

Toutes les pages du site vont présenter une structure commune :

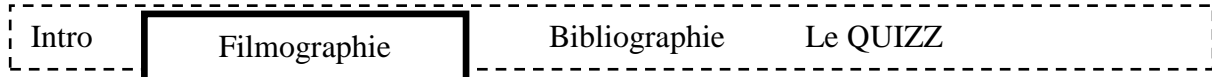
- un en-tête
- un contenu (très variable)
- un pied de page

Le travail que vous allez faire faire pour la page générale qui présente la filmographie de Martin Scorsese sera donc "capitalisable" par la suite.

Activité 1

I:/_xampp/htdocs/dev.fande.fr/apps/templates/layout.tpl

L'en tête du site doit contenir une barre de menu.



Ce type de menu est pratiquement pré-intégré dans le Bootstrap de TWITTER.

On peut trouver une source d'inspiration sur la page

getbootstrap.com/examples/starter-template/

La structure **HTML** est la suivante :

```
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="{&#36;smarty.const.ROOT}">FAN de ...</a>
    </div>
    <div id="navbar" class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="{&#36;smarty.const.ROOT}">Introduction</a></li>
        <li><a href="{&#36;smarty.const.ROOT}filmo">Filmographie</a></li>
        <li><a href="{&#36;smarty.const.ROOT}biblio">Bibliographie</a></li>
        <li><a href="{&#36;smarty.const.ROOT}quizz">Le QUIZZ</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Remarque : on utilise la constante ROOT spécifiée dans le fichier config.inc.php.

L'écriture SMARTY est {$smarty.const.ROOT}.

1. Ajoutez ce code HTML dans la page layout.tpl avant le {block name=body}{/block}.

Activité 2

I:/_xampp/htdocs/dev.fande.fr/apps/templates/filmo.tpl


Le corps de la page présente la filmographie de Martin Scorsese.

HTML

c'est une boîte div#filmo englobante qui va contenir des sections de classe .film

```
<div id="filmo">
  <section class="film"></section>
  <section class="film"></section>
  ...
</div>
```

Le rendu visuel attendu est le suivant (pour une section) :

<div>2002</div>	<div>  </div> <div> <div>★★★★★ [4/5]</div> <div>★★★★★ [4/5]</div> </div>	<div> <div>Gang of New York</div> <div>En 1863, un quartier de New York est mis à feu et à sang bla bla bla ...</div> </div>
<div>.year (7%)</div>	<div>.poster (33%)</div>	<div>.infos (70%)</div>

```

<section class="film">
  <div class="year">
    <p>2002</p>
  </div>
  <div class="poster">
    
    <p class="rate"> * * * * [4/5]</p>
  </div>
  <div class="infos">
    <p class="title">Gang of New York</p>
    <p class="description">
      En 1863, un quartier de New York est mis à feu et à sang bla bla bla ...
    </p>
  </div>
</section>

```

1. Utilisez les classes de positionnement de Bootstrap (.row, .col-x-x) pour obtenir les dimensions identiques à celles spécifiées dans la structure ci-dessus (7%, 33%, 70%).
2. Dans votre fichier de style main.css, ajoutez les modifications nécessaires pour mettre en forme les éléments de section.film


```

      /** la filmographie */
      .film .year {}
      
```
3. Affichez 3 films fictifs (pour faire un maquetage plus réaliste).

Activité 3

I:/_xampp/htdocs/dev.fande.fr/apps/models/movie.class.php

Toutes les données qui vont apparaître dans la page filmo.tpl proviennent de la table **fande_movies**. Vous allez mettre en place une classe qui va exposer 5 **méthodes** qui vous permettront de manipuler les informations de la table **fande_movies**.

Voici le modèle de cette classe nommée **Movie** (même nom que la table associée, sans le s et avec une majuscule) :

Movie
\$table = 'fande_movies'
add (\$datas) get (\$id) getAll () delete (\$id) edit (\$id, \$datas)

Cette classe fait, elle même, appel à la classe générique **Base** de gestion des base de données.

1. Ajoutez le nouveau fichier de classe movie.class.php :

