

# Projet P2 : Systèmes d'exploitation PHP

Conrad HILLAIRET et Alexandre VIEIRA

5 mai 2013

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Présentation du PHP</b>	<b>3</b>
1.1 Fonctionnement global du PHP	3
1.2 A quoi ressemble ce langage ?	3
1.2.1 Les variables	4
1.2.2 Les boucles, les conditionnelles	4
1.2.3 Les fonctions intégrées	5
1.3 Les bases de données	5
<b>2 Présentation de l'implémentation</b>	<b>8</b>
2.1 index.php	8
2.2 formul_news.php	9
2.3 enreg_news.php	11
2.4 contact.php	12
2.5 envoi_mail.php	13
<b>Conclusion</b>	<b>14</b>
<b>A Bibliographie</b>	<b>15</b>
<b>B Crédit d'illustration</b>	<b>15</b>
<b>C Annexes</b>	<b>16</b>
C.1 Feuilles CSS	16
C.2 entete.php et piedDePage.php	22

## Introduction

Le but de ce projet était de nous faire découvrir le langage PHP et d'approcher un peu la gestion d'une base de données.

Le PHP est un langage de script libre inventé en 1994, utilisé principalement sur les serveurs HTTP pour créer des sites internet dynamiques. A la base impératif, ce langage intègre également des fonctionnalités tirées des modèles objets, comme nous le verrons dans les implémentations faites un peu plus tard.

Un autre avantage du PHP est la possibilité qu'offre ce langage d'utiliser une base de données. Ce point n'est pas le plus développé dans ce projet, mais il est clairement utilisé via un autre langage, le SQL.

Les possibilités offertes par ce langage sont donc très nombreuses et permettent de créer un lien direct entre l'utilisateur et le site internet. C'est ce lien que nous avons voulu montrer dans ce projet, en développant principalement trois grands axes :

1. Création d'une page affichant différentes annonces stockées dans une base de données
2. Création d'un formulaire (dont l'accès est protégé par un mot de passe) permettant d'ajouter une annonce dans la base de données
3. Création d'une page de contact

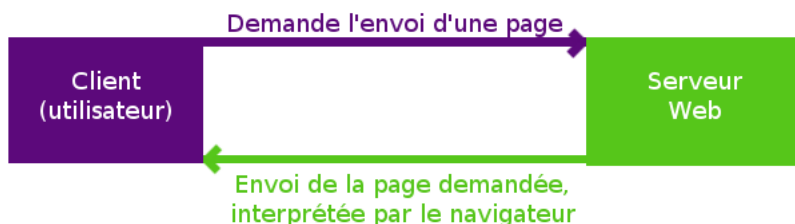
Ce dossier présentera d'abord plus en détail le fonctionnement du PHP, puis s'attardera un peu plus sur les différentes implémentations que nous avons effectuées.

# 1 Présentation du PHP

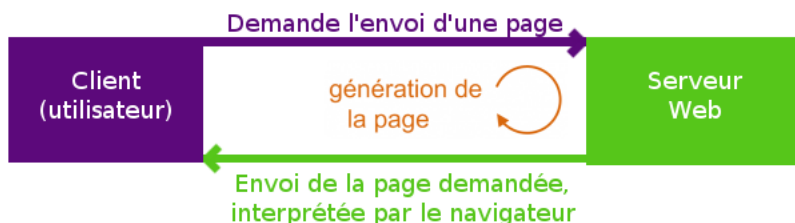
Comme nous venons de le dire, le PHP est un langage de script. Cependant, son fonctionnement est un peu plus complexe que ce qu'on peut connaître avec le Bash ou le CSH, où l'interprétation du code se fait directement sur la machine. Car après tout, on doit tout de même générer une page compréhensible par un navigateur internet ! C'est ce qu'on va expliquer dans une première sous-partie. On s'intéressera ensuite aux quelques spécificités du codage en PHP, avant de s'intéresser aux bases de données.

## 1.1 Fonctionnement global du PHP

Pour bien comprendre la logique qui se cache derrière le PHP, le mieux est encore de l'illustrer par des schémas, et surtout, de faire le parallèle avec le HTML/CSS. En gros, le fonctionnement d'un site classique fonctionne, schématiquement, ainsi :



Dans ce cas, la page renvoyée est en HTML/CSS. C'est le code source, qui est interprété par le navigateur. Il nous affiche alors la page telle qu'il la reçoit, ou plutôt telle qu'il la comprend. Cependant, le navigateur ne peut pas interpréter le PHP. Il agit donc au niveau du serveur, ainsi :



La page renvoyée est toujours en HTML/CSS, mais elle a été générée par le serveur auparavant. Cela permet donc de faire des pages dynamiques, qui peuvent être modifiées en fonction de nombreux paramètres !

