

INF3005 – Programmation web avancée

Introduction à Flask

Jacques Berger

Objectifs

Servir des pages statiques

Utiliser un formulaire

Prérequis

Python

HTML

HTTP

HTTP

Anatomie d'une URL

Uniform Resource Locator

Protocole

Nom de domaine

Port

Route

Query parameters

HTTP

Méthodes

GET : Lire une ressource

POST : Créer une nouvelle ressource

Flask

Un microframework simplifiant l'utilisation d'HTTP

Installation

```
pip install Flask
```

Flask

Une fois installé, l'exécutable flask est disponible pour lancer l'application

Voir un Hello World!

Flask

Définir la variable d'environnement `FLASK_APP`
avant de lancer l'application

Flask

Les décorateurs permettent de spécifier les routes et les fonctions à exécuter pour chacune d'elles

La valeur de retour de la fonction sera envoyée au navigateur

Templating

Plutôt que de générer le HTML avec des chaînes de caractères, on place le HTML dans des templates

Avec Flask, le moteur de templating est Jinja2

On place les templates dans le répertoire templates

Templating

On appelle un template avec la fonction
`render_template`

Voir un exemple

Static

Les éléments statiques d'un site sont les ressources qui ne changent jamais

CSS

Images

Pages HTML statiques

Static

On place les éléments statiques dans un répertoire static et on configure l'application pour utiliser ce répertoire statique

Voir exemple

Redirection

Une route peut faire une redirection avec la fonction `redirect`

Request

Avec un objet request, on peut accéder aux données que le navigateur envoie au serveur

```
request.form['password']
```

POST

On peut également spécifier la méthode HTTP à utiliser sur une route

```
@app.route('/envoyer', methods=['POST'])
```


Post-redirect-get

Voir un exemple

Plus loin...

Flask

<http://flask.pocoo.org/>