

```

<h1 id="introduction-au-html">Introduction au HTML</h1>
<p>Vous vous demandez sûrement ce qu'est un template.</p>
<p>Un template est un fichier que vous pouvez réutiliser afin de
pré-sélectionner des informations différentes sous un seul et même
format. Par exemple, vous pourriez avoir envie d'utiliser un template
pour écrire une lettre : bien que son contenu varie ou qu'elle
puisse être adressée à des personnes différentes, sa
forme reste la même.</p>
<p>Le format d'un template Django est écrit en ce langage qui s'appelle HTML (c'est le même HTML que celui dont
nous parlions dans le chapitre un, <strong>Comment fonctionne
l'Internet</strong>).</p>
<h2 id="quest-ce-que-le-html">Qu'est-ce que le HTML ?</h2>
<p>HTML est un code qui est interprété par votre navigateur
(Chrome, Firefox ou Safari) et qui permet d'afficher une page web
à l'utilisateur.</p>
<p>L'abréviation HTML signifie HyperText Markup Language. HyperText signifie que c'est un type de texte
qui supporte les liens entre les pages. Markup signifie que nous avons pris un document et que nous avons balisé le
code pour signifier (ici, au navigateur) comment il faut interpréter la
page. Le code HTML est construit à l'aide de
balises, chacune commençant par < et
finissant par >. Ces balises représentent des
éléments markup.</p>
<h2 id="votre-premier-template">Votre premier template !</h2>
<p>Créer un template signifie créer un fichier template. Et oui,
encore des fichiers ! Vous aviez déjà probablement remarqué que tout
tourne autour des fichiers.</p>
<p>Les templates sont sauvegardés dans le dossier
blog/templates/blog. Tout d'abord, créons un dossier
appelé templates à l'intérieur du dossier
de notre blog. Ensuite, créez un autre dossier appelé
blog à l'intérieur de votre dossier templates
:</p>
<pre><code>blog
#x2514;#x2500;#x2500;#x2500;templates
  #x2514;#x2500;#x2500;#x2500;blog
</code></pre><p>Vous pourriez vous demander pourquoi nous avons besoin de
deux dossiers portant tous les deux le nom blog. Comme vous le
découvrirez plus tard, c'est une convention de nommage qui va nous
faciliter la vie quand les choses vont commencer à devenir
compliquées.</p>
<p>Et maintenant, créez un fichier post_list.html (laissez
le vide pour le moment) dans le dossier
templates/blog/blog.</p>
<p>Allons regarder à quoi ressemble notre site maintenant : <a
href="http://127.0.0.1:8000/" target="_blank">http://127.0.0.1:8000/</a></p>
<blockquote>
<p>Si vous avez encore l'erreur TemplateDoesNotExist,
essayez de redémarrer votre serveur. Allez sur votre ligne de commande

```

et arrêtez votre server en appuyant simultanément sur Ctrl+C (les touches Control et C de votre clavier). Vous pouvez le relancer en tapant la commande `python manage.py runserver`.



Et voilà, il n'y a plus d'erreurs ! Bravo :) Cependant, notre site ne peut rien faire d'autre pour le moment qu'afficher une page blanche. La faute, notre template que nous avons laissée vide. Allons corriger ça.

Ouvrez le nouveau fichier dans l'éditeur de code et ajoutez le morceau suivant :

```
blog/templates/blog/post_list.html
<code class="lang-html"><span class="hljs-tag">&lt;<span class="hljs-name">html</span>&gt;</span>
<span class="hljs-tag">&lt;<span class="hljs-name">body</span>&gt;</span>
  <span class="hljs-tag">&lt;<span class="hljs-name">p</span>&gt;</span>Hi there!<span class="hljs-tag">&lt;</span>
  <span class="hljs-tag">&lt;<span class="hljs-name">p</span>&gt;</span>It works!<span class="hljs-tag">&lt;</span>
<span class="hljs-tag">&lt;<span class="hljs-name">body</span>&gt;</span>
<span class="hljs-tag">&lt;<span class="hljs-name">html</span>&gt;</span>
</code></pre>
```

Alors, quoi ressemble notre site web maintenant ? Allons le découvrir : <http://127.0.0.1:8000/>



ça marche ! Bon boulot :)

- La balise la plus élémentaire, `&lt;html&gt;`, figure toujours au début de n'importe quelle page web tandis que `&lt;/html&gt;` est toujours située à la fin. Comme vous pouvez le constater, l'indentation du contenu de notre page web est située entre la balise de départ, `&lt;html&gt;`, et la balise fermante, `&lt;/html&gt;`
- `&lt;p&gt;` est la balise pour les éléments de type paragraphe. `&lt;/p&gt;` permet de fermer chaque paragraphe.

