

# TP : Intégration en PHP

*DWWM 2022-2023*

## Présentation

Apprendre l'inclusion de fichier et de variable PHP dans un template HTML.

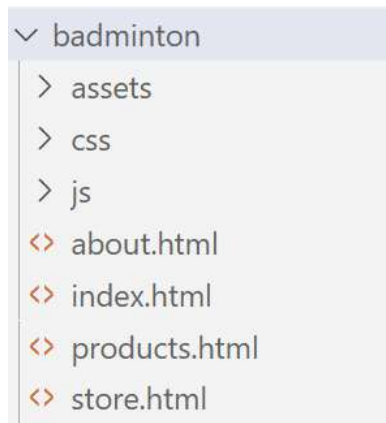
## Objectif

Passer d'un template bootstrap à un site en php. Sans toucher au CSS. On ne modifiera que le contenu. (Texte et image)

Template de base : <https://startbootstrap.com/previews/business-casual>

## Préparation

1. Télécharger le template. Vous obtenez un fichier **startbootstrap-business-casual-gh-pages.zip**. Il faudra l'extraire.
2. Ensuite, il faut renommer le dossier "**badminton**" pour obtenir la structure suivante



3. Changer l'extension des fichiers suivants en **.php** au lieu de .html

- index.html => index.php
- store.html => store.php
- about.html => about.php
- products.html => products.php

Remarque :

Comme nous modifions les extensions, les liens de la barre de navigation ne fonctionnent plus.

## Travail à faire :

1- Changer les extensions de fichier dans la barre de navigation pour qu'on puisse naviguer de nouveau correctement sur le site.

Remarque, vous pouvez faire une recherche et remplacer .html par .php

2- Changer le background. supprimer l'actuel fichier `assets/img/bg.jpg` et remplacez là par une autre image sur le thème du badminton, en la renommant `bg.jpg`

Je vous propose cette image par exemple : <https://pixabay.com/images/id-92411/>

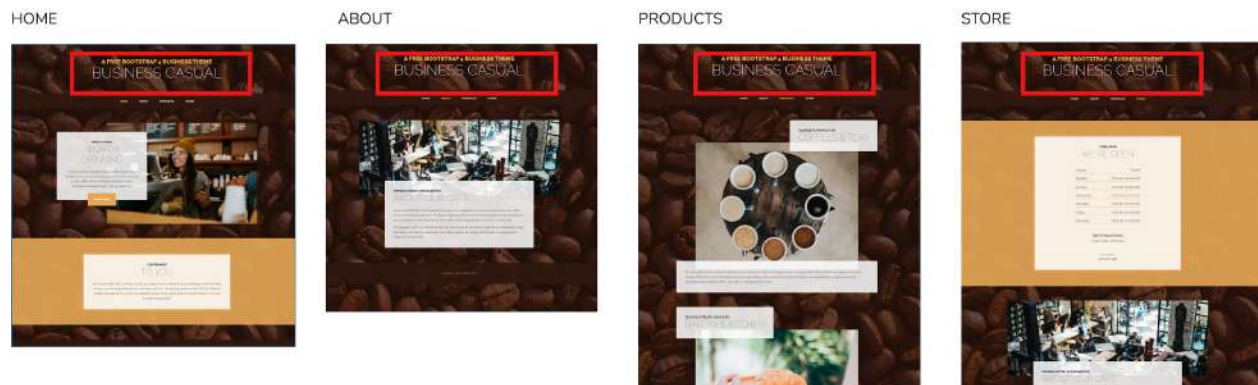
## Découpage du code HTML

### Faire un changement

Nous allons commencer par modifier le titre du site. Modifier le titre pour celui-là :

**Gréta Badminton** .Et le sous-titre (qui est au-dessus en jaune! ) par : **TP**  
intégration en PHP

On se rend compte que, malheureusement, nous sommes obligés de le modifier sur chaque page une par une.



Ici, il n'y a pas seulement 4 pages, mais imaginez que le site contienne 10 pages, 50 pages, 100 pages...

### Inclure de l'HTML

*Utilisation de la fonction include( )*

On se rend compte que <header> de la page est **commune à toutes les pages** :



Créer un nouveau fichier ( header.php) dans un dossier **common/** et copier le code du titre <h1> dans le fichier header.php

**header.php**



Attention : on copie l'extérieur de la balise <header> et je modifie les titres, si ce n'est pas déjà fait.

