

Mini projet

M. Tellene

Conseil d'organisation : il vous est conseillé pour ce mini-projet de créer un dossier dans votre dossier « recherche textuelle ».

Le but de ce mini-projet sera de reproduire le fonctionnement du `ctrl + f` pour recherche des mots dans une page web. Voici un exemple de rendu :

Voici toutes les occurrences de "amet"

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit.

Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis. Curabitur aliquet pellentesque diam. Integer quis metus vitae elit lobortis egestas. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vel erat non mauris convallis vehicula. Nulla et sapien. Integer tortor tellus, aliquam faucibus, convallis id, congue eu, quam. Mauris ullamcorper felis vitae erat. Proin feugiat, augue non elementum posuere, metus purus iaculis lectus, et tristique ligula justo vitae magna.

Aliquam convallis sollicitudin purus. Praesent aliquam, enim at fermentum mollis, ligula massa adipiscing nisl, ac euismod nibh nisl eu lectus. Fusce vulputate sem at sapien. Vivamus leo. Aliquam euismod libero eu enim. Nulla nec felis sed leo placerat imperdiet. Aenean suscipit nulla in justo. Suspendisse cursus rutrum augue. Nulla tincidunt tincidunt mi. Curabitur iaculis, lorem vel rhoncus faucibus, felis magna fermentum augue, et ultricies lacus lorem varius purus. Curabitur eu amet.

Pour ce faire, nous utiliserons un serveur via la librairie `flask` fourni par Python. Afin d'apprendre à manipuler avec cette librairie, nous allons faire un travail préliminaire : une page d'authentification.

1 Travail préliminaire

1. Créer un dossier « travail preliminaire »
2. Dans ce dossier créer un fichier `app.py` et un dossier « templates ». `app.py` contiendra le code Python de notre serveur. Le dossier « templates » contiendra nos pages HTML
3. Dans le dossier « templates », créer un fichier `index.html` et mettre le code suivant :

```

<html>
<head>
  <title>Mon site</title>
</head>

<body>
  <form action="formulaire" method="GET">
    <p>Nom :</p>
    <input type="text" name="nom">
    <p>Prénom :</p>
    <input type="text" name="pnom">
    <p>Age :</p>
    <input type="text" name="age">
    <br>
    <button type="submit">Valider</button>
  </form>
</body>
</html>

```

Ce code crée un page web avec un formulaire HTML¹. Ce formulaire envoie des données via la méthode GET, méthode qui écrit dans l'URL. Le bouton « Valider » permet de valider le formulaire et d'envoyer les données au serveur car il a `type="submit"`.

4. Aller maintenant dans `app.py` et copier le code suivant :

```

1 from flask import Flask, render_template, request
2
3 app = Flask(__name__)
4
5
6 @app.route('/')
7 def index():
8     """
9     charge la page index.html (accueil du site)
10    :return: index.html
11    """
12    return render_template("index.html")

```

5. Une fois fait, ouvrir un terminal ayant pour répertoire de travail le dossier « travail préliminaire » et lancer la commande

```
$ export FLASK_APP=app.py
```

puis la commande

```
$ flask run
```

1. Les formulaires HTML sont un des vecteurs principaux d'interaction entre un utilisateur et un site web ou une application. Ils permettent à l'utilisateur d'envoyer des données au site web. La plupart du temps, ces données sont envoyées à des serveurs web mais la page peut aussi les intercepter et les utiliser elle-même.

Une adresse devrait être donnée, aller sur un navigateur, taper l'adresse donnée et la page créée apparaîtra. Bien que cela fonctionne, si l'on essaye de soumettre le formulaire, une erreur se produit. Il faut donc récupérer les données envoyées et rediriger l'utilisateur.

6. Toujours dans `app.py`, ajouter le code suivant :

```
1 @app.route('/formulaire', methods=['GET'])
2 def formulaire():
3     nom = request.args.get("nom")
4     prenom = request.args.get("prenom")
5     age = request.args.get("age")
6     print(nom, prenom, age)
7     return render_template(
8         "resultat.html",
9         nom = nom,
10        prenom = prenom,
11        age = age
12    )
```

7. Une fois le code ajouté, relancer le serveur (cf. item 5) et rafraîchir votre page. Essayer de remplir le formulaire et le soumettre. Une nouvelle erreur se produit, mais si vous regardez le terminal, vous pouvez voir des données apparaître, en l'occurrence celles que vous avez renseigné. L'erreur de redirection vient du fait que la page `resultat.html` n'existe pas encore... créons la

Comprendre les arguments du `render_template` : le premier argument est la page HTML (qui doit être dans un dossier « templates ») vers laquelle l'utilisateur va être redirigé. Les arguments sont les variables que l'on va faire passer à la page HTML afin de pouvoir les utiliser pour de l'affichage par exemple. Il est à noter que de telles variables seront mises entre ... pour être utilisées.

8. Aller dans le dossier « templates », créer un nouveau fichier `resultat.html` et mettre le code suivant :

```

<html>
<head>
  <title>Mon site</title>
</head>

<body>
  <p>Bonjour {{ prenom }} {{ nom }}, vous avez {{ age }} ans !</p>

  <form action="courses" method="POST">
    <input type="checkbox" name="checkbox" value="radis"> Radis
    <br>
    <input type="checkbox" name="checkbox" value="salade"> Salade
    <br>
    <input type="checkbox" name="checkbox" value="tomate"> Tomate
    <br>
    <input type="checkbox" name="checkbox" value="pomme"> Pomme
    <br>
    <input type="checkbox" name="checkbox" value="poire"> Poire
    <br>
    <input type="checkbox" name="checkbox" value="ananas"> Ananas
    <br>
    <input type="checkbox" name="checkbox" value="oeuf"> Oeuf
    <br>
    <input type="checkbox" name="checkbox" value="beurre"> Beurre
    <br>
    <input type="checkbox" name="checkbox" value="lait"> Lait
    <br>
    <input type="checkbox" name="checkbox" value="creme"> Crème
    <br>
    <input type="checkbox" name="checkbox" value="jambon"> Jambon
    <br>
    <input type="checkbox" name="checkbox" value="gigot"> Gigot
    <br>
    <input type="checkbox" name="checkbox" value="steak"> Steak
    <br>
    <button type="submit">Valider ma liste</button>
  </form>
</body>
</html>

```

9. Créer un fichier `courses.html` et créer la fonction dans `app.py` permettant de récupérer les éléments cocher par l'utilisateur et de les afficher dans `courses.html`

2 Mini-projet

Maintenant que vous avez compris comment fonctionne un serveur `flask`, créer un nouveau dossier « projet » permettant de faire les choses suivantes :

- Créer une barre de recherche dans un fichier `index.html`, cette barre contiendra le motif

-
- `m` à chercher
- Dans ce même fichier, mettre un ou plusieurs paragraphe qui sera le texte `t`
 - Utiliser le fichier `app.py` pour récupérer le motif et le texte
 - Dans le `app.py`, lancer la recherche de Boyer-Moore et récupérer les indices des occurrences du motif dans le texte
 - Rediriger l'utilisateur vers une autre page où le motif est mis en évidence dans le texte (voir l'exemple donné au début)