

Twig

Twig est un **moteur de template**, c'est-à-dire un ensemble de fonctions permettant de générer ou produire du code HTML à partir de données. Le principe est de séparer les **gabarits** ou **patrons** qui sont des modèles de pages et les **contenus** de ces pages :

- Un **gabarit (template)** en anglais) est une page avec des *trous*
 - C'est-à-dire un fichier HTML contenant toutes les balises nécessaires, mais dont les contenus sont remplacés par des blocs variables
- Les contenus sont les éléments variables des pages placés dans des **variables** formant un **jeu de données**
 - Les contenus sont récupérés depuis des sources de données variés (fichiers, bases de données, sources externes)
- Twig fusionne un gabarit et un jeu de données pour **produire** efficacement une page HTML

Twig est le moteur de template de Symfony.

Il existe d'autres moteurs de template comme : Smarty (PHP historique), Blade (Laravel), Liquid (Shopify)...

Avantages de Twig

Utiliser un moteur de template comme Twig permet de mieux organiser son code et de :

- Séparer le code HTML du PHP ce qui permet aux développeurs Front et Back de travailler en parallèle
- Alléger le code PHP : produire du HTML en PHP impose de multiplier les guillemets et opérateurs de concaténation illisibles
- De mutualiser et réutiliser des portions de page HTML et d'**optimiser** ainsi leur codage

Syntaxe Twig

Un gabarit Twig est un fichier HTML contenant des balises spéciales placées entre accolades qui sont :

- des variables : {{ pseudo }}, {{ livre.titre }}
- des instructions :
 - boucles : {%- for element in tableau %} ... {%- endfor %}
 - tests : {%- if condition %} ... {%- else %} ... {%- endif %}

Exemple d'utilisation

Imaginons vouloir générer une page HTML composée d'un titre, d'un mot de bienvenue et d'une liste d'images. Les contenus peuvent être placés dans des variables PHP ainsi :

```
<?php  
$titre = 'Galerie';
```

```
$pseudo = 'David';
$tableau_url = ['image1.jpeg', ..., 'image4.jpeg'];
```

La page à trous est codée avec Twig en remplaçant les contenus par des variables ou instructions Twig :

```
<html>
  <head> ... </head>
  <body>
    <h1>{{ titre }}</h1>
    <p>Bienvenue {{ pseudo }}</p>
    {% for url in tableau_url %}
      <img src={{ url }}>
    {% endfor %}
  </body>
</html>
```

Le fichier PHP injectant les contenus dans le gabarit Twig s'écrit ainsi :

```
<?php
// Chargement de Twig
$loader = new \Twig\Loader\FilesystemLoader('templates');
$twig = new \Twig\Environment($loader);

// Exécution de Twig
echo $twig->render('essai.twig', [
  'titre' => $titre,
  'pseudo' => $pseudo,
  'tableau_url' => $tableau_url
]);
```

Le code HTML produit après injection des variables PHP dans le template Twig :

```
<html>
  <head> ... </head>
  <body>
    <h1>Galerie</h1>
    <p>Bienvenue David</p>
    
    
    
    
  </body>
</html>
```

Exemple d'une page affichant une fiche client

- Le contenu peut également être placé dans des variables structurées comme ici dans un tableau associatif qui contient les informations concernant un client :

```
// Fichier data_client.php
<?php
// Exemple de données pour un client
$client = [
    'nom' => 'De Niro',
    'prénom' => 'Robert',
    'adresse' => '1 rue principale, 67400 Haguenau',
    'code_client' => 'XYZ123'
];
```

- Le modèle (ou template) est une structure HTML qui est commune à tous les clients à afficher.
- Dans le code ci-dessous, les éléments en gris sont les éléments communs à toutes les pages, les éléments coloriés sont les variables.

```
// Fichier detail_client.twig
{# Exemple de modèle d'une page client #}
<div class="client">
    <h3>Fiche client n° {{ client.code_client }}</h3>
    <p>Nom : {{ client.nom }} - Prénom : {{ client.prenom }}</p>
    <p>Adresse : {{ client.adresse }}</p>
</div>
```

- Pour générer une page Web en utilisant Twig, on procède alors en 3 étapes :
 1. Initialisation de Twig : la fonction `init_twig()` définit dans `twig.php` s'en charge.
 2. Chargement des données : les contenus sont définis dans des variables PHP et inclus.
 3. Appel du moteur Twig : on indique le modèle de page à utiliser, ici `exemple.twig`, et la liste des variables avec leurs noms

```
<?php
// Initialisation de Twig
include('include/twig.php');
$twig = init_twig();

// Chargement des contenus ou données client
include('data_client.php');

// Remplissage du modèle avec les données et affichage du rendu HTML
echo $twig->render('detail_client.twig', [
    'client' => $client
]);
```

Les modèles de pages Twig

Je présente les instructions de base pour utiliser Twig dans cette page.

Pour aller plus, consultez la documentation en ligne sur twig.symfony.com

Afficher le contenu d'une variable

En PHP, pour afficher le contenu d'une variable dans une balise, on utilise l'instruction echo, des chaînes de caractères et l'opérateur de concaténation ;

```
echo '<h1>' . $titre . '</h1>';
```

Avec Twig, on écrit les balises dans le fichier HTML et on indique l'emplacement où la variable sera insérée :

```
<h1>{{ titre }}</h1>
```

Afficher le contenu d'un tableau associatif

En PHP, pour afficher le contenu d'un tableau associatif, on affiche le contenu de chaque case en les nommant ;

```
echo '<p>Nom du client : ' . $client['nom'] . '</p>';
echo '<p>Adresse : ' . $client['adresse'] . '</p>';
```

Avec Twig, on procède de la même manière avec la syntaxe suivante :

```
<p>Nom du client : {{ client.nom }}</p>
<p>Adresse : {{ client.adresse }}</p>
```

Afficher le contenu d'un tableau de taille variable

En PHP, pour afficher le contenu d'un tableau de taille variable, on utilise une boucle pour afficher chacun de ses éléments ;

```
echo '<ol>';
foreach ($tableau as $element) {
    echo '<li>' . $element . '</li>';
}
echo '</ol>';
```

Avec Twig, on écrit également une boucle avec la syntaxe :

```
<ol>
{% for element in tableau %}
    <li>{{ element }}</li>
{% endfor %}
</ol>
```

Arborescence des fichiers et organisation du code

Un site Web utilisant le moteur Twig doit contenir les éléments suivants :

- Un dossier `templates` contenant les modèles de pages
- Un dossier `include` contenant vos fichiers PHP
 - Le projet contient le fichier `twig.php` qui définit la fonction `init_twig()`
 - Le fichier `data.php` qui contient des exemples de données
- Les dossiers classiques des projets Web (`css`, `images`, etc.)
- Un fichier PHP par pages du site
 - Chacun de ces fichiers utilise le schéma décrit plus haut (initiation de twig, chargement des données, appel à Twig)

From:

<https://intra.aha.unistra.fr/> -



Permanent link:

https://intra.aha.unistra.fr/doku.php?id=intranet_mmi:david_cazier:twig

Last update: **28/01/2025 14:22**