Maintenant, au lieu de recopier le code sur chaque page, on va inclure notre fichier **header.php** dans le fichier **index.php**. Il faut l'inclure juste après le body, à la place de l'ancien <header>

index.php

```
<body>
  <?php
    include 'common/header.php' ;
  ?>
  <!-- Navigation-->
```

## Explication :

On ouvre les balises PHP. On fait appelle à la fonction include de php qui permet d'inclure le contenu d'un fichier. La fonction **include()** inclut et exécute le fichier spécifié en argument. Voir la doc : <https://www.php.net/manual/fr/function.include.php>

Le code contenu dans header.php sera comme "copier-coller" à l'endroit où se trouve le code : **include 'common/header.php' ;**

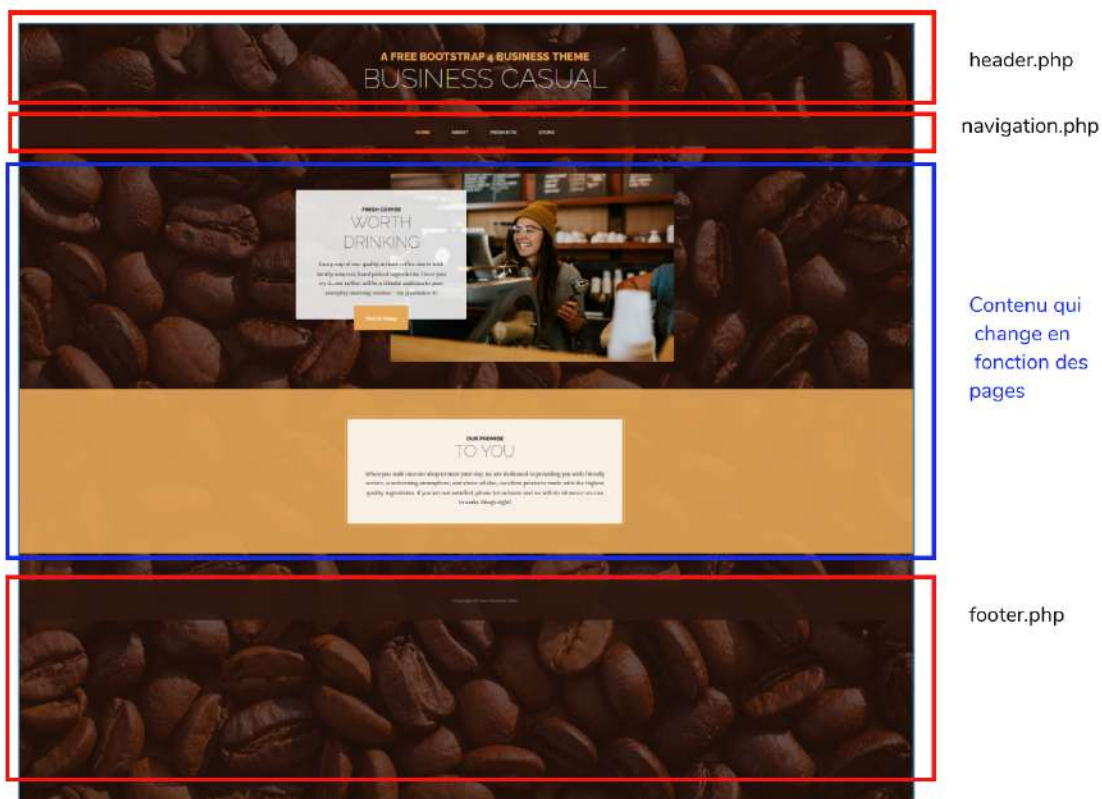
L'avantage, c'est que maintenant, on peut copier l'inclusion du header sur les autres pages autant de fois que l'on veut. Toute modification faite dans le **header.php** sera automatiquement répercuté partout.

**Attention** : on ne peut pas utiliser la fonction include dans un fichier HTML. Il est important que l'extension du fichier soit .php pour que le php, soit interprété.

## Travail à faire :

1. Inclure le header sur les pages **about.php**, **products.php** et **store.php**
2. Modifiez le titre pour "Greta Badminton Club" dans **header.php**, vous remarquerez que le titre est changé sur toutes les pages.
3. Faire la même chose pour la navigation dans un fichier **navigation.php**
4. Faire de même pour le pied-de-page dans un fichier **footer.php**

*Remarque : Vérifiez bien à chaque changement que tout n'est pas cassé. Le site doit garder la même apparence... pour l'instant.*



## Contenu des pages

Le contenu de chaque page est différent selon la page où on se trouve. C'est pour cela que nous n'allons pas le ranger dans le dossier commun.

