

Moteur de template Jinja

Moteur de template - c'est quoi ?

Un moteur de template est un outil qui simplifie la syntaxe pour assurer une bonne maintenabilité de son projet web. <https://pypi.org/project/Jinja2/>

Il permet de dissocier la partie présentation (HTML) de la partie programmation (PYTHON, PHP, JS...).

```
>>> pip install jinja2
```

Moteur de template - utilisation basique

```
#!/usr/bin/env python3
# coding: utf-8
import cgi
from jinja2 import Template # on importe le module de génération de template

template = Template(""" # on créé le template html
    <!DOCTYPE html>
    <head>
        <title>Mon site</title>
    </head>
    <body>
        <h1>Hello World</h1>
        <h2>Bonjour {{ name }}</h2> # jinja permet l'utilisation de variable
    </body>
</html>""")

output = template.render(name='John Doe') # on définit la variable à passer au template

print("Content-type: text/html; charset=utf-8\n")
print(output)
```

Jinja - exercice 1 (page0.py)

À partir de votre serveur web, créer une nouvelle page.

- Cette page doit pouvoir générer de l'HTML grâce à la librairie "jinja"
- Passer un nom en paramètre dans votre template HTML, pour afficher "Bonjour <votre_nom>"
- Passer une liste contenant plusieurs items et afficher tous les éléments de cette liste au format :

```
<ul>
```

```
    <li>item 1</li>
```

```
    <li>item 2</li>
```

```
    ...
```

```
</ul>
```

Jinja - exercice 2 *(page1.py)*

À partir de votre serveur web, créer une nouvelle page.

- Cette page doit pouvoir générer de l'HTML grâce à la librairie "jinja"
- Ajouter une condition à l'intérieur de votre template html :
Si le paramètre == 'toto', afficher la phrase "Toto n'est pas votre vrai prénom :)"
- Sinon afficher la phrase "Bonjour <param>"

Moteur de template - une meilleure implémentation

Jinja permet d'utiliser directement des templates au format html.

- Commencez par créer un dossier “templates” à la racine de votre projet
- Créez un template “content.html” dans ce dossier et insérez-y du code html
- Créez un module python qui va utiliser ce template :

```
#!/usr/bin/env python3
# coding: utf-8
import cgi
from jinja2 import Environment, FileSystemLoader

file_loader = FileSystemLoader('templates')
env = Environment(loader=file_loader)

template = env.get_template('content.html')
output = template.render(name='Samuel')

print("Content-type: text/html; charset=utf-8\n")
print(output)
```

dossier où se trouvent les templates html à utiliser
on crée un environnement avec ce dossier

on définit le template que l'on souhaite utiliser
on génère le template, en passant optionnellement des variables

Moteur de template - exemple de langage de template

layout.html

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Title</title>
  </head>

  <body>
    {% block content %}{% endblock %}
  </body>

</html>
```

content.html

```
{% extends "layout.html" %}

{% block content %}
  <p>Ceci est un test d'ajout de contenu dans une page</p>

  <p>
    {% if name == "sam" %}
      Bonjour {{ name }} !
    {% else %}
      Salut {{ name }} !
    {% endif %}
  </p>

  <p>Hello {{ truc }}</p>
{% endblock %}
```

Jinja - exercice 3 *(page2.py)*

À partir de votre serveur web,

- utiliser le module de rendu de template jinja (utiliser le module `template_render.py` fourni)
- créer un dossier “templates” dans lequel vous ajouterez les fichier .html
- créer un nouveau script python qui fait appel à un fichier “content.html”.
- passer des paramètres à ce template

Jinja - exercice 3 *(page_liste.py)*

À partir de votre serveur web,

- utiliser le module de rendu de template jinja (utiliser le module `template_render.py` fourni)
- dans votre dossier templates, ajouter un nouveau fichier `liste.html`
- créer un nouveau script python qui fait appel à “`liste.html`”
- passer une liste en paramètre à ce template