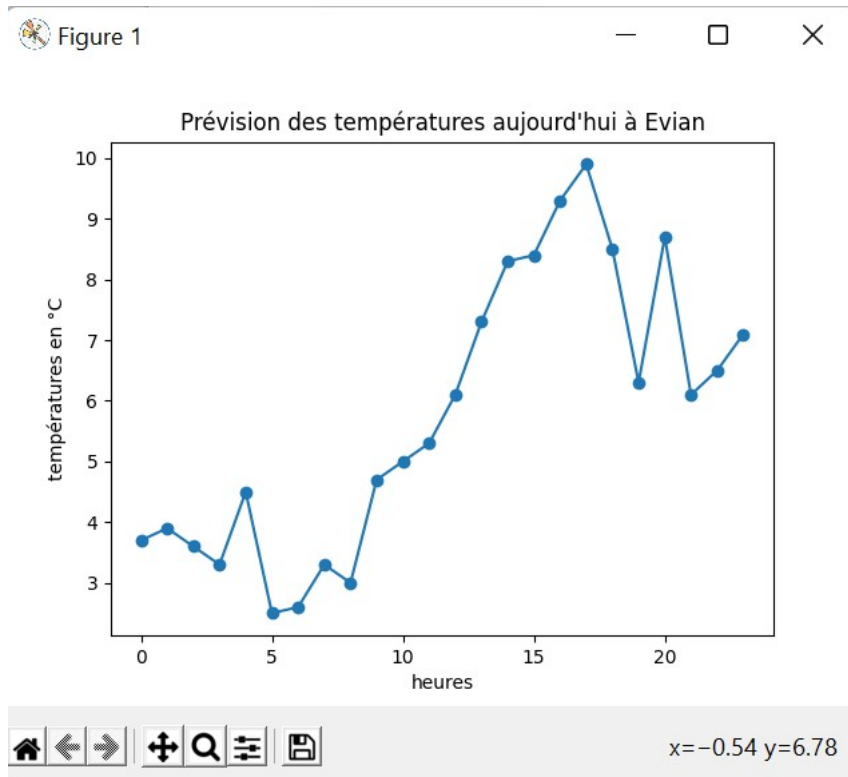
- Compléter la fonction **tp3()** en :
stockant les heures de 0 à 23 dans la liste **heures**
stockant les températures précédemment trouvées dans la liste **temperatures**
ajoutant les instructions suivantes dans la fonction.

```
plt.xlabel('heures')
plt.ylabel('températures en °C')
plt.title("Prévision des températures aujourd'hui à Evian")
```

```
plt.scatter(heures,temperatures)
plt.plot(heures,temperatures)
```

```
plt.show()
```

Voici le rendu attendu :



 **Lever la main pour valider ce TP.**

TP4 - Flask – Exploration

- Récupérer le dossier **TP4** et l'enregistrer dans le dossier **TP-Chapitre**.
- Lancer le programme **site.py** et noter l'adresse url sur lequel le serveur tourne.

- Lancer un navigateur à l'adresse url précédente.
- Cliquer sur le lien « cliquer pour continuer » et noter l'adresse url de la deuxième page.

- Modifier le fichier **site.py** pour que la seconde page renvoie un lien vers la première page

Voici le rendu attendu:

Nous sommes maintenant sur la deuxième page !
[Lien vers l'index](#)

 **Lever la main pour valider ce TP.**

TP5 - Flask – Ajout d'une route simple

- Dupliquer le dossier **TP4** et renommer le **TP5**.
- Ajouter une troisième route **"/troisieme"** qui appelle la fonction **p3** qui renvoie une troisième page accessible depuis l'index, qui elle aussi contiendra un lien de retour vers l'index.

Voici le rendu attendu pour la page index et la troisième page:

Le super site

Ce site est construit à partir du framework Flask.
[Lien vers la suite](#)
[Lien vers la troisième page](#)

Nous sommes maintenant sur la troisième page !
[Lien vers l'index](#)

 **Lever la main pour valider ce TP.**

TP6 - Flask – Ajout d'une route dynamique

- Dupliquer le dossier **TP5** et renommer le **TP6**.
- Modifier la route **"/troisieme"** et remplacer la par **"/troisieme/<name>"** afin d'avoir une route dynamique.
- Modifier la fonction **p3** en recopiant le code ci-dessous :

```
def p3(nom):  
    return "Nous sommes maintenant sur la page personnel de {name} ! <br>\n<a href = {lien}> Lien vers l'index</a>".format(lien=url_for('index'), name=nom)
```

Le paramètre **nom** sert à la construction de la page et **{name}** sera remplacé par le nom donné en paramètre.