Le serveur n'étant rien de plus qu'un gros ordinateur, il peut facilement faire différents calculs, stocker des variables, et surtout les utiliser. La page envoyée au client est donc remodelée à chaque envoi, et le tout est totalement transparent aux yeux de l'utilisateur.

## 1.2 A quoi ressemble ce langage ?

Il faut d'abord voir une chose : le PHP s'utilise en plein milieu de codes HTML. Pour l'utiliser, il faut donc, au préalable, connaître un minimum le langage HTML.

Pour différencier le code HTML du code en PHP que le serveur doit interpréter, le code en PHP est placé entre deux balises : `<?php` et `?>`.

Ces balises peuvent être insérées n'importe où dans le code HTML, et ne seront plus affichées dans le code HTML final, envoyé à l'utilisateur.

Ensuite, il faut bien pouvoir dire au serveur ce qu'il doit générer ! On a pour cela, comme dans bien d'autres langages, accès à l'utilisation de variables, l'implémentation de boucles et de conditionnelles, ou encore de fonctions intégrées.

### 1.2.1 Les variables

Elles se distinguent très facilement du reste du code par la présence du caractère \$ avant l'identifiant de ces dernières. Comme en bash par exemple, le type des variables n'a pas à être précisé avant d'utiliser la variable, ni même de la déclarer. Le type est déterminé à l'affectation et peut être changé à tout moment.

L'affectation se fait d'ailleurs via le signe "=".

### 1.2.2 Les boucles, les conditionnelles

A ce niveau, les choses ressemblent beaucoup au C. En commençant par les conditionnelles, on a deux syntaxes possibles, qui sont sensiblement les mêmes qu'en C.

- **if...else...** dont la syntaxe est :

```
if (condition1)
{
    \\Suite d'instructions si condition1 est vraie
}
elseif (condition2)
{
    \\Suite d'instructions si condition1 est fausse et
    condition2 est vraie
}
else
{
    \\Suite d'instructions si condition1 et condition2
    sont fausses
}
```

Les conditions sont construites de la même façon qu'en C, avec ! pour le NON, && pour le ET, || pour le OU, et ainsi de suite.

- **switch** qui permet de construire des conditionnelles plus complexes :

```
switch(variable)
{
    case valeur1 : //Si variable vaut une certaine valeur
        Instructions;
    case valeur2 :
        Instructions;
    ...
    default //Si aucune des conditions ne sont remplies au-dessus
        Instructions;
}
```

Les boucles ont une syntaxe qui est également sensiblement la même qu'en C. Elle se font ainsi :

- **while** pour faire une boucle indéterministe :

```
while (condition)
{
    //Instructions a realiser tant que condition est vrai
}
```

- **for** pour faire une boucle déterministe :

```
for ($i=0,$i<$n,$i++)
{
    //Instructions a realiser
}
```

### 1.2.3 Les fonctions intégrées

Elles sont très nombreuses, et toutes les lister ici serait superflu. Cependant, on remarque encore une fois un lien très proche avec d'autres langages. En voici quelques exemples :

- **echo** qui permet d'afficher telle quelle une chaîne de caractères ou une variable (qu'on retrouve en bash)
- **strlen** ou **strtolower** qui renvoie la longueur d'une chaîne dans le premier cas, et qui convertit tous les caractères d'une chaîne en minuscule dans le deuxième cas (qu'on retrouve en C par exemple)
- Plusieurs fonctions mathématiques, telles que **cos** ou **rand** (qu'on retrouve également en C)