Dans un dossier **content/**, créer les fichiers :

- index\_content.php
- about\_content.php
- store\_content.php
- product\_content.php

Chacun de ces fichiers va contenir la partie spécifique de la page dont elle est le contenu.

par exemple index\_content.php contient le code suivant

```

60 <section class="page-section clearfix">
61   <div class="container">
62     <div class="intro">
63       
64       <div class="intro-text left-0 text-center bg-faded p-5 rounded">
65         <h2 class="section-heading mb-4">
66           <span class="section-heading-upper">Fresh Coffee</span>
67           <span class="section-heading-lower">Worth Drinking</span>
68         </h2>
69         <p class="mb-3">Every cup of our quality artisan coffee starts with locally sourced, hand picked ingredients. Once you try it, our coffee will be a blissful
70         </p>
71         <div class="intro-button mx-auto">
72           <a class="btn btn-primary btn-xl" href="#">Visit Us Today</a>
73         </div>
74       </div>
75     </div>
76   </div>
77 </section>
78
79 <section class="page-section cta">
80   <div class="container">
81     <div class="row">
82       <div class="col-xl-9 mx-auto">
83         <div class="cta-inner text-center rounded">
84           <h2 class="section-heading mb-4">
85             <span class="section-heading-upper">Our Promise</span>
86             <span class="section-heading-lower">To You</span>
87           </h2>
88           <p class="mb-0">When you walk into our shop to start your day, we are dedicated to providing you with friendly service, a welcoming atmosphere, and above
89           </p>
90         </div>
91       </div>
92     </div>
93   </section>
94

```

C'est-à-dire les deux sections :

`<section class="page-section clearfix">...`

et

`<section class="page-section cta">...`

Donc, vous COUPEZ ce code pour le mettre dans le fichier **index\_content.php** et dans le fichier **index.php**, le code se résume à ceci :

```
<?php
include './content/index_content.php';
?>
```


Vous faites une inclusion comme précédemment. La seule différence est que le fichier n'est pas rangé au même endroit. Le tout est de l'inclure au bon endroit.

Nous obtenons ceci, dans **index.php**

```
<body>
  <?php
  include './common/header.php';
  ?>
  <?php
  include './common/navigation.php';
  ?>
  <?php
  include './content/index_content.php';
  ?>
  <?php
  include './common/footer.php';
  ?>
  <!-- Bootstrap core JS-->
```



Remarque : Nous avons finalement quatre lignes de PHP, nous pouvons faire qu'une seule ouverte fermeture des balises `<?php ... ?>`



```
<body>
  <?php
    include './common/header.php';
    include './common/navigation.php';
    include './content/index_content.php';
    include './common/footer.php';
  ?>
  <!-- Bootstrap core JS-->
```