- Remplacer dans la fonction **index**, **url_for('p3')** par **url_for('p3', nom="Zian")**
Celle-ci fera un lien la route **"/troisieme/Zian"** et appellera **"p3"** avec le paramètre **"Zian"**

Voici le rendu attendu:



 **Lever la main pour valider ce TP.**

TP7 - Flask – Template

- Récupérer le dossier **TP7** et l'enregistrer dans le dossier **TP-Chapitre**.
- Modifier le fichier **main.css** pour que les titres de niveau 1 soit en violet.
- Modifier le fichier **deuxieme.html** pour que le message «Vous êtes sur la page personnelle de <votre prénom> » soit remplacé par «Page perso de <votre prénom> »

Voici le rendu attendu :

Bienvenue

page perso de Zian

Vous êtes sur le supersite !

[Cliquer pour continuer](#)

[retourner à la page d'accueil](#)

 **Lever la main pour valider ce TP.**

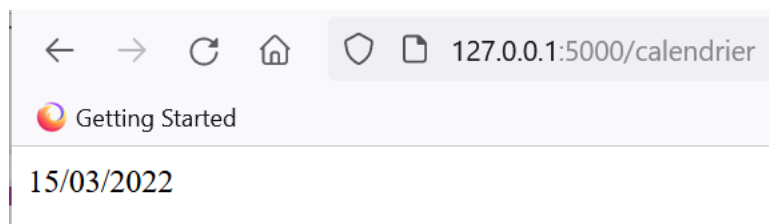
TP8 - Flask – Ajout d'une route

- Dans le fichier **site.py**, ajouter une route `"/calendrier"` qui appelle la fonction `date` suivante :

```
def date():  
    temps = datetime.now()  
    date_format = temps.strftime('%d/%m/%Y')  
    return date_format
```

et importer la librairie **datetime** en début de fichier avec :
`import datetime`

- Relancer le serveur et aller à l'url `...../calendrier`



 **Lever la main pour valider ce TP.**

TP9 - Flask – Ajout d'un template

- Dans le dossier **template**, faire une copie du fichier `deuxieme.html` et le renommer **calendrier.html**.
- Modifier le contenu du corps de la page (`<body>`) et le remplacer le par :

```
<h1 > Vous êtes sur le calendrier </h1>  
Aujourd'hui, nous sommes le {{texte_date}}.  
<a href="{{ url_for('index') }}"> retourner à la page d'accueil </a>
```

- Dans le fichier **site.py**, modifier la fonction **date** et remplacer

```
return date_format
```

par

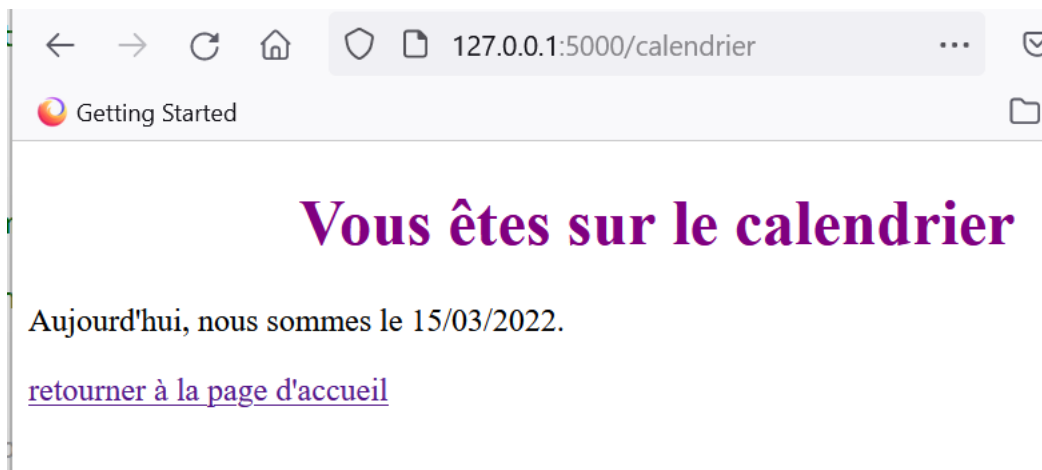
```
return render_template("calendrier.html", texte_date = date_format)
```

- Vérifier le fonctionnement.

Après avoir relancé le serveur, avec le navigateur aller à l'adresse `....../calendrier`

Cette route lance la fonction **date** qui récupère la date à l'aide de la librairie **datetime** renvoie le modèle de page **calendrier.html** avec le paramètre **texte_date**.

Voici le résultat attendu.



- Pour terminer, modifier le template **index.html** afin d'y ajouter un lien vers la page calendrier que l'on vient de créer.

Voici les 3 pages obtenus

Bienvenue

Vous êtes sur le supersite !

[Cliquer pour continuer](#)

[Lien vers la page calendrier](#)

Page perso de Zian

[retourner à la page d'accueil](#)

Vous êtes sur le calendrier

Aujourd'hui, nous sommes le 27/03/2022.

[retourner à la page d'accueil](#)



Lever la main pour valider ce TP.