

Développement Web

Gestion de rendu et décoration

Halim Djerroud <hdd@ai.univ-paris8.fr>

LIASD - Université Paris 8

2018

Plan de cours général

8 semaines de cours :

- 1 Introduction au web dynamique
- 2 **Gestion de rendu et décoration**
- 3 Le duo magique : PHP et MySql
- 4 Programmation OO et les design patterns
- 5 TP noté pour faire une pause et un peu de JS avec ses saveurs (jQuery).
- 6 Les frameworks et Symfony4
- 7 Symfony4 et ses rouages
- 8 Correction de projets

Plan de cours

- 1h30 Cours
 - 2h00 Travaux pratiques.
- 1 HTML
 - 2 CSS
 - 3 Framework CSS **Bootstrap**
 - 4 Moteurs de templates Twig (et Smarty s'il nous reste du temps)

Support de cours

- <http://djerroud.halim.info/index.php/cours/programmation-web/>

Introduction HTML

- HTML (Hyper Texte Markup Language) est d'un langage de description (et non pas d'un langage de programmation) qui va nous permettre de décrire l'aspect d'un document, d'y inclure des informations variées (textes, images, sons, animations etc.) et d'établir des relations cohérentes entre ces informations grâce aux liens hypertextes

Avantages HTML

Les avantages du langage HTML sont nombreux

- Peux coûteux en effet un simple éditeur de texte suffit à écrire ses premiers documents HTML
- Relativement facile à aborder,
- Il représente en outre un bon moyen de dépasser les problèmes de compatibilité entre des systèmes et des formats informatiques différents

Versions HTML

- HTML5 :

```
<!DOCTYPEhtml>
```

- HTML 4.01

```
<!DOCTYPEHTMLPUBLIC"-//W3C//DTDHTML4.01Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
```

- XHTML 1.0

```
<!DOCTYPEhtmlPUBLIC"-//W3C//DTDXHTML1.0Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Structure Document HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mon premier code HTML !</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Ceci est un gros titre</h1>
    <p>Ceci est un paragraphe</p>
  </body>
</html>
```


Structure Document HTML

Ceci est un gros titre

Ceci est un paragraphe

FIGURE – Structure document HTML

Balises HTML (paire de balises)

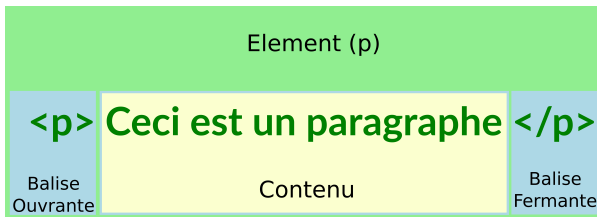


FIGURE – Paire de balises

Balises HTML (balise orpheline)

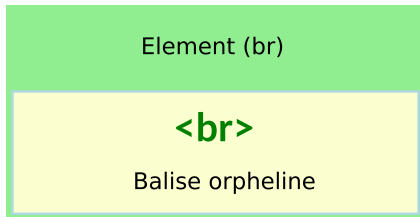


FIGURE – Balise orpheline

Les commentaires HTML

```
...  
<!-- mise en relief -->  
<!--  
un commentaire sur  
plusieurs lignes  
-->  
...
```

