

# Templates

Le moteur de templates est responsable de l'affichage des vues. Il est conseillé d'utiliser Twig, plutôt que PHP, comme moteur de templates.

## Installation

```
composer require twig
```

## Éléments syntaxiques

Twig utilise 3 types de tags :

<b>{{ ... }}</b>	“affiche quelque chose” équivalent du echo de php : affiche une variable ou le résultat d'une expression.
<b>{% ... %}</b>	“Fait quelque chose” : tag agissant sur la logique du template ; il est utilisé par exemple pour faire des itérations, ou des contrôles.
<b>{# ... #}</b>	“Commente quelque chose” : équivalent du PHP /* comment */. Il est utilisé pour ajouter des commentaires sur 1 ou plusieurs lignes. Le commentaire ne sera pas présent dans la page réponse.

Twig contient aussi des filtres (filters), qui permettent de modifier le contenu affiché.

L'appel suivant affiche la variable **title** en majuscules :

```
{{ title|upper }}
```

## Tags

La liste complète des tags est fournie dans la [documentation twig](#)

### Afficher {{...}}

Afficher une variable	Nom : <code>{{nom}}</code>
Afficher un élément de tableau	Utilisateur en position 0 : <code>{{users[0]}}</code>
Accès aux membres (utilisation implicite du getter <code>getNom()</code> )	Nom d'utilisateur : <code>{{user.nom}}</code>
Application d'un filtre	Pseudo en majuscules : <code>{{ pseudo upper }}</code>
Utilisation d'une combinaison de filtres	Message : <code>{{ news.texte striptags title }}</code>
Affichage d'une date de type <code>DateTime</code>	Date : <code>{{ date date('d/m/Y') }}</code>
Concaténation	Identité : <code>{{ nom ~ ' ' ~ prenom }}</code>

### Filtres

<b>upper</b>	Met toutes les lettres en majuscules.	<code>{{ var upper }}</code>
--------------	---------------------------------------	------------------------------

<b>striptags</b>	Supprime toutes les balises XML.	<code>{{var striptags}}</code>
<b>date</b>	Formate la date selon le format donné en argument. La variable en entrée doit être une instance de DateTime.	<code>{{ date date('d/m/Y') }}</code>
<b>format</b>	Insère des variables dans un texte, équivalent à printf.	<code>{{ "Il y a %s pommes et %s poires" format(153, nb_paires) }}</code>
<b>length</b>	Retourne le nombre d'éléments du tableau, ou le nombre de caractères d'une chaîne.	Longueur de la variable : <code>{{ texte length }}</code>
<b>raw</b>	Désactive l'auto-escape sur l'affichage de la variable.	Élément HTML : <code>{{ element raw }}</code>

## Faire `{{#...}}`

### if

voir [Doc twig if](#)

Permet de poser une condition, comme en php

```
{% if online == false %}
    <p>Our website is in maintenance mode. Please, come back later.</p>
{% endif %}
```

### for

voir [Doc twig for](#)

Equivalent du foreach php :

```
<h1>Members</h1>
<ul>
    {% for user in users %}
        <li>{{ user.username|e }}</li>
    {% endfor %}
</ul>
```

## Fonctions utiles

### attribute

Permet d'accéder au membre d'un objet :

```
{{ attribute(object, method) }}
{{ attribute(object, method, arguments) }}
{{ attribute(array, item) }}
```

**isDefined**

Test l'affectation d'une variable (isset de php)

```
{% if var is defined %} ... {% endif %}
```

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/framework-web/symfony/templates?rev=1517253824>

Last update: **2019/08/31 14:43**