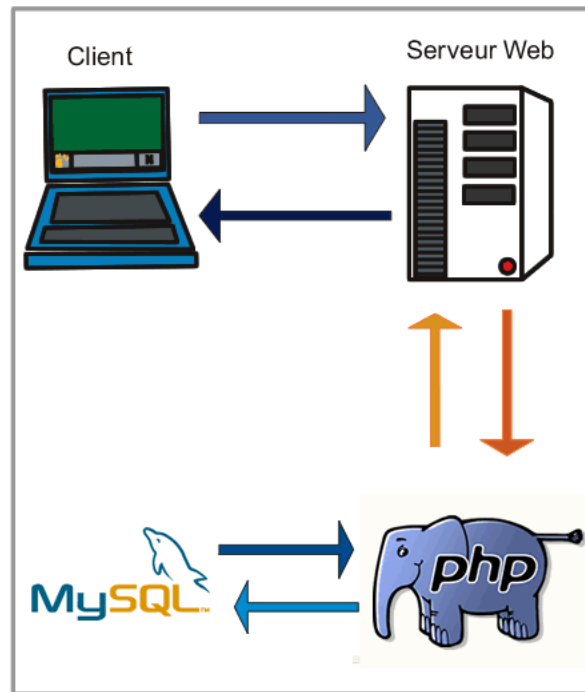
## 1.3 Les bases de données

Pour créer des sites dynamiques, les serveurs ont bien souvent besoin d'enregistrer des informations de façon durable. C'est là qu'interviennent les bases de données. On utilise pour cela plusieurs programmes, parmi lesquels :

- MySQL
- PostgreSQL
- Oracle
- Et bien d'autres encore

Ces programmes s'appellent des SGBD, pour Systèmes de Gestion de Bases de Données. Tous ces SGBD utilisent le même langage pour faire le lien entre le code PHP et la base de données : le SQL. Ce qui change le plus d'un SGBD à l'autre est surtout la manière dont les données seront organisées. Dans notre projet, nous avons choisi d'utiliser MySQL.

On peut se demander à quoi sert le PHP dans cette histoire. Il fait en fait le lien entre le développeur et le SGBD. Un schéma valant mieux que de grands mots :



Le serveur, utilisant PHP, interprète le script qui lui est donné. Il rencontre, à un moment, une ligne de SQL. PHP se charge alors de demander à MySQL de traiter l'information. MySQL traite la commande, et renvoie le résultat à PHP, qui se charge de la traiter.

A présent, voyons comment s'organise une base de données.

On distingue trois grandes catégories dans cette organisation :

- Il y a tout d'abord **la base**, qui est la plus haute "boîte" qu'on peut concevoir
- Une base est constituée **d'une ou plusieurs tables**, qui permettent de classer les informations suivant différents critères (le plus souvent, suivant l'utilisation qu'on en fera)
- Chaque table est constituée de **champs** et **d'entrées**, un peu comme les colonnes et les lignes d'un tableau.

A chaque fois qu'on utilise une base de données, on doit connaître toutes ces informations pour pouvoir correctement l'utiliser.

On utilise, entre le PHP et le MySQL, différentes fonctions :

- On commence par se connecter à la base de données avec la commande suivante :

```
$bdd = new PDO('mysql:host=NomDeLHost;dbname=NomDeLaBase', 'NomDUtilisateur', 'MotDePasse');
```

On reconnaît là le style de la programmation par objet. On crée en effet via cette commande un objet, qui nous permettra par la suite de pouvoir extraire des données de cette base.

- On souhaite, ensuite, récupérer des champs dans une table. On utilise pour cela l'objet précédent :

```
$reponse = $bdd -> query(SELECT nomDuChamp1 nomDuChamp2 ... FROM NomDeLaTable
WHERE ConditionSurUnChamp)
```

On peut remplacer les nomDuChamp par \* pour sélectionner tous les champs.

Les commandes comme WHERE forment des critères de sélection. On a, par exemple :

- WHERE permet de sélectionner seulement les champs qui nous intéressent
- ORDER BY permet de choisir un champ sur lequel les données seront triées (par ordre alphabétique ou numérique). On rajoute le mot clé DESC si on veut un tri par ordre décroissant
- ou encore LIMIT qui permet de limiter le nombre de champ sélectionné

- On récupère enfin les données, entrée par entrée, grâce à l’instruction suivante :

```
$donnees = $reponse->fetch();
```

On utilise souvent cette instruction dans une boucle while, car on récupère les données entrée par entrée, et on peut les afficher ensuite comme on les avait ordonné lors de la récupération des données.

- On veut aussi pouvoir ajouter des données dans la base de données. On utilise pour cela :

```
$bdd->exec('INSERT INTO NomTable(NomDesChamps)
VALUES(ValeurAAffecterAuxChamps)');
```

Cependant, on peut aussi vouloir y rentrer des données variables. On procède pour cela ainsi :

```
$req = $bdd->prepare('INSERT INTO NomTable(NomDesChamps)
VALUES(:identifiants)');
```

```
$req->execute(array('identifiant' => $ValeurVariable,));
```

Bien sûr, on doit avoir autant de champs que de valeurs affectées ou d’identifiants, et toujours dans le même ordre pour que l’affectation se fasse bien.

