

Ressource – Le module Flask

Objectif :

- Créer une application web client-serveur grâce au module Python Flask

Principe sans page html

A l'aide d'un exemple :

import flask # On importe le module après l'avoir installé

app = flask.Flask(__name__) # Crée l'objet application Flask, qui contient les données de l'application et les méthodes correspondant aux actions susceptibles d'être effectuées sur l'objet.

app.config["DEBUG"] = True # lance le débogueur, ce qui permet d'afficher un message autre que « Bad Gateway » s'il y a une erreur dans l'application.

@app.route('/', methods=['GET']) # indique à Flask que la fonction home correspond au chemin /.
 # HTTP (HyperText Transfert Protocol) : c'est le principal moyen de communiquer de l'information sur Internet.
 HTTP implémente un certain nombre de « méthodes » qui disent dans quelle direction les données doivent se déplacer et ce qui doit en être fait. Les deux plus fréquentes sont GET, qui récupère les données à partir d'un serveur, et POST, qui envoie de nouvelles données vers un serveur.

```
def home(): # affichage du code html ici ou autre
    return "<h1>Le monde des oiseaux est magique</h1><br>list_latin = ['Vanellus vanelli', 'Ardea cinerea', 'Carduelis carduelis']\n\n<br>tuple_francais = ('Vanneau huppé', 'Héron Cendré', 'Chardonneret élégant')\n\n<br>print(list_latin)\n\n<br>print(tuple_francais)\n\n<br>print(list_latin[0])\n\n<br>list_latin[0] = 'Vanellus vanellus' # Modification du premier élément de la liste\n\n<br>print(list_latin)\n\n<br>print(tuple_francais[1])\n\n<br># Modification du premier élément du tuple\n\n<br>#tuple_francais[1] = 'Vanneau pas huppé'\n\nprint(tuple_francais)\n\n<br>print('Poids en octets du tuple : ',tuple_francais.__sizeof__()) #taille de la variable, occupation en mémoire\n\n<br>print('Poids en octets de la liste : ',list_latin.__sizeof__())"
```

App.run() # lancement de l'application sur le port 5000 (Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

Avec une page html

Il faut indiquer à Flask des templates de pages et des styles prédéfinis pour l'application. La structure du dossier et le nom des dossiers static et templates sont importants : Flask sait que c'est dans ces dossiers là qu'il doit aller chercher les éléments qui l'intéressent:

- un dossier *static* qui contient la feuille de style css
- un dossier *templates* qui contient une page html

from flask import Flask, render_template # pour pouvoir appliquer un template

app = Flask(__name__)

```

app.config["DEBUG"] = True

list_latin = ['Vanellus vanelli', 'Ardea cinerea', 'Carduelis carduelis']

tuple_francais = ('Vanneau huppé', 'Héron Cendré', 'Chardonneret élégant')

@app.route("/")
def home():
    return render_template("home.html", message = "le dernier oiseau "
                        + "que j'ai photographié est un " + tuple_francais[-1])

# la chaîne de caractère message pourra être affichée dans le template home.html

app.run()

```

home.html

```

</doctype html>
<html lang="fr">
    <head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    </head>
    <body>
        <h1 > {{ message }}</h1>
    </body>
</html>

```

main.css

```

h1 {
    font-size: 2em;
    color: green;
    text-align: center;
}

table {
    text-align: center;
}

```

Test conditionnel

flask1.py

```

from flask import Flask, render_template

app = Flask(__name__)

```

```
app.config["DEBUG"] = True
```

```
list_latin = ['Vanellus vanelli', 'Ardea cinerea', 'Carduelis carduelis']
tuple_francais = ('Vanneau huppé', 'Héron Cendré', 'Chardonneret élégant')
```

```
@app.route("/")
def home():
    return render_template("home.html", message = "le dernier oiseau "
        + "que j'ai photographié est un "
        + tuple_francais[-1], nb = len(tuple_francais))

app.run()
```

home.html

```
<!doctype html>
<html lang="fr">
    <head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    </head>
    <body>
        <h1> {{ message }} </h1>
        <p> Mon humeur est : {% if nb > 5 %} 😊 {% else %} ☹️ {% endif %} </p>
    </body>
</html>
```

Boucle for

flask1.py

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

```
app = Flask(__name__)
```

```
app.config["DEBUG"] = True
```

```
list_latin = ['Vanellus vanelli', 'Ardea cinerea', 'Carduelis carduelis']
tuple_francais = ('Vanneau huppé', 'Héron Cendré', 'Chardonneret élégant')
```

```

@app.route("/")
def home():
    return render_template("home.html", message = "le dernier oiseau "
        + "que j'ai photographié est un "
        + tuple_francais[-1], nb = len(tuple_francais) , franc = tuple_francais)

app.run()

```

home.html

```

<!doctype html>
<html lang="fr">
    <head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    </head>
    <body>
        <h1> {{ message }} </h1>
        <p> Mon humeur est : {% if nb > 5 %} 😊 {% else %} ☹️ {% endif %} </p>
        <h2>Liste des oiseaux :</h2>
        <ul>
            {% for f in franc %}
                <li>{{ f }}</li>
            {% endfor %}
        </ul>
    </body>
</html>

```

Avec deux pages html

Flask2.py

```

from flask import Flask, render_template

app = Flask(__name__)

app.config["DEBUG"] = True

list_latin = ['Vanellus vanelli', 'Ardea cinerea', 'Carduelis carduelis']
tuple_francais = ('Vanneau huppé', 'Héron Cendré', 'Chardonneret élégant')

```

```
@app.route("/")
def home():
    return render_template("home.html", message = "le dernier oiseau "
        + "que j'ai photographié est un "
        + tuple_francais[-1], nb = len(tuple_francais),
        franc = tuple_francais)

@app.route("/next")
def suite():
    return render_template("page_suivante.html", nb = len(tuple_francais))

app.run()
```

home.html

```
<!doctype html>
<html lang="fr">
    <head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
    </head>
    <body>
        <h1> {{ message }} </h1>
        <p> Mon humeur est : {% if nb > 5 %} 😊 {% else %} ☹️ {% endif %} </p>
        <h2> Liste des oiseaux : </h2>
        <ul>
            {% for f in franc %}
                <li> {{ f }} </li>
            {% endfor %}
        </ul>
        <a href="{{ url_for('suite') }}"> Cliquer pour continuer </a>
    </body>
</html>
```

page_suivante.html

```
<!doctype html>
<html lang="en">
    <head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">
```

```
</head>
```

```
<body>
```

```
    <h1> Vous êtes sur la 2eme page </h1>
```

```
    <p> Mon humeur est : {% if nb < 5 %} 😊 {% else %} 😞 {% endif %}</p>
```

```
    <a href="{% url_for('home') %}"> retourner à la page d'accueil </a>
```

```
</body>
```

```
</html>
```