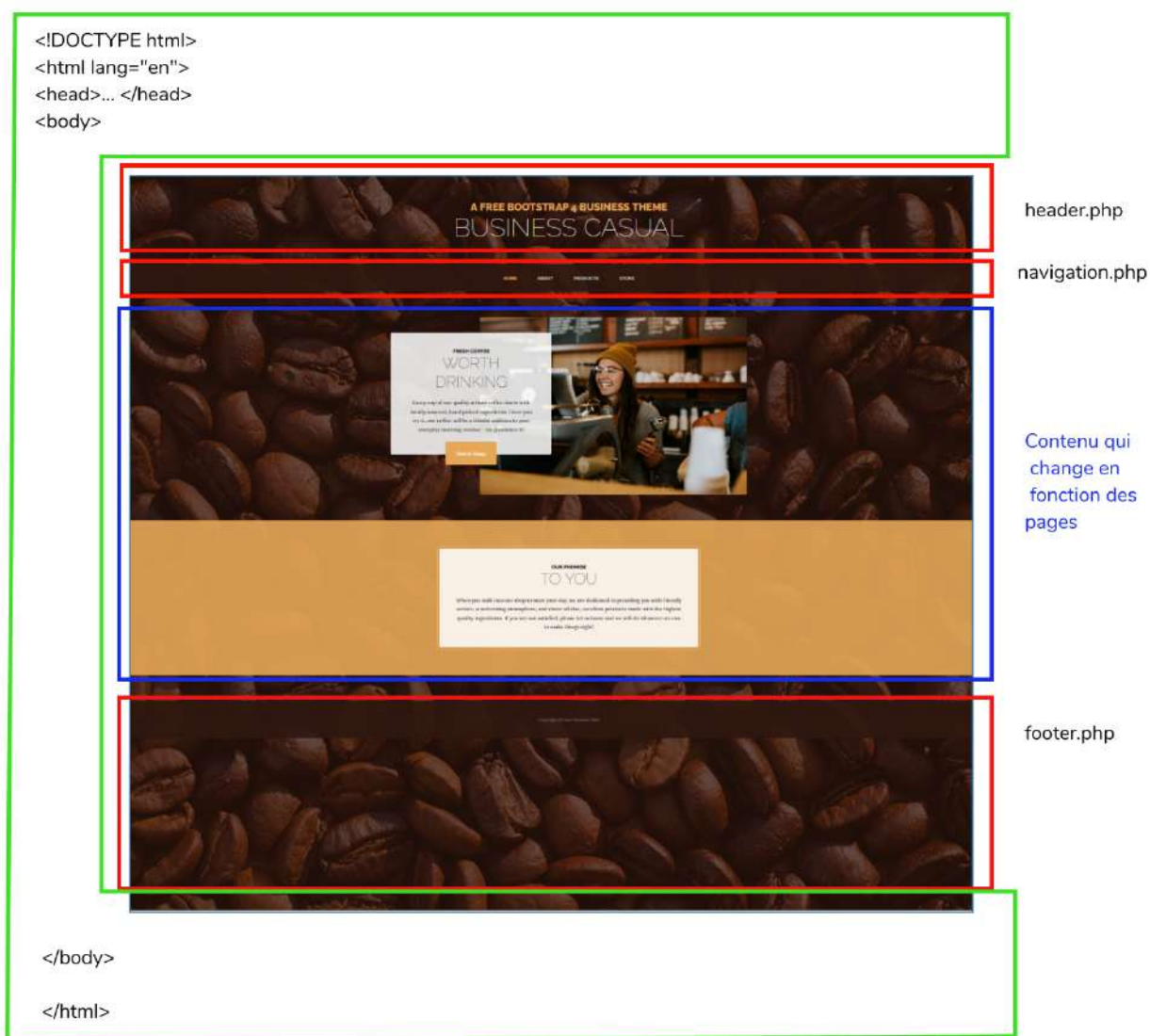
## Travail à faire

Découper les contenus de toutes les pages et les inclure au bon endroit pour que le site reste fonctionnel. Attention de bien rester organisé.

Maintenant que tout est découpé, nous allons pouvoir les manipuler à notre guise.

## Template

Désormais, au lieu d'inclure les parties communes, nous allons garder la structure commune dans un fichier **template.php** et inclure la partie qui change. C'est dans ce fichier que l'on décidera de la structure de la page HTML. Pour commencer, faites une copie du fichier **index.php** et nommer le **template.php**



**template.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
<?php include('common/head.php' );
</head>
<body>
<?php
    include('common/header.php' );

    include( 'common/navigation.php' );

    include( 'content/index_content.php' );

    include( 'common/footer.php' );
?>
<!-- Bootstrap core JavaScript -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

</body>

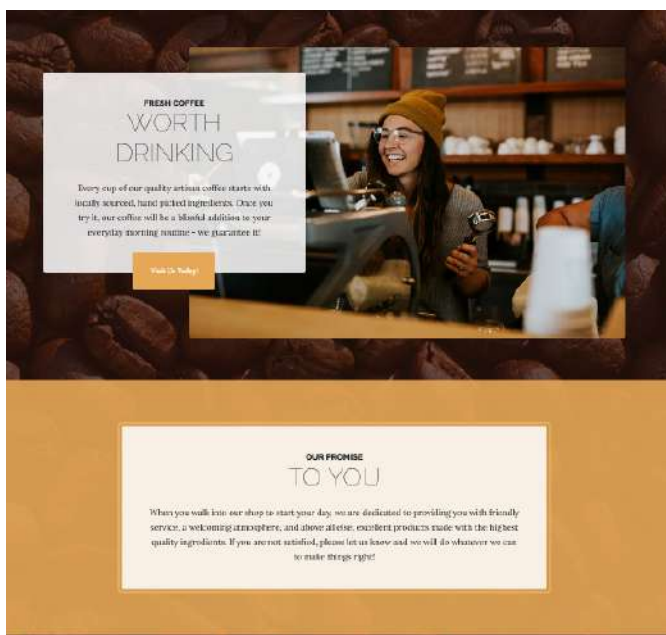
</html>
```

Chaque code est de la couleur de sa partie correspondante à l'image au-dessus.

- En vert, la structure invisible de la page HTML
- En rouge, les parties communes
- En bleu, ce qu'il change à chaque page.

À partir de ce fichier template.php, nous allons être capable de d'afficher toutes les pages. Pour cela, nous allons ajouter un peu de logique.

L'algorithme est simple :



Si

nous sommes sur la page  
"home"

ALORS

il faut inclure le contenu  
d'index\_content.php



Si

nous sommes sur la page  
"about"

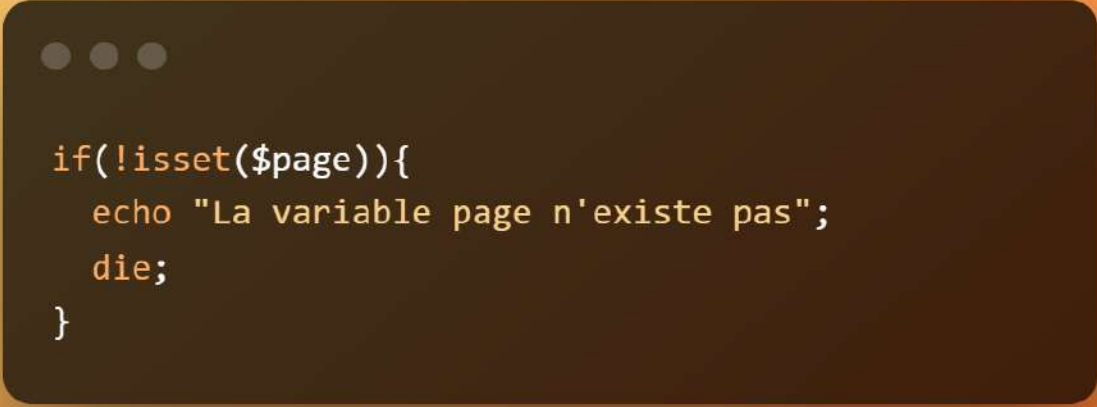
ALORS

il faut inclure le contenu  
d'about\_content.php

Et ainsi de suite.... Supposons que le nom de la page soit sauvegardé dans la variable `$page`.

