Jinja2

Le moteur de patron

Présentation

Jinja est ce que l'on appelle un moteur de templates : patrons en français…).

- Les moteurs de templates fonctionnent quasiment tous de la même manière :
  - à partir d'un patron (template), on effectue le rendu en transmettant un contexte.
- Un patron est constitué de parties fixes et de parties variables
  - Jinja fournit même des boucles et des tests. Il est ainsi possible de générer différentes sorties avec un même patron en fonction du contexte.

Exemple

Juste pour comprendre le principe :

- J'utilise la fonction Template() de Jinja2 pour définir un patron :
  - « Bonjour » est la partie fixe
  - {{ nom }} est la partie variable
- La méthode render() permet d'obtenir un rendu à partir de mon patron

```
from jinja2 import Template

reponse = input("Votre nom : ")

tm = Template("Bonjour {{ nom }}")
texte = tm.render(nom=reponse)

print(texte)
```

```
Votre nom : Rafik
Bonjour Rafik
```

Jinja et le HTML

- En fait, Jinja a été créé pour la **génération de pages HTML**.
  - Dans un site web, beaucoup de choses reviennent de page en page comme l'en-tête, le pied et certains paramétrages.
  - Jinja permet de préparer l'équivalent d'un fond de page et de générer une page avec ce fond.
  - Cela impose une certaine structure dans les fichiers, car Jinja doit pouvoir retrouver les fichiers appelés par d'autres.

- Il est donc nécessaire de créer un répertoire (ex : tmpl) pour stocker les patrons.
  - Ces derniers seront des fichiers avec le suffixe **.jinja**, mais c'est juste pour bien les différencier, il n'y a pas d'obligation de ce côté.

Jinja et le HTML

Pour rentrer un peu plus dans le détail, voici le patron base.jinja.

- Il reprend la structure d'une page HTML et définit les balises 'html', 'head' et 'body'.

```
<!doctype html>
<html lang="fr">

<head>
    <title>Titre de la page</title>
</head>

<body>
    <h1>Le nom entré est {{ nom }}
    <footer>
    {{ time_stamp }}
    </footer>
</body>
</html>
```

- Le patron est définis, il ne reste plus qu'à le passer dans la moulinette du moteur Jinja pour obtenir une page HTML.

Jinja et le HTML

```python
from jinja2 import Environment, FileSystemLoader

## Necessaire pour les dates en français
import datetime

TMPL_DIR = "tmpl"
fichier = "base.jinja"

reponse = input("Votre nom : ")

templateLoader = FileSystemLoader(searchpath=TMPL_DIR)
templateEnv = Environment(loader=templateLoader)
template = templateEnv.get_template(fichier)

data={
        'nom':reponse,
        'time_stamp':datetime.datetime.now().strftime("%x %X")
    }

print( template.render( data ) )
```

```
Votre nom : Rafik
```

```
<!doctype html>
<html lang="fr">

<head>
        <title>Titre de la page</title>
</head>

<body>
    <h1>Le nom entré est : Rafik</h1>
        <footer>
        09/14/20 00:21:47
        </footer>

</body>
</html>
```

Jinja et le HTML

- **FileSystemLoader()** : méthode qui permet de sélectionner avec l'argument « searchpath » le répertoire ou chercher notre patron.
- **Environment()** : définition de l'environnement de Jinja en lui demandant de lire le répertoire indiqué précédemment.
- **templateEnv.get_template()** : permet de lire le patron sur le quel on souhaite travailler.
- Avec **data** on initialise les variables du contexte dans un dictionnaire que l'on transmettra pour effectuer le rendu.
- **template.render()** : production du rendue avec pour argument les variable que l'on souhaite transmettre.

Jinja et le HTML

- On crée dans le même répertoire que son programme python un fichier « index.html » vide, dans le quel on va venir écrire sont code html.

```
outputText = template.render( data )
html_file = open('index.html', 'w')
html_file.write(outputText)
html_file.close()
```