Attributs HTML

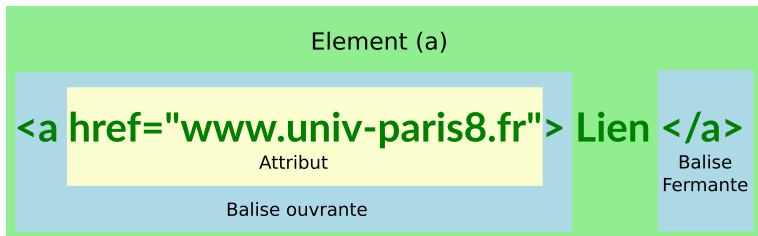


FIGURE – Attribut

Les titres en HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Première page HTML</title>
  <meta charset= "utf-8">
</head>
<body>
  <!--Un titre très important-->
  <h1>Mon titre principal</h1>
  <!--Un titre important-->
  <h2>Je suis un titre important</h2>
  <!--Un autre titre important-->
  <h2>Moi aussi, je suis un titre important</h2>
  <!--Un titre un peu moins important-->
  <h3>Je suis un titre d'important moyenne</h3>
  <h4>Un titre pas très important</h4>
  <h5>Un titre peu important</h5>
  <h6>Un titre vraiment peu important</h6>
</body>
</html>
```

Les titres en HTML

Mon titre principal

Je suis un titre important

Moi aussi, je suis un titre important

Je suis un titre d'important moyenne

Un titre pas très important

Un titre peu important

Un titre vraiment peu important

FIGURE – Les titres en HTML

Les titres en HTML

```
<!DOCTYPE html>
...
<body>
  ...
  <!--Deux paragraphes différents-->
  <p>Voici mon premier paragraphe.</p>
  <p>Et en voilà un second !</p>
</body>
</html>
```

Voici mon premier paragraphe.

Et en voilà un second !

FIGURE – Paragraphe en HTML

Les espaces et retours à la ligne en HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Première page HTML</title>
    <meta charset= "utf-8">
  </head>

  <body>
    <!--Les espaces ne seront pas affichés-->
    <h1>Mon titre                principal</h1>
    <h2>Je suis          un          titre important</h2>

    <!--br nous sert à créer des retours à la ligne-->
    <p>Voici <br>mon <br>premier <br>paragraphe.</p>
    <p>Et en voilà un second !</p>
  </body>
</html>
```

Les espaces et retours à la ligne en HTML

Mon titre principal

Je suis un titre important

Voici
mon
premier
paragraphe.

Et en voilà un second !

FIGURE – Les espaces et retours à la ligne en HTML

Mise en forme Texte (Gras et Italique)

```
<!DOCTYPE html>
...
<h1>Le langage HTML</h1>
<p>
  HTML signifie <strong>HyperText Markup Language</strong>. Ce
  langage est utilisé pour marquer <em>sémantiquement</em> un contenu.
</p>
...
```

Le langage HTML

HTML signifie **HyperText Markup Language**. Ce langage est utilisé pour marquer *sémantiquement* un contenu.

FIGURE – Mise en forme texte en HTML

Les listes en HTML

```
<!DOCTYPE html>
...
<h1>Les listes</h1>
<!--Un exemple de liste non-ordonnée-->
<ul>
  <li>Art</li>
  <li>Musique</li>
  <li>Roman</li>
</ul>
...
```

Les listes

- Art
- Musique
- Roman

FIGURE – Liste non-ordonnée en HTML

Les listes ordonnées en HTML

```
<!DOCTYPE html>
...
<h1>Les listes</h1>
<!--Un exemple de liste ordonnée-->
<ol>
  <li>Art</li>
  <li>Musique</li>
  <li>Roman</li>
</ol>
...
```

Les listes

1. Art
2. Musique
3. Roman

FIGURE – Liste ordonnée en HTML

Les liens en HTML

```
<!DOCTYPE html>
...
<h1>Les liens</h1>

<p>Cliquez sur <a href="http://wikipedia.org">ce lien </a>pour
aller sur Wikipédia.</p>
...
```

Les liens

Cliquez sur [ce lien](http://wikipedia.org) pour aller sur Wikipédia.

FIGURE – Lien en HTML

Les types de liens en HTML

- Les liens peuvent pointer sur des pages interne c-à-d à d'autres pages au sein du même site
- Les liens peuvent pointer sur la même page, pour aller à un paragraphe particulier, pour cela il faut définir l'attribut ID

```
<!DOCTYPE html>
...
<h1>Les liens internes</h1>

<p>Cliquez sur <a href="#p1">ce lien </a>pour aller au paragraphe 1</p>
<p>Cliquez sur <a href="#p1">ce lien </a>pour aller au paragraphe 2</p>

<h1 id="p1">Paragraphe 1</h1>
<p>Contenu du paragraphe 1</p>
<h1 id="p2">Paragraphe 2</h1>
<p>Contenu du paragraphe 2</p>
...
```

Les images en HTML

```
<!DOCTYPE html>
...

...
```



FIGURE – Les images en HTML

Les entités en HTML

Nom	Résultat
<	<
>	>
&	&
 	—
...	...