## Head et body

Chaque page HTML est divisée en deux éléments : **head** (entête) et **body** (corps).

- head** est un élément qui contient des informations sur le document : son contenu ne s'affichera pas à l'écran.
- body** est un élément qui contient tout le reste. Son contenu s'affichera à l'écran et constituera notre page web.

```

</li>
</ul>
<p>Nous utilisons <code>&lt;head&gt;</code> pour transmettre la
configuration de la page au navigateur tandis que <code>&lt;body&gt;</code>
l&apos;informe sur le contenu de la page.</p>
<p>Par exemple, vous pouvez donner un titre &#xE0; votre page web en
utilisant l&apos;&#xE9;l&#xE9;ment titre dans le <code>&lt;head&gt;</code>
:</p>
<p></p><p class="code-label">blog/templates/blog/post_list.html</p><p></p>
<pre><code class="lang-html"><span class="hljs-tag">&lt;<span class="hljs-
name">html</span>&gt;</span>
  <span class="hljs-tag">&lt;<span class="hljs-
name">head</span>&gt;</span>
    <span class="hljs-tag">&lt;<span class="hljs-
name">title</span>&gt;</span>0la&apos;s blog<span class="hljs-
tag">&lt;/span class="hljs-name">title</span>&gt;</span>
  <span class="hljs-tag">&lt;/span class="hljs-
name">head</span>&gt;</span>
  <span class="hljs-tag">&lt;<span class="hljs-
name">body</span>&gt;</span>
    <span class="hljs-tag">&lt;<span class="hljs-
name">p</span>&gt;</span>Hi there!<span class="hljs-tag">&lt;/span
class="hljs-name">p</span>&gt;</span>
    <span class="hljs-tag">&lt;<span class="hljs-
name">p</span>&gt;</span>It works!<span class="hljs-tag">&lt;/span
class="hljs-name">p</span>&gt;</span>
    <span class="hljs-tag">&lt;/span class="hljs-
name">body</span>&gt;</span>
<span class="hljs-tag">&lt;/span class="hljs-name">html</span>&gt;</span>
</code></pre>
<p>Sauvegardez votre fichier et actualisez la page.</p>
<p></p>
<p>Vous avez vu comment le navigateur a compris que &#xAB; Le Blog
d&apos;0la &#xBB; est le titre de votre page ? Il a interpr&#xE9;t&#xE9;
<code>&lt;title&gt;Le blog d&apos;0la&lt;/title&gt;</code> et a plac&#xE9;
ce texte dans la barre de titre de votre navigateur (c&apos;est ce titre qui
va &#xEA;tre aussi utilis&#xE9; lorsque vous cr&#xE9;ez un marque-page,
etc.).</p>
<p>Vous avez aussi probablement remarqu&#xE9; que chaque balise ouvrante
poss&#xE8;de sa <em>balise fermante</em>, compos&#xE9;e d&apos;un
<code>/</code>, est qu&apos;elles <em>encadrent</em> les diff&#xE9;rents
&#xE9;l&#xE9;ments. Cela signifie que vous ne pouvez pas fermer une balise
si celles imbriqu&#xE9;es &#xE0; l&apos;int&#xE9;rieur de celle-ci
n&apos;ont pas &#xE9;t&#xE9; ferm&#xE9;es.</p>
<p>Pensez &#xE0; lorsque vous mettez des choses &#xE0; l&apos;int&#xE9;rieur
de bo&#xEE;tes. Vous avez une grosse bo&#xEE;te,
<code>&lt;html&gt;&lt;/html&gt;</code>; &#xE0; l&apos;int&#xE9;rieur de
celle-ci, on trouve une plus petite bo&#xEE;te,
<code>&lt;body&gt;&lt;/body&gt;</code>, qui contient elle-m&#xEA;me
d&apos;autres petites bo&#xEE;tes, <code>&lt;p&gt;&lt;/p&gt;</code>.</p>
<p>Essayez de vous rappeler de cet exemple lorsque vous utilisez les balises

