



ADRAR\o/ PÔLE NUMÉRIQUE

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > TRANSFORMATION NUMÉRIQUE DES ENTREPRISES

www.adrar-numerique.com

Twig est un **moteur de templates**.

Il permet la séparation du code PHP (au niveau des contrôleurs) et du code HTML (au niveau des vues).

- ❖ Il simplifie l'affichage des données et rend nos pages plus lisibles.
- ❖ Le système d'héritage est très utile pour gérer nos vues comme on le souhaite.

L'**héritage** est une des clés de Twig.

Un fichier de base (base.html.twig) est créé à la racine du dossier **templates**. Tout ce qui se trouve dans ce fichier sera présent dans tous les fichiers qui en hérite à l'aide de la ligne:

```
{% extends "base.html.twig" %}
```

test.html.twig

Bien entendu vous pouvez hériter de n'importe quel fichier twig.
Tout ce qui peut être répété dans plusieurs pages devrait être hérité.

Comment ça fonctionne ?

Twig

Nous allons regarder le contenu de *base.html.twig* et le décortiquer.

```
<!DOCTYPE html>
```

base.html.twig

```
<html lang="fr">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>{% block title %}{% endblock %}</title>
```

```
{% block css %}{% endblock %}
```

```
</head>
```

```
<body>
```

```
<p>Texte provenant de la page base.html.twig</p>
```

```
{% block body %}{% endblock %}
```

```
{% block js %}{% endblock %}
```

```
</body>
```

```
</html>
```

Le contenu de ces balises sera remplacé par les titres spécifiés dans chaque page

Le contenu de ces balises sera remplacé par les liens vers nos feuilles de style CSS

C'est ici que le contenu de la page courante sera chargé. Nous pouvons bien entendu mettre du code avant/après pour qu'il apparaisse partout

Ici ça sera nos liens vers nos fichiers de script JS qui seront chargés

Comment ça fonctionne ?

Twig

Nous allons regarder le contenu de *index.html.twig* et le décortiquer.

```
{% extends "base.html.twig" %}
```

On hérite de la base

index.html.twig

```
{% block title %}Titre de ma page{% endblock %}
```

Le contenu de ces balises sera remplacé par les titres spécifiés dans chaque page

```
{% block css %}
```

```
<link rel="stylesheet" href="{{ asset('css/lien/vers/la/feuille.css') }}" />
```

```
{% endblock %}
```

Le contenu de ces balises sera remplacé par les liens vers nos feuilles de style CSS

```
{% block body %}
```

```
<h1>Titre index</h1>
```

```
<p>Vue provenant de la page index.html.twig</p>
```

```
{% endblock %}
```

C'est ici que le contenu de la page courante sera chargé. Nous pouvons bien entendu mettre du code avant/après pour qu'il apparaisse partout

```
{% block js %}
```

```
<script src="{{ asset('js/lien/vers/le/script.js') }}"></script>
```

```
{% endblock %}
```

Ici ça sera nos liens vers nos fichiers de script JS qui seront chargés



Voilà le résultat sur pour la page *index.html.twig*

Afficher une variable	<code>{{ variable }}</code>
Afficher l'attribut d'un objet	<code>{{ objet.attribut }}</code>
Déclarer un bloc	<pre>{% block nom_du_bloc %} {# Ici on placera notre code HTML #} {% endblock %}</pre>
Conditions	<pre>{% if ... %} {% elseif ... %} {% else %} {% endif %}</pre>
Boucle	<pre>{% for element in liste %} {# Code #} {% endfor %}</pre>

Concaténation	<pre>{{ variable_a ~ variable_b }}</pre>
Formater une date	<pre>{{ utilisateur_date_inscription date("d/m/Y") }}</pre> BDD: 2023-02-15 Twig: 15/02/2023
Permettre l'affichage de données brutes sans échappement	<pre>{{ donnees raw }}</pre>
Autoriser l'interprétation de balises HTML	<pre>{{ article striptags("<p>") raw }}</pre> <p>Ici nous autorisons l'interprétation des balises <code></code> et <code><p></p></code>.</p>
Formatage de nombres	<pre> {{ -9800.333 number_format(2, '.', ',') }} {# outputs : -9 #} {{ (-9800.333) number_format(2, '.', ',') }} {# outputs : -9,800.33 #} {{ 1 + 0.2 number_format(2) }} {# outputs : 1.2 #} {{ (1 + 0.2) number_format(2) }} {# outputs : 1.20 #} </pre>

Pour appeler une route par exemple dans une barre de navigation, nous utiliserons cette syntaxe (`path("nom_route", { 'param': 'value' })`), les paramètres ne sont pas obligatoires):

```
<a href="{{ path('show_account', {'param': 'value'}) }}">Mon compte</a>
```

Pour charger une image, vous devrez utiliser celle-ci:

```

```

La fonction `asset` va rechercher dans le dossier *public*, à la racine du projet.

Les variables twig peuvent être suivies d'un pipe (`|`) pour préciser certaines choses (la date, le type de texte, ...).