Liste complète

<https://dev.w3.org/html5/html-author/charref>

Les tableaux en HTML

```
<!DOCTYPE html>
...
<h1>Un tableau simple</h1>

<table border="1.0">
  <tr>
    <td>Nom</td>
    <td>Prénom</td>
    <td>Age</td>
  </tr>
  <tr>
    <td>Dupond</td>
    <td>Dupont</td>
    <td>25 ans</td>
  </tr>
  <tr>
    <td>Durand</td>
    <td>Durant</td>
    <td>52 ans</td>
  </tr>
</table>
...
```

Les tableaux en HTML

Un tableau simple

Nom	Prénom	Age
Dupond	Dupont	25 ans
Durand	Durant	52 ans

FIGURE – Un tableau simple en en HTML

Les tableaux avancés en HTML

```
<!DOCTYPE html>
...
<h1>Un tableau avec fusion de cellules</h1>
<table border="1.0">
  <tr>
    <th colspan="3">Prénom(s)</th>
    <th colspan="2">Adresse(s) e-mail</th>
  </tr>
  <tr>
    <td>Pierre</td>
    <td colspan="2">Victor</td>
    <td colspan="2">pierre.giraud@edhec.com</td>
  </tr>
  <tr>
    <td>Pauline</td>
    <td>Jeanne</td>
    <td>Mireille</td>
    <td>pauline.j@gmail.com</td>
    <td>p.joly@gmail.com</td>
  </tr>
</table>
...
```

Les tableaux avancés en HTML

Un tableau avec fusion de cellules

Prénom(s)			Adresse(s) e-mail	
Pierre	Victor		pierre.giraud@edhec.com	
Pauline	Jeanne	Mireille	pauline.j@gmail.com	p.joly@gmail.com

FIGURE – Un tableau avec fusion de cellules

Les formulaires en HTML

- Permettre de recueillir des données envoyées par nos utilisateurs
- Deux méthodes pour envoyer les données (POST vs GET)

Les éléments de formulaire en HTML

- **form**: Définit un formulaire
- **input**: Définit un champ de données pour l'utilisateur
- **label**: Définit une légende pour un élément input
- **textarea**: Définit un champ de texte long
- **select**: Définit une liste de choix
- **optgroup**: Définit un groupe d'options dans une liste
- **option**: Définit une option dans une liste
- **fieldset**: Permet de regrouper les éléments d'un formulaire en différentes parties
- **legend**: Ajoute une légende à un élément fieldset

Structure d'un formulaire en HTML

```
<!DOCTYPE html>
...
<form method="post" action="traitement.php">
  <label for="pseudo">Entrez un pseudo : </label>
  <input type="text" name="pseudo" id="pseudo">
  <br><br>

  <label for="presentation">Présentation : </label>
  <textarea name="prez" id="presentation"></textarea>
  <br><br>
  <input type="submit" value="Envoyer">
</form>
</table>
...
```

Les tableaux avancés en HTML

Les formulaires HTML

Entrez un pseudo :

Présentation :

Envoyer

FIGURE – Formulaire en HTML

Les formulaires en HTML (l'élément form)

- **form**: Sert à définir un formulaire
- Il a besoin de deux attributs obligatoires :
 - **method**: Définit comment doivent être envoyées les données saisies par l'utilisateur, deux valeurs possibles :
 - **GET**: Les données transitent via l'**URL**
 - Taille de données limité
 - Les données sont envoyées en clair (visibles) dans l'URL
 - **POST**: Les données transitent via l'**HTTP post transaction**
 - Taille de données illimité
 - Les données ne sont visibles.
 - **action**:
 - Sert à indiquer l'adresse de la page à laquelle les données sont envoyées

Les éléments HTML input

- L'élément `input` permet de créer les champs de saisie du formulaire
- L'élément `input` se présente sous la forme d'une balise orpheline
- L'attribut `type` de l'élément `input` permet de définir le type de données de ce champs (nombre, mail, mot de passe ...)

