

Activité Flask : utiliser les cases à cocher (checkbox)

Présentation

Une case à cocher est **associée à chaque tâche** pour indiquer si elle est **terminée ou pas** :

- en cochant la case à cocher, la tâche est considérée comme terminée ;
- une **information supplémentaire** est utilisée dans le tableau associatif pour connaître l'état de la tâche (terminée ou nonterminée) ;
- les **tâches terminées** seront affichée **en dessous** des autres tâches.

Exemple d'informations pour une tâche :

- **id** = 5
- **libelle** = "Préparer mon sac"
- **accomplie** = Non

Une tâche est soit terminée ou pas. Il s'agit d'une **information binaire** qui est représentée en Python par une variable booléenne : **False** ou **True**.

Dans le **dictionnaire** (tableau associatif) une tâche est représentée de la manière suivante :

- `{"id":5, "libelle": "Préparer mon sac", "accomplie": False}`

Le fichier de l'application

app.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

from flask import Flask, render_template, request, redirect, url_for
app = Flask(__name__)

#taches : dictionnaire (tableau associatif) de tâches
taches=[]
#id : identifiant de la tache
id = 1

@app.route('/')
def index():
    global taches
    # fournir au template index.html le tableau taches accomplie et non accomplie
    accomplies = []
    nonaccomplies = []
    for tache in taches:
        if tache["accomplie"] == False:
            nonaccomplies.append(tache)
        else:
            accomplies.append(tache)
    return render_template('index.html', accomplies=accomplies, nonaccomplies=nonaccomplies)

@app.route('/ajout', methods=['POST'])
def ajout():
    # utiliser les variables globales taches et id
    global taches, id
    vartache = request.form['tache']
    taches.append({"id":id, "libelle":vartache, "accomplie":False})
    #incrémenter id pour la prochaine tâche
    id = id + 1
    return redirect(url_for('index'))
```

```

@app.route('/miseajour', methods=['POST'])
def miseajour():
    # utiliser les variables globales taches et id
    global taches, id
    varid = request.form['choix']
    # mise à jour de la tâche
    for tache in taches:
        if tache["id"] == int(varid):
            tache["accomplie"] = True
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(debug=True, port=5005)

```

La page HTML

index.html

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Application Todo</title>
</head>
<body>
    <h1>Todo liste</h1>
    <div>Ajouter une nouvelle tâche :
        <form action="/ajout" method="POST">
            <input type="text" name="tache">
            <input type="submit"></div>
        </form>
    </div>
    <div>
        <h2>Tâches non accomplies</h2>
        <form action="/miseajour" method="POST">
            <ul>
                {% for tache in nonaccomplies %}
                    <li><input type="checkbox" name="choix" value="{{tache.id}}>{{tache.libelle}} </li>
                {% endfor %}
            </ul>
            <input type="submit" value="Mettre à jour"></div>
        </form>
        <h2>Tâches accomplies</h2>
        <ul>
            {% for tache in accomplies %}
                <li>{{tache.libelle}}</li>
            {% endfor %}
        </ul>
    </div>
</body>
</html>

```

Explications

- un **nouveau formulaire** permet de gérer les cases à cocher avec un **nouveau bouton** ;
- le **nom** des cases à cocher est le même **choix**.
- La **valeur** de chaque case à cocher sera la valeur de l'**ID** de la tâche

La gestion des cases à cocher

Quand une case à cocher est **cochée**, l'information **est transmise** par le formulaire de la page HTML à la vue du programme serveur. L'information est récupérée de la manière suivante :

```
vartache = request.form['tache']
```

Mais si la case à cocher n'est **pas cochée**, l'information **n'est pas transmise** par le formulaire de la page HTML. Il faut donc s'assurer que l'information n'a pas été reçue par la vue du programme serveur :

```
vartache = 'existe' if 'tache' in request.form else 'n'existe pas'
```

Retour à l'activité : Utiliser le framework Flask pour créer un site Web dynamique

- Activité : Utiliser le framework Flask pour créer un site Web dynamique

From:
[/- Les cours du BTS SIO](#)

Permanent link:
/doku.php/isn/flask_todo05?rev=1576160005

Last update: **2019/12/12 15:13**