```
if($page == 'home'){
    include './content/index_content.php';
}
if ($page == 'about') {
    include './content/about_content.php';
}
if ($page == 'products') {
    include './content/products_content.php';
}
if($page == 'store'){
    include './content/store_content.php';
}
```

Attention, si la variable `$page` n'existe pas, nous allons avoir une erreur. Il faut prévoir le cas.



```
if(!isset($page)){  
    echo "La variable page n'existe pas";  
    die;  
}
```

la fonction **die** permet d'arrêter le script.

Le contenu de la balise <body> doit ressembler à ceci :

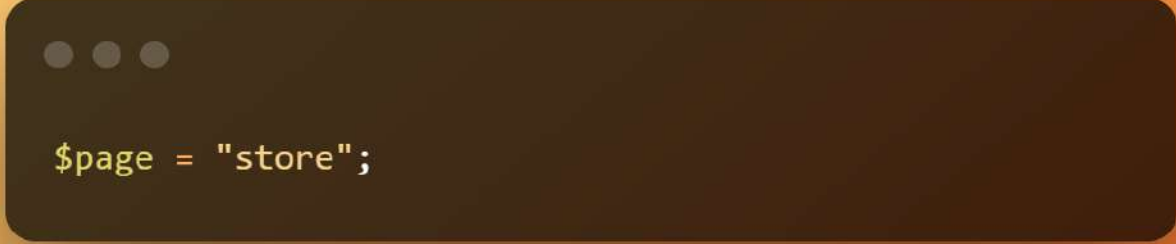
```
<body>
    <?php
    include './common/header.php';
    include './common/navigation.php';
    if(!isset($page)){
        echo "La variable page n'existe pas";
        die;
    }
    if($page == 'home'){
        include './content/index_content.php';
    }
    if ($page == 'about') {
        include './content/about_content.php';
    }
    if ($page == 'products') {
        include './content/products_content.php';
    }
    if($page == 'store'){
        include './content/store_content.php';
    }
    include './common/footer.php';
    ?>
    <!-- Bootstrap core JS-->
```

## Travail à faire

Le code ci-dessus n'est pas optimisé. Quel autre structure logique pouvez-vous utiliser ? Faites des modifications sur le code pour **l'optimiser** en gardant le même fonctionnement.

Nous décidons que même si la variable n'existe pas, nous affichons quand même la page home. Il s'agit d'un affichage par défaut. Faites la modification pour que cela soit le cas.

Pour faire des tests, vous pouvez ajouter une valeur à `$page` juste avant l'algorithme. Toujours dans les balises `<?php ... ?>`.



```
$page = "store";
```



## Comment savoir où nous sommes ?

Autrement dit, Comment savoir sur quelle page-t-on se trouve ?

### Utilisation d'une super globale \$\_GET

Nous allons transmettre l'information, via une variable transmise par l'url.

Ceci est rendu possible par la super globale : **\$\_GET**

Dans l'url : <.....template.php?page=home>

Modifiez le code php à l'intérieur du fichier **template.php** :