```

class Movie extends Base {

    // table associée
    public $table = 'fande_movies';

}

```

2. Dans le traitement contrôleur de la page filmo.tpl (fonction filmo_action), vous devez instancier la classe Movie; écrivez :

```

$oMovie = new Movie( $fpdo );
debug($oMovie, 'objet oMovie');

```

3. Pour le moment, la table des films est vide; vous allez ajouter 3 films par du code en utilisant les méthodes de la l'objet \$oMovie. Ce code sera écrit dans le contrôleur (puis mis en commentaire à la fin de l'exercice). La méthode **add** de la classe Movie attend un paramètre de type tableau associatif. Ce tableau associatif doit contenir tous les éléments que vous souhaitez ajouter dans les champs de la table.

Écrivez :

```

$oMovie->add(
  array(
    'title' => "Gang of New York",
    'shortdesc' => "En 1863, un quartier de New York est mis à feu et à sang ...",
    'year' => 2002,
  )
)

```

```

        'poster' => "gangofnewyork.png"
    )
};

```

Vérifiez que l'enregistrement est bien créé dans la table `fande_movies`.
Normalement, le film ajouté se voit affecter l'identifiant 1.

4. Ajoutez 2 nouveaux enregistrements en utilisant le même principe

title	shortdesc	year	poster	rates	crates
Casino	Dans les années 1970, Sam « Ace » Rothstein1 (Robe...	1995	casino.png	0	0
Les affranchis	Le film est basé sur le livre Wiseguy, de Nicholas...	1990	lesaffranchis.png	0	0

5. La méthode **get** permet d'afficher un film spécifique à partir de son identifiant. Elle attend deux paramètres :

- l'identifiant du film récupérer
- (en option) la liste des colonnes à récupérer (si ce second paramètre n'est pas renseigné, alors toutes les colonnes sont retournées).

Testez le code

```
debug($oMovie->get(1), 'méthode get');
```

6. La méthode **getAll** retourne tous les films sous forme d'un tableau que l'on peut relire avec la boucle `foreach()`.

Testez le code

```
$films = $oMovie->getAll();
```

Complétez la boucle `foreach` suivante pour afficher les informations sur chaque film un par un :

```
foreach($films as $film) {
    debug($film, $film');
}
```

7. Avec **PHPMysqladmin**, ajoutez un film de test fictif; repérez son identifiant (4 par exemple). En test, écrivez le code qui utilise la méthode **delete** et permet de supprimer le film numéro 4 (vous adapterez le numéro à votre situation).

8. Avec la méthode **edit**, modifiez le colonne 'title' du film numéro 2 en écrivant 'Casino royal'.

Activité 4

I:/_xampp/htdocs/dev.fande.fr/apps/templates/filmo.tpl

Exploitez la méthode `getAll()` de la question 6 activité 3 pour préparer une variable **SMARTY** nommée `films`. Dans la boucle `foreach` de la page `filmo.tpl`, vous injecterez les données d'information :

```

{foreach from=$films item=film}
    <section class="film ">
        ... à compléter
    </section>
{/foreach}

```

Activité 5 (pour le pros – donc tout le monde)

I:/_xampp/htdocs/dev.fande.fr/apps/templates/filmo.tpl

- Le nombre d'étoiles affichées doit correspondre à l'information de la base de données (champ `rates`).
- L'affichage de l'information `shortdesc` est tronqué, au départ, à 70 caractères (par exemple). Vous pouvez ajouter un bouton `...` qui lorsqu'il est cliqué affiche la totalité du texte. Ce traitement doit être écrit en JavaScript. La difficulté est de produire une structure HTML qui permette au JavaScript de faire le traitement. Il faut envisager de présenter les 70 premiers caractères du texte dans un `span.show` et le reste du texte dans un `span.hidden` (la classe `.hidden` existe sous TWB). Cliquer sur le bouton consiste alors à ré-afficher la partie du texte initialement invisible.