```

**<em>**fermantes**</em>** et que vous avez des **&#xE9;l&#xE9;ments**  
**<em>**imbriqu**&#xE9;s</em>**. Si vous ne suivez pas ces r**&#xE8;gles**, votre navigateur risque de ne pas **&#xEA;tre** capable d**&apos;**interpr**&#xE9;**ter votre code correctement et votre page web sera mal affich**&#xE9;**e.**</p>**

**<h2 id="personnaliser-votre-template">**Personnaliser votre template**</h2>**

**<p>**Et si nous en profitons pour nous amuser un peu ? Essayons de personnaliser notre template ! Voici quelques balises que vous pouvez utiliser :**</p>**

**<ul>**

**<li><code>&lt;h1&gt;Titre 1&lt;/h1&gt;</code>** pour vos titres les plus importants**</li>**

**<li><code>&lt;h2&gt;Titre 2&lt;/h2&gt;</code>** pour les sous-titres**</li>**

**<li><code>&lt;h3&gt;Titre 3&lt;/h3&gt;</code>** ... et ainsi de suite jusqu**&apos;****&#xE0;** **<code>&lt;h6&gt;</code></li>**

**<li><code>&lt;p&gt;**Un paragraphe contenant du texte**&lt;/p&gt;</code></li>**

**<li><code>&lt;em&gt;**texte**&lt;/em&gt;</code>** permet de mettre l**&apos;**accent sur une partie du texte**</li>**

**<li><code>&lt;strong&gt;**texte**&lt;/strong&gt;</code>** permet de mettre encore plus l**&apos;**accent sur une partie de texte**</li>**

**<li><code>&lt;br&gt;</code>** permet d**&apos;**ins**&#xE9;**rer un saut de ligne (vous ne pouvez rien mettre **&#xE0;** l**&apos;**int**&#xE9;**rieur d**&apos;**un **&#xE9;l&#xE9;**ment **br** et il n**&apos;**y a pas de balise fermante)**</li>**

**<li><code>&lt;a**

href=**&quot;**https://djangogirls.org**&quot;****&gt;**link**&lt;/a&gt;</code>** permet de cr**&#xE9;**er un lien**</li>**

**<li><code>&lt;ul&gt;&lt;li&gt;**premier item**&lt;/li&gt;&lt;li&gt;**second item**&lt;/li&gt;&lt;/ul&gt;</code>** permet de cr**&#xE9;**er des listes, comme celle que nous sommes en train de faire !**</li>**

**<li><code>&lt;div&gt;&lt;/div&gt;</code>** permet de cr**&#xE9;**er une section au sein de la page**</li>**

**</ul>**

**<p>**Voici un exemple d**&#x2019;**un mod**&#xE8;**le complet, copiez et collez-le dans **<code>**blog/templates/blog/post\_list.html**</code>** :**</p>**

**<p></p><p class="code-label">**blog/templates/blog/post\_list.html**</p></p>**

**<pre><code class="lang-html"><span class="hljs-tag">&lt;<span class="hljs-name">**html**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**head**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**title**</span>&gt;</span>**Django Girls blog**<span class="hljs-tag">&lt;/span>&lt;/span>**title**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**head**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**body**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**div**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**h1**</span>&gt;</span>**

