

INF5190 - MVC - Vue et contrôleur

Jean-Philippe Caissy

25 septembre 2019

Vue

- ▶ Représentation de l'information retourné par le contrôleur
- ▶ Possibilité d'avoir plusieurs vues pour la même représentation des données
 - ▶ Exemple : HTML et JSON

Vue

Engin de templating

- ▶ Utiliser une méthode Python pour construire la vue n'est pas efficace :
 - ▶ Difficile à tester
 - ▶ Code difficile à réutiliser
 - ▶ Manque de flexibilité
 - ▶ Peut introduire des failles de sécurité

Solution : utiliser un outil de génération de pages HTML!

Vue

Engin de templating

Jinja2

Jinja2 est un engin de templating HTML

Avantages :

- ▶ Compilation des templates dans un bac à sable (sandbox)
- ▶ Escaping des entrées HTML automatique pour prévenir des attaques de type cross site scripting (XSS)
- ▶ Héritage de thèmes : permet de réutiliser des pages HTML
- ▶ Syntaxe configurable et extensible
- ▶ Facile à débbugger

Vue

Engin de templating

Jinja2

```
>>> from jinja2 import Template
>>> template = Template('Hello World, {{ name }}!')
>>> template.render(name='John Doe')
'Hello World, John Doe!'
```

Vue

Engin de templating

Jinja2

Syntaxe de base :

- ▶ `{{ variable }}` Pour afficher une variable
- ▶ `{% %}` Bloc d'exécution
 - ▶ `{% if i > 10 %} [...] {% endif %}`
 - ▶ `{% for item in navigation %}`
 ``
 `{{ item.name }}`
 ``
 `{% endfor %}`

Vue

Engin de templating

Jinja2

Filtres

On peut appliquer des filtres sur des variables

- ▶ `{{ variable|lower }}` pour afficher variable en minuscules
- ▶ `{{ 123.5321|round }}` retourne 123
- ▶ etc ...

Liste des filtres inclus

Vue

Engin de templating

Jinja2

Fichier view.html

```
{% extends "layout.html" %}  
{% block body %}  
    <h1>Hello World, {{ user }}!</h1>  
{% endblock %}
```

Fichier layout.html

```
<html>  
    <body>  
        {% block body %}{% endblock %}  
    </body>  
</html>
```


Contrôleur

Internationalization

TODO

Contrôleur

Localisation

TODO

Contrôleur

Authentication

TODO