```
$page = $_GET['page'];
```

On vérifie dans un premier temps qu'on récupère bien la variable. Éventuellement, avec un `echo $page;`

Si c'est OK... Maintenant que nous savons où nous sommes, nous n'avons plus qu'à vérifier que notre algorithme précédent fonctionne :

<.....template.php?page=home>

<.....template.php?page=about>

<.....template.php?page=store>

<.....template.php?page=products>

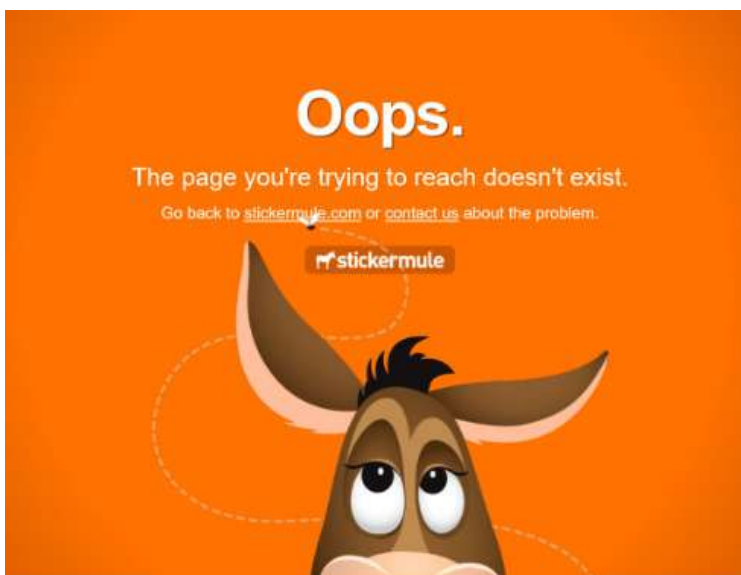
## Travail à faire

Ceci s'appelle un **contrôleur de contenu**. La page est transmise par l'url, par l'intermédiaire du tableau global `$_GET`. Puis en fonction de cette variable, et donc en fonction de la page sur laquelle, on se trouve, on affiche un contenu ou un autre.

Il est de coutume de prévoir un affichage par défaut, s'il la page n'est pas trouvée.

Créer une page 404 qui sera afficher dans le cas où la valeur de page n'est ni vide ni l'une des valeurs suivantes : home, store, products, about