Entrez un pseudo :

Les éléments HTML textarea

- L'élément `textarea` va nous permettre de créer un champ de texte

Présentation :

A screenshot of a web form. It features a label "Présentation :" followed by a rectangular text area input field. The text area is empty and has a thin border. A small diagonal line is visible in the bottom right corner of the text area, possibly indicating a scroll or a specific styling.

FIGURE – Formulaire en HTML (textarea)

Les éléments HTML label

- L'élément `label` pour décrire à l'utilisateur ce qu'il doit rentrer dans chaque champ du formulaire
- Pour lier un label à un élément de formulaire il faut renseigner les attributs `for` (pour le label) et `id` (pour l'élément de formulaire).

Entrez un pseudo :

Bonnes pratiques pour bien coder en HTML

- Indentation
- Commentaires HTML
- Écrire du HTML Standard de préférence utiliser la version 5 de HTML
- Un bon éditeur avec coloration syntaxique
- Utiliser des validateur HTML
- Tester sur différents navigateurs

Introduction CSS

Problèmes HTML :

- HTML n'a jamais été destiné à contenir des balises de mise en forme d'une page web !
- HTML a été créé pour décrire le contenu d'une page web

Résolution par le CSS :

- Permet la mise en page d'un contenu
- Appliquer un style à un élément pour changer son apparence
- Avant d'appliquer un style il faut obligatoirement **sélectionner** un ou plusieurs éléments.

Syntaxe CSS

- Les points de sélection à l'élément HTML que vous voulez appliquer le style
- Le bloc de déclaration contient une ou plusieurs déclarations séparées par des points-virgules
- Chaque déclaration inclut un nom de propriété CSS et une valeur, séparés par deux points
- Une déclaration de CSS se termine toujours par un point-virgule, et blocs de déclaration sont entourés par des accolades

Syntaxe CSS

- Dans l'exemple suivant tous les éléments `<p>` seront aligné au centre, avec une couleur de texte en rouge



FIGURE – Formulaire en HTML (input)

Sélecteur CSS

Sélection de l'élément P

```
p {  
    color: red;  
    text-align: center;  
}
```

Sélecteur CSS

Sélection par l'ID

```
...  
#para1 {  
    text-align: center;  
    color: red;  
}  
...  
<p id="para1"> mon text </p>  
...
```

Sélecteur CSS

Sélection par classe

```
...  
.center {  
  text-align: center;  
  color: red;  
}  
...  
<h2 class="center"> titre </h2>  
...  
<p class="center"> mon text </p>  
...
```

Regrouper les sélecteurs CSS

sélecteurs individuels

```
h1 {
  text-align: center;
  color: red;
}
h2 {
  text-align: center;
  color: red;
}
p {
  text-align: center;
  color: red;
}
```

sélecteurs regroupés

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Les commentaires CSS

Les commentaires CSS

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}  
  
/* This is  
a multi-line  
comment */
```

Intégration feuilles de styles

Il existe trois façons d'intégrer le CSS dans une page HTML :

- Dans la page HTML
- Dans une feuille de style externe
- Intégrer le styles dans les balises HTML

Intégration feuilles de styles Dans la page HTML

```
<head>
  <style>
    body {
      background-color: linen;
    }

    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
```

Intégration feuilles dans une feuille de style externe

Page HTML

```
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Contenu du fichier : mystyle.css

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Intégrer le styles dans les balises HTML

- Un style en ligne peut être utilisé pour appliquer un style unique pour un seul élément.
- Pour utiliser les styles en ligne, ajoutez l'attribut de style à l'élément concerné. L'attribut de style peut contenir toute propriété CSS.
- L'exemple ci-dessous montre comment changer la couleur et la marge de gauche d'un `<h1>` élément :

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

Intégration dans plusieurs feuilles de style

Page CSS 1

```
h1 {  
    color: navy;  
}
```

Page CSS 2

```
h1 {  
    color: orange;  
}
```

- Le dernier style est appliqué

Les propriétés CSS

- Permet de sélectionner un style et lui appliquer une valeur
- Liste complète à consulter :

<https://www.w3schools.com/cssref/>

Couleurs nommées

Color	Name
	Red
	Green
	Blue
	Orange
	Yellow
	Cyan
	Black

FIGURE – Couleurs en HTML

Couleurs RGB

Color	RGB
	<code>rgb(255,0,0)</code>
	<code>rgb(0,255,0)</code>
	<code>rgb(0,0,255)</code>
	<code>rgb(255,165,0)</code>
	<code>rgb(255,255,0)</code>
	<code>rgb(0,255,255)</code>

