

# 5 minutes

## Les fonctions Twig

### Les fonctions Twig

Les fonctions Twig permettent de prendre un ou plusieurs arguments et de retourner une valeur.

Il existe des fonctions Twig permettant d'utiliser des fonctions PHP dans les templates Twig, et d'autres fonctions plus spécifiques à Symfony.

### Les fonctions permettant d'utiliser des fonctions PHP

Nous allons commencer par voir les principales fonctions permettant d'utiliser des fonctions PHP.

#### La fonction date()

La fonction date() permet de convertir l'argument passé en date pour permettre une comparaison.

Si aucun argument n'est passé, la fonction retourne la date courante.

```
{% if date(user.last_connexion) < date('-1 week') %}  
    Cela ne fait pas longtemps !  
{% endif %}
```

Copier

#### Les fonctions max() et min()

La fonction max() permet de retourner la plus grande valeur d'une séquence d'éléments.

La fonction min() permet de retourner la plus petite valeur d'une séquence d'éléments.

```
{# Affiche 3 #}  
{{ max(1, 3, 2) }}  
  
{# Affiche 4 #}  
{{ min([4, 5, 6]) }}
```

Copier

#### La fonction random()

La fonction random() permet de retourner aléatoirement un élément : soit un des éléments d'une séquence, soit un caractère d'une chaîne, soit un entier aléatoire (avec optionnellement une limite inférieure et supérieure incluse).

```
{# Affiche par exemple -81 #}  
{{ random(-100, 10) }}  
  
{# Affiche par exemple 7 #}  
{{ random(0, 10) }}  
  
{# Affiche par exemple 25 #}  
{{ random([2, 42, 25, 8]) }}
```

Copier

## La fonction range()

La fonction range() permet de retourner une suite dans un tableau en utilisant une intervalle :

```
{# Affiche 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, #}
{% for n in range(0, 10) %}
  {{ n }},
{% endfor %}
```

Copier

Nous verrons d'autres fonctions par la suite.

## Les fonctions spécifiques à Symfony

Nous allons voir maintenant les fonctions asset(), path() et url() puis nous verrons les autres quand nous aurons plus avancé dans le cours.

### La fonction path()

**La fonction path() permet de générer automatiquement une URL en utilisant la configuration du Routing Symfony.**

C'est extrêmement utile pour générer des URLs dynamiques dans les templates Twig, aussi n'hésitez pas à lire plusieurs fois.

Supposons que nous ayons un UserController.php :

```
<?php
```

```
namespace App\Controller;
```

```
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
```

```
#[Route(name: 'user_')]
class UserController extends AbstractController
{
    const users = [
        ['id' => 12, 'name' => 'Paul'],
        ['id' => 14, 'name' => 'Jean'],
        ['id' => 22, 'name' => 'Marie'],
        ['id' => 1, 'name' => 'Sophie'],
    ];

    #[Route(path: '/', name: 'index')]
    public function index(): Response
    {
        return $this->render('index.html.twig', ['users' => self::users]);
    }

    #[Route('/{id}', name: 'profile')]
    public function profile(string $id): Response
    {
        $userIndex = array_search($id, array_column(self::users, 'id'));
        if ($userIndex === false) {
            throw $this->createNotFoundException('L\'utilisateur n\'existe pas !');
        }
    }
}
```

```

    }
    return $this->render('user_profile.html.twig', ['user' => self::users[$userIndex]]);
}
}

```

Copier

Nous avons deux routes : `user_index` sur le chemin `/` et `user_profile` sur le chemin `/id`.

*Si vous ne comprenez pas la recherche d'élément dans un tableau multidimensionnel PHP, relisez le chapitre sur les tableaux dans la formation PHP.*

La première route rend le template `index.html.twig` :

```

<a href="{{ path('user_index') }}">Accueil</a>

{% for user in users %}
<h2>
    <a href="{{ path('user_profile', {id: user.id}) }}">{{ user.name }}</a>
</h2>
{% endfor %}

```

Copier

Notez bien l'utilisation de la fonction `path()`.

**Nous lui passons le nom de la route qui doit être générée en premier argument, et en second argument la valeur dynamique du paramètre à utiliser.**

C'est vraiment un cas très courant d'utilisation de cette fonction. Essayez de reproduire l'exemple pour bien comprendre.

La seconde route, rend le template `user_profile.html.twig` avec l'id utilisé en paramètre :

```

<a href="{{ path('user_index') }}">Accueil</a>

{% if user %}
<p>{{ user.id }}
:
    {{ user.name }}</p>
{% endif %}

```

Copier

## La fonction `url()`

La fonction `url()` fonctionne comme la fonction `path()` mais retourne un chemin absolu au lieu d'un chemin relatif.

Dans la majorité des cas ce n'est pas nécessaire.

## La fonction `asset()`

**La fonction `asset()` permet de générer des URLs pour les assets (images, css, scripts JavaScript etc).**

Il faut installer une dépendance pour pouvoir l'utiliser :

```
composer require symfony/asset
```

Copier

La fonction construit l'URL à partir du dossier public.

Par exemple, pour une image située dans public/images/image1.png :

```

```

Copier