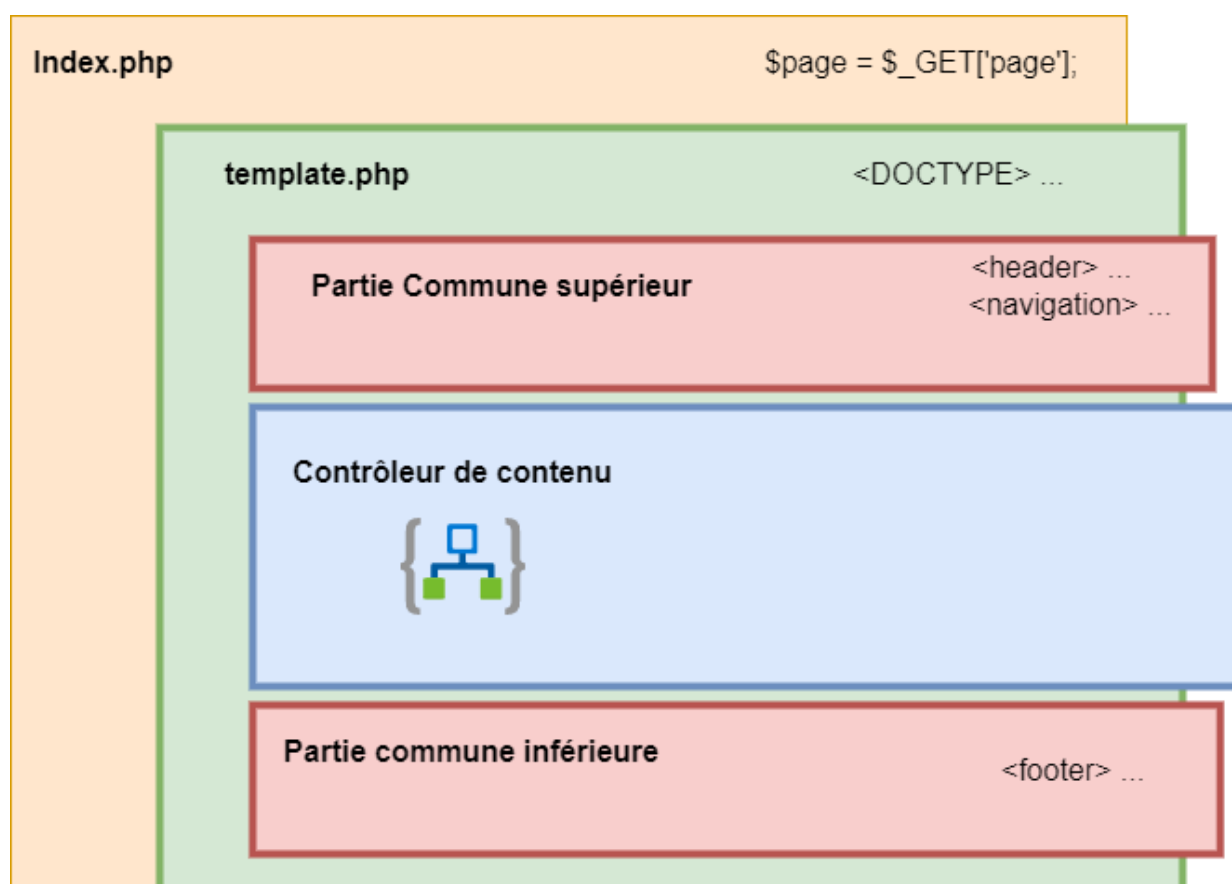
Exemple



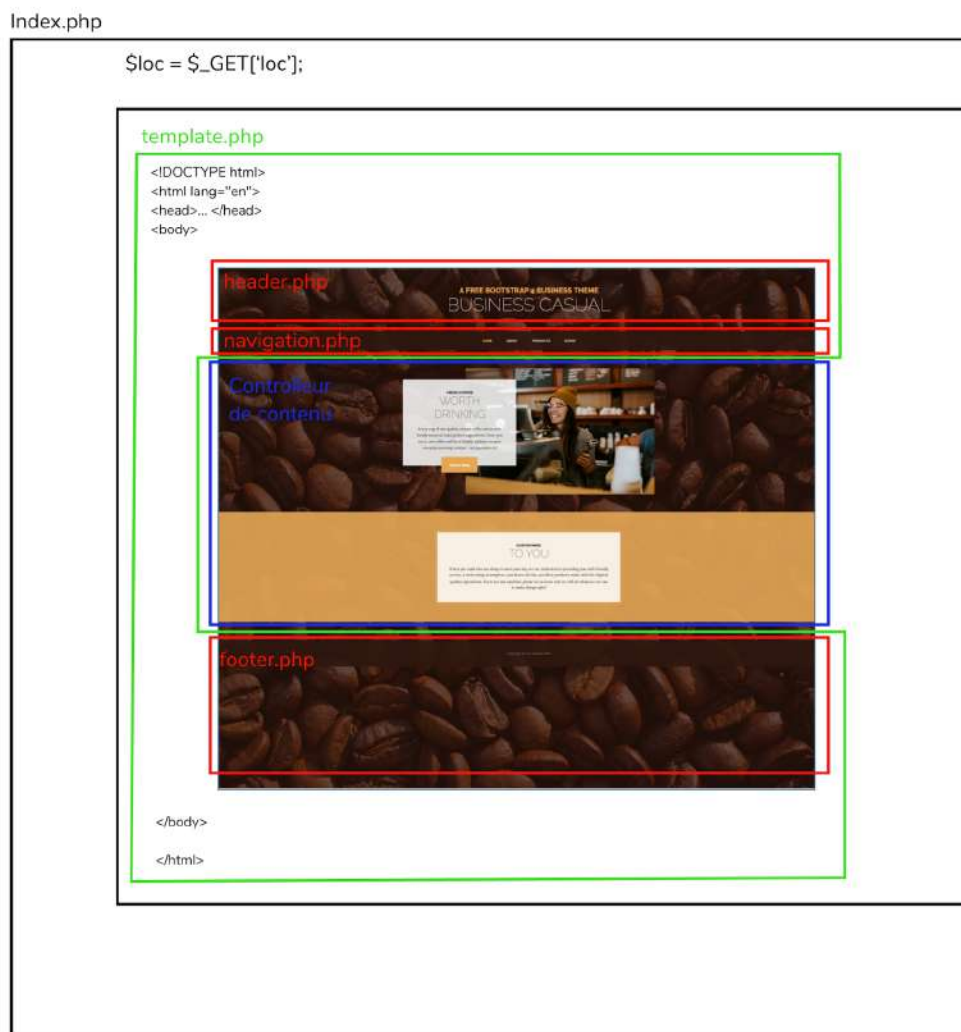
## Navigation

Il est coutume de passer toujours par le fichier `index.php`. C'est le **contrôleur général**. Il permet de contrôler les données avant d'afficher le code HTML. C'est ici que sont fait les appels à la session, aux données, etc. Pour l'instant, la seule donnée que nous avons, c'est le nom de la page. Nous allons donc tout faire passer par **`index.php`** qui va inclure le fichier **`template.php`**. Pour rappel, le fichier **`template.php`** est capable d'afficher le contenu HTML de n'importe quel page.

Il faut bien comprendre cette architecture.



Dans notre cas :



NB : remplacer 'loc' par 'page'

## Travail à faire

Modifier les liens du menu :

1. <...badminton/index.php?page=home>
2. <...badminton/index.php?page=about>
3. <...badminton/index.php?page=products>

#### 4. <...badminton/index.php?page=store>

L'architecture de votre projet doit ressembler à cela :