FIGURE – Couleurs en HTML

Couleurs Hexadécimal

Color	HEX
	#FF0000
	#00FF00
	#0000FF
	#FFA500
	#FFFF00
	#00FFFF

FIGURE – Couleurs en HTML

Les Bordures

```
div {  
  width: 300px;  
  padding: 25px;  
  border: 25px solid navy;  
  margin: 25px;  
}
```

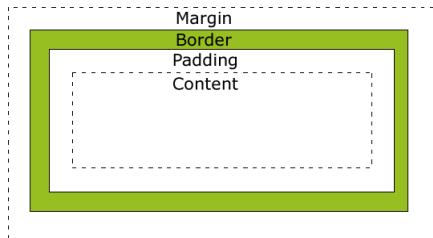


FIGURE – Couleurs en HTML

Bonnes pratiques pour bien coder en CSS

- Indentation
- Commentaires CSS
- Écrire du CSS intelligemment (Utiliser l'héritage, **LESS** (Leaner Style Sheets), etc.),
- Un bon éditeur avec coloration syntaxique
- Utiliser des validateur CSS
- Tester sur différents navigateurs

Introduction Bootstrap

- Écrit par les développeurs de Twitter en 2010 (en licence Open-Source depuis 2011)
- Bootstrap est un framework CSS permettant de faciliter l'écriture de la mise en forme CSS
- Bootstrap permet gagnerez du temps

Avantages Bootstrap

- Responsive : Adaptable à l'écran
- Bon rendu (beau) : Bootstrap propose plusieurs thèmes
- Un gain de temps : Facile à utiliser et apprentissage rapide avec une très bonne documentation
- Compatibilité : Compatible avec la plupart des Navigateurs
- Mise en cache (Rapidité)

Note :

Un framework est avant tout un cadre de travail permettant de faire profiter aux développeurs d'une arborescence et d'une architecture communes.

site officiel Bootstrap

- Téléchargeable sur <https://getbootstrap.com/>
- Consulter la documentation **Get Started**

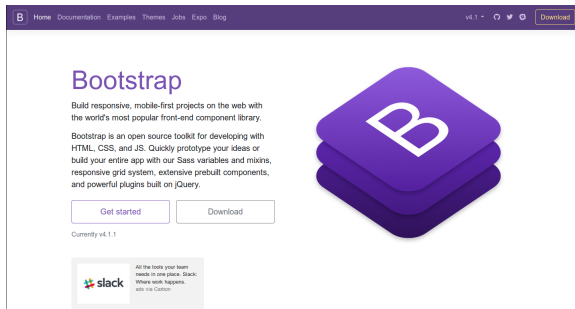


FIGURE – Bootstrap

Contenu Bootstrap

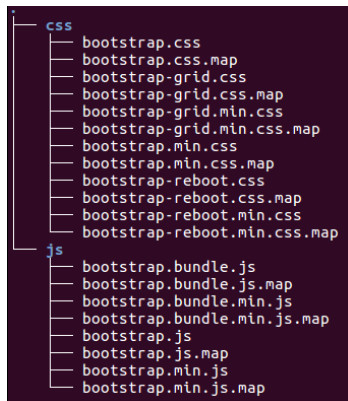


FIGURE – Contenu Bootstrap

Contenu du framework Bootstrap (1/2)

- `bootstrap.css` : fichier principal de bootstrap avec la déclaration de toutes les classes CSS
- `bootstrap.css.map` : permet de retrouver la position d'une ligne de code dans la version normale à partir du code minifié
- `bootstrap.min.css` : même contenu que le fichier `bootstrap.css` de manière minifié
- `bootstrap-theme.css` : permet de créer des thèmes spécifiques
- `bootstrap-theme.css.map` : permet de retrouver la position d'une ligne de code dans la version normale à partir du code minifié

Contenu du framework Bootstrap (2/2)