**<span class="hljs-tag">&lt;<span class="hljs-name">**a**</span>** **<span class="hljs-attr">**href**</span>=<span class="hljs-string">****&quot;**;/**&quot;****</span>&gt;</span>**Django Girls Blog**<span class="hljs-tag">&lt;/span>**

```

tag">&lt;/span class="hljs-name">a</span>&gt;</span><span class="hljs-
tag">&lt;/span class="hljs-name">h1</span>&gt;</span>
    <span class="hljs-tag">&lt;/span class="hljs-
name">div</span>&gt;</span>

    <span class="hljs-tag">&lt;/span class="hljs-
name">div</span>&gt;</span>
        <span class="hljs-tag">&lt;/span class="hljs-
name">p</span>&gt;</span>published: 14.06.2014, 12:14<span class="hljs-
tag">&lt;/span class="hljs-name">p</span>&gt;</span>
            <span class="hljs-tag">&lt;/span class="hljs-
name">h2</span>&gt;</span><span class="hljs-tag">&lt;/span class="hljs-
name">a</span> <span class="hljs-attr">href</span>=<span class="hljs-
string">&quot;&quot;</span>&gt;</span>My first post<span class="hljs-
tag">&lt;/span class="hljs-name">a</span>&gt;</span><span class="hljs-
tag">&lt;/span class="hljs-name">h2</span>&gt;</span>
                <span class="hljs-tag">&lt;/span class="hljs-
name">p</span>&gt;</span>Aenean eu leo quam. Pellentesque ornare sem lacinia
quam venenatis vestibulum. Donec id elit non mi porta gravida at eget metus.
Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut
fermentum massa justo sit amet risus.<span class="hljs-tag">&lt;/span
class="hljs-name">p</span>&gt;</span>
                    <span class="hljs-tag">&lt;/span class="hljs-
name">div</span>&gt;</span>

                        <span class="hljs-tag">&lt;/span class="hljs-
name">div</span>&gt;</span>
                            <span class="hljs-tag">&lt;/span class="hljs-
name">p</span>&gt;</span>published: 14.06.2014, 12:14<span class="hljs-
tag">&lt;/span class="hljs-name">p</span>&gt;</span>
                                <span class="hljs-tag">&lt;/span class="hljs-
name">h2</span>&gt;</span><span class="hljs-tag">&lt;/span class="hljs-
name">a</span> <span class="hljs-attr">href</span>=<span class="hljs-
string">&quot;&quot;</span>&gt;</span>Mon second post<span class="hljs-
tag">&lt;/span class="hljs-name">a</span>&gt;</span><span class="hljs-
tag">&lt;/span class="hljs-name">h2</span>&gt;</span>
                                    <span class="hljs-tag">&lt;/span class="hljs-
name">p</span>&gt;</span>Aenean eu leo quam. Pellentesque ornare sem lacinia
quam venenatis vestibulum. Donec id elit non mi porta gravida at eget metus.
Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut
f.<span class="hljs-tag">&lt;/span class="hljs-name">p</span>&gt;</span>
                                        <span class="hljs-tag">&lt;/span class="hljs-
name">div</span>&gt;</span>
                                            <span class="hljs-tag">&lt;/span class="hljs-
name">body</span>&gt;</span>
                                                <span class="hljs-tag">&lt;/span class="hljs-
name">html</span>&gt;</span>
</code></pre>
<p>Nous avons cr&#xE9;&#xE9; trois sections &#xE0; l&apos;aide de
<code>div</code>.</p>
<ul>
<li>Le premier <code>div</code> contient le titre de notre blogpost -