```
index.php // à la racine
common/
-- template.php
-- header.php
-- footer.php
-- navigation.php
content/
-- index_content.php
-- about_content.php
-- products_content.php
-- store_content.php
-- 404.html
```

NB: je choisis de mettre le template dans les parties communes. J'aurais pu le laisser à la racine. Il faut juste vérifier que les chemins des fichiers soient corrects.

Voici, le code “découpé” :



NB: Pour l'instant, il ne contient que cela... Mais plus tard, il pourra contenir plus de logique et plus de données. Par exemple, on y placera nos requêtes SQL.

```

template.php

<!DOCTYPE html>
<html lang="fr">

<head>
...
</head>

<body>
  <?php
    include './common/header.php';
    include './common/navigation.php';
    if (!isset($page)) {
      $page = "home";
    }
    switch ($page) {
      case 'home':
        include './content/index_content.php';
        break;
      case 'about':
        include './content/about_content.php';
        break;
      case 'products':
        include './content/products_content.php';
        break;
      case 'store':
        include './content/store_content.php';
        break;
      default:
        include './content/404.html';
        break;
    }
    include './common/footer.php';
  ?>
  <!-- Bootstrap core JS-->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
  <!-- Core theme JS-->
  <script src="js/scripts.js"></script>
</body>

</html>

```

NB : J'ai apporté quelques modifications au code, mais vous devez normalement le comprendre; Ceci est une correction du "travail à faire" précédent;

Les anciens fichiers "about.php", "store.php", "products.php" peuvent être supprimés. Vérifiez bien que le site fonctionne correctement.

## Do not Access Superglobal Directly

Tous les fichiers passent par le même fichier **index.php**. On l'appelle le **contrôleur général**. Le fichier **template.php** donne le squelette HTML de vos pages. il doit être inclus en dernier dans le fichier **index.php**. Avant cela, le contrôleur va contrôler les données. Par exemple : Nous allons contrôler la valeur de `$page`.

```
<?php
$page = $_GET['page'];
```

Selon l'éditeur de code que l'on utilise, vous pouvez avoir ce message d'erreur :

```
Do not Access Superglobal $_GET Array Directly.

Use some filtering functions instead (e.g. filter_input(), conditions with is_*() functions, etc.).
```

Il est déconseillé d'accéder directement à une Superglobale. Il faut vérifier qu'elle existe et qu'elle n'est pas vide. Pour cela, on utilise une fonction de filtrage.

```
<?php
$page = filter_input(INPUT_GET, "page");
```

Cette fonction va permettre de vérifier sur la **clé** "page" existe bien dans le **tableau** `$_GET` et qu'elle n'est pas vide.

Le 1er paramètre est une constante `INPUT_GET` précise qu'il faut aller chercher dans `$_GET`

On peut ajouter une option en 3<sup>e</sup> paramètre pour préciser le type attendu.

```
$page = filter_input(INPUT_GET, "page", FILTER_SANITIZE_STRING);
```

Par exemple :

**FILTER\_SANITIZE\_STRING** : Supprime les balises, et supprime ou encode les caractères spéciaux.

Pour plus d'information :

<https://www.php.net/manual/fr/filter.filters.sanitize.php>

**FILTER\_VALIDATE\_EMAIL** : Valide un email

Pour plus d'information : <https://www.php.net/manual/fr/filter.filters.validate.php>



# Application

## Partie 1

Modifier le <footer> sur toutes les pages, pour y ajouter votre nom, prénom

## Partie 2

Remplacer les images et les titres pour vous approprier complètement le template. Modifier aussi la barre de navigation et les titres des onglets.

Voici les consignes :

- La page **Home** devient la page **Actu**
  - section 1 - Tournois Inter-secrétaires
  - section 2 - Espoir Féminin, Isabella Radenos
- La page **About** devient la page **Equipe**
  - une seule section "à propos"  
Prendre une photo de la promo. Demander à votre formateur de prendre une photo grand angle pour respecter les distances sanitaires. En titre : Développeur Web - Web Mobile, Promo 2022-23



- La page **Products** devient la page **Stages**
  - section 1 : Plein air, stage du printemps
  - section 2 : Démonstration de champion, Pandrine Saul
  - section 3 : Stage Grau-du-Roi, "On the beach"
- La page **Store** devient la page **Créneaux**
  - section 1 : planning ( voir partie 3)
  - section 2 : c'est de nouveau la section "a propos". Comment faire pour réutiliser le code déjà fait à la page **Equipe**.

## Partie 3

Utiliser une boucle sur le tableau suivant pour la page Créneaux. Le tableau devra être instancié dans le contrôleur général.

```
$array = [  
    "lundi" => "18h à 22h",  
    "mardi" => "18h à 20h",  
    "mercredi" => "16h30 à 18h",  
    "jeudi" => "19h30 à 22h",  
    "vendredi" => "18h à 22h",  
];
```

Rendu attendu :

PLANING	
JEU LIBRE	
Lundi	18h à 22h
Mardi	18h à 20h
Mercredi	16h30 à 18h
Jeudi	19h30 à 22h
Vendredi	18h à 22h
Gymnase Jean Mermoz	
717 Av. Jean Mermoz, 34000 Montpellier	
Contact	
(+33) 6.45.78.32.45	

**Question bonus** : faire en sorte que le jour actuel soit en jaune comme le mardi dans l'exemple ci-dessus.

### Partie 3

Intégrer une carte interactive en dessous de l'adresse du gymnase. On pourra faire un zoom dessus.