- `bootstrap-theme.min.css` : même contenu que le fichier `bootstrap-theme.css` de manière mimifiée
- `glyphicons-halflings-regular.eot` : icône au format EOT ... (même chose pour les autres formats d'icônes)
- `bootstrap.js` : contient le code javascript de bootstrap
- `bootstrap.min.js` : Même contenu que le fichier `bootstrap.js` de manière mimifié
- `npm.js` : fichier généré automatiquement pouvant être utile dans l'environnement CommonJS

Intégration de bootstrap dans un projet

```
<!Doctype html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Mon Site avec Bootstrap</title>
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
    ...
  </body>
</html>
```

Introduction moteur de templates

- Séparer la présentation des données du traitement
- Personnalisation de pages web
- Rendre les pages web plus lisibles, plus claires
- Rapidité de développement
- Sécurité
- Rapidité à l'exécution (compilation du code)

Fonctionnement moteur de templates

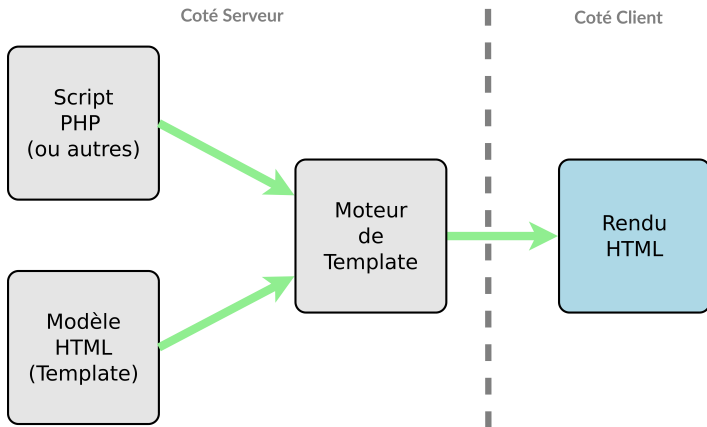


FIGURE – Fonctionnement moteur de templates

Moteur de templates

- **Twig** : Moteur de template développé pour Symfony
- **Smarty** : L'un des premier moteur de templates, très bien documenté et puissant
- **Mustache** : Assure une compatibilité avec beaucoup de langages (Php, Python, Bash ...)
- **Plates** : Surcharge du PHP natif, pas de compilation (interprétation) inspirée de Twig
- **Liquid** : Multiplate-forme et personnalisation coté utilisateur
- **Haml** : Écrit pour Ruby, fonctionne avec indentation (inspiré par Python)
- ...

Avantage Twig

- Facile à apprendre
- Les messages d'erreur Twig sont clairs et précis
- Coloration syntaxique supportée par plusieurs éditeurs de code
- Utilisé par le framework Symfony

Exemple Twig

PHP

```
<ul>
<?php foreach ($list as $value):
    if ($value.element) {?>
        <li>"<?php echo element;?></li>
<?php } endforeach;?> </ul>
```

Twig

```
<ul>
{% for value in list if value.element %}
    <li> {{ value.element }} </li>
{% endfor %}
</ul>
```

Obtenir Twig avec composer

```
composer
```

```
$composer require twig/twig
```

Exemple (1/4)

Création du projet et installation Twig

```
mkdir monProjet  
cd monProjet  
composer require twig/twig
```


Exemple (2/4)

Le fichier maPage.php

```
<?php
require_once("vendor/autoload.php");

$loader = new \Twig_Loader_Filesystem(__DIR__.'/templates');
$twig = new \Twig_Environment($loader);

echo $twig->render('demo.twig', ['name' => 'hello world']);
```

Exemple (3/4)

Le fichier demo.twig

```
mkdir templates
```

```
cd templates
```

```
echo "Message: {{ name }}" > demo.twig
```

Exemple (4/4)

Exécution

```
cd ..  
php maPage.php
```

Résultat

```
Message : hello world
```

Fonctionnement

- `{% %}` : pour faire une action (Affectation, boucle, condition ...)
- `{{ }}` : Affichage (pour afficher quelque chose)

Consulter la documentation Twig

- <https://twig.symfony.com/doc/2.x/>
- <https://iner-dukoid.developpez.com/tutoriels/web/moteur-template-twig/>