```

c&apost;est &#xE0; la fois un titre et un lien</li>  
<li>Les deux autres <code>div</code> contiennent nos posts avec leur date de publication, un titre de post <code>h2</code> qui est cliquable ainsi que deux <code>p</code>s (paragraphe) de texte : un pour la date et l&apost;autre pour notre post.</li>  
</ul>  
<p>Ce qui nous donne :</p>  
<p></p>  
<p>Yaaay ! Pour l&apost;instant, notre template nous permet seulement d&apost;afficher les <strong>m&#xEA;mes informations</strong> alors que nous disions pr&#xE9;c&#xE9;demment qu&apost;il doit nous permettre d&apost;afficher des informations <strong>diff&#xE9;rentes</strong> utilisant le <strong>m&#xEA;me format</strong>.</p>  
<p>Ce qu&apost;on aimerait pouvoir faire maintenant, c&apost;est d&apost;afficher les posts que nous avons cr&#xE9;&#xE9;s pr&#xE9;c&#xE9;demment dans l&apost;interface d&apost;administration de Django. Penchons-nous l&#xE0; dessus.</p>  
<h2 id="une-derni&#xE8;re-chose--d&#xE9;ployer-">Une derni&#xE8;re chose : d&#xE9;ployer !</h2>  
<p>Ne serait-il pas g&#xE9;nial de pouvoir voir tous ces changements en ligne ? Hop, d&#xE9;ployons &#xE0; nouveau !</p>  
<h3 id="commiter-et-pusher-votre-code-sur-github">Committer et pusher votre code sur GitHub</h3>  
<p>Tout d&apost;abord, allons voir quels sont les fichiers qui ont chang&#xE9; depuis notre dernier d&#xE9;ploiement (lancez ces commandes dans votre console locale et non celle de PythonAnywhere) :</p>  
<p></p><p class="code-label">command-line</p><p></p>  
<pre><code>\$ git status</code></pre><p>Assurez-vous d&apost;&#xEA;tre dans le dossier <code>djangogirls</code>. Voici la commande qui permet de dire &#xE0; <code>git</code> d&apost;inclure tout les changements qui ont eu lieu dans ce dossier :</p>  
<p></p><p class="code-label">command-line</p><p></p>  
<pre><code>\$ git add --all .</code></pre><blockquote class="clearfix alert alert-info"><strong class="fa fa-2x fa-edit"></strong>  
<p> <code>--all</code> signifie que <code>git</code> va aussi voir si vous avez supprim&#xE9; des fichiers (par d&#xE9;fait, il ne s&apost;int&#xE9;resse qu&apost;aux nouveaux fichiers ou &#xE0; ceux modifi&#xE9;s). Essayez de vous rappeler du chapitre 3 : <code>.</code> permet de d&#xE9;signer le dossier courant.</p>  
</blockquote>  
<p>Avant que nous puissions uploader nos fichiers, regardons ce que <code>git</code> &#xE0; l&apost;intention de faire (tous les fichiers que <code>git</code> va uploader vont appara&#xEE;tre en vert) :</p>  
<p></p><p class="code-label">command-line</p><p></p>  
<pre><code>\$ git status</code></pre><p>On y est presque : nous devons maintenant lui dire de sauvegarder ces changements dans son historique. Nous allons y ajouter un &quot;message de commit&quot; qui nous permettra de d&#xE9;crire ce qui a &#xE9;t&#xE9; chang&#xE9;. Vous pouvez mettre ce que vous voulez dans un

message de commit. Généralement, il est préférable de mettre quelque chose d'utile qui vous permettra de vous souvenir plus tard de ce que vous avez fait.

```
command-line
```

```
$ git commit -m "Modification du HTML du site"
```

**fa-2x fa-edit**

N'oubliez pas d'utiliser de doubles guillemets autour de votre message de commit.

Une fois que nous avons fait cela, nous pouvons mettre en ligne (pusher) nos modifications sur GitHub :

```
command-line
```

```
$ git push
```

### Puller les modifications sur PythonAnywhere et recharger son appli web

- Allez sur la page des [consoles de PythonAnywhere](https://www.pythonanywhere.com/consoles/). Retournez dans votre **console Bash** ou ouvrez-en une nouvelle puis tapez la commande suivante :

```
PythonAnywhere command-line
```

```
$ cd ~/&lt;your-pythonanywhere-domain>.pythonanywhere.com
$ git pull
[...]
```

N'oubliez pas de remplacer `&lt;your-pythonanywhere-domain>` avec votre propre sous-domaine PythonAnywhere, sans les chevrons. Votre nom de sous-domaine est normalement votre nom d'utilisateur PythonAnywhere, mais dans certains cas, il peut être un peu différent (par exemple si votre nom d'utilisateur contient des lettres capitales). Donc, si cette commande ne fonctionne pas, utilisez la commande `ls` (lister fichiers) pour trouver votre sous-domaine/nom de dossier, puis `cd` jusque `~`.

Maintenant, vous voyez votre code en train d'être téléchargé. Si vous voulez vérifier que le nouveau code est arrivé sur PythonAnywhere, vous pouvez aller sur la page **"Files"** et y retrouver les fichiers (vous pouvez naviguer dans les différentes pages de PythonAnywhere depuis le menu).

- Pour finir, n'oubliez pas de recharger votre application web : onglet [Web](https://www.pythonanywhere.com/web_app_setup/) puis cliquez sur le bouton **Reload**.

Retournez sur votre site en cliquant sur l'adresse en haut de la page : normalement, vous devriez voir la dernière version. Si ce n'est pas le cas, ce n'est pas grave : n'hésitez pas à demander de l'aide à votre coach. :)

From:

<https://wiki.centrale-med.fr/informatique/> - **WiKi informatique**

Permanent link:

[https://wiki.centrale-med.fr/informatique/public:appro-s7:td\\_web:test](https://wiki.centrale-med.fr/informatique/public:appro-s7:td_web:test)

Last update: **2019/11/22 14:49**