## 2 Présentation de l'implémentation

Nous avons, dans le cadre de ce projet, réaliser un site web réalisant différentes tâches réalisables en PHP :

- Réaliser une page présentant un ensemble de news enregistrées dans une base de données
- Créer un formulaire d'ajout de news
- Créer un formulaire de contact

Le tout est arrangé dans du code HTML pour la mise en page, avec deux feuilles CSS (présentées en Annexe). On inclut également trois pages php qui ne servent qu'à la mise en page, entete.php, menu.php et piedDePage.php, toutes trois également mises en Annexe. Le résultat est visible sur la page suivante :

<http://essaissitesweb.alwaysdata.net/>

### 2.1 index.php

Cette page est la première page sur laquelle nous arrivons. Elle présente les news enregistrées dans la base de données, et permet via un formulaire d'accéder au formulaire d'enregistrement de news. Le code est le suivant :

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

<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="style.css" />
<title> Essais php – Conrad Hillairet et Alexandre Vieira</title>
</head>

<body>

<!--Chargement de l'entete-->
<?php include("entete.php"); ?>
<!--Chargement du menu-->
<?php include("menu.php"); ?>

<!--Premier bloc: affichage des news-->
<div id="news">
  <?php
  $bdd = new PDO('mysql:host=mysql2.alwaysdata.com;dbname=
    essaissitesweb_news','67162','penrchalper'); //Connexion a la base de
    donnees
  $req = $bdd->query('SELECT titre, contenu, DATE_FORMAT(date_ecriture, \'%
    d/%m/%Y\') AS date_ecr FROM articles ORDER BY date_ecriture DESC LIMIT
    0, 10'); //recuperation des donnees

  while ($donnees = $req->fetch()) //Conversion des donnees dans un format
    exploitable
  {
  ?>
    <div class="une_news">
      <h1><?php echo htmlspecialchars($donnees['titre']); ?></h1> <!--
        Affichage du titre-->
      <h2>le <?php echo $donnees['date_ecr']; ?></h2> <!--Affichage de
        la date-->
```



```

        <p>
        
        <?php echo nl2br(htmlspecialchars($donnees['contenu']));?><br></br></p>
        </div>
    <?php
    }
    ?>
</div>

<!--Second bloc: ajouter des news-->
<div id="addnews">
    <p>Renseignez le mot de passe pour accéder au formulaire de nouvelles</p>

    <form action="formul_news.php" method="post">
        <p>
            <input type="password" name="MdP" /><br>
            <input type="submit" value="Valider" />
        </p>
    </form>
</div>

<!--Chargement du pied de page-->
<?php include("piedDePage.php"); ?>

</body>

</html>

```

## 2.2 formul\_news.php

Via le formulaire de la page précédente, en entrant le bon mot de passe (qui est, sur notre site, MotDePasse), on accède à cette page de formulaire. Elle est très similaire à l'index, mais propose à la place du formulaire d'entrée de mot de passe un formulaire d'entrée de news. Chaque champ rempli est ainsi entré dans la base de données et directement affiché au-dessus.

```

<?php session_start(); //Ouverture d'une session, pour conserver le mot
    de passe
    if (isset($_POST['MdP']))
    {
        $_SESSION['MdP']=$_POST['MdP'];
    }
    ?>

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

<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="style.css" />

```

```

<title> Essais php – Conrad Hillairet et Alexandre Vieira</title>
</head>

<body>

<!--Chargement de l'entete-->
<?php include("entete.php"); ?>
<!--Chargement du menu-->
<?php include("menu.php"); ?>

<!--Premier bloc: affichage des news-->
<div id="news">
  <?php
  $bdd = new PDO('mysql:host=mysql2.alwaysdata.com;dbname=
    essaissitesweb_news','67162','penrchalper'); //Connexion a la base de
    donnees
  $req = $bdd->query('SELECT titre, contenu, DATE_FORMAT(date_ecriture, \'%
    d/%m/%Y\') AS date_ecr FROM articles ORDER BY date_ecriture DESC LIMIT
    0, 10'); //Recuperation des donnees dans la table article, avec un
    certain ordre, et un format special pour la date

  while ($donnees = $req->fetch())
  {
  ?>
    <div class="une_news">
      <h1><?php echo htmlspecialchars($donnees['titre']); ?></h1>
      <h2>le <?php echo $donnees['date_ecr']; ?></h2>
      <p>
        
        <?php echo nl2br(htmlspecialchars($donnees['contenu']));?></br></br>
        <!--nl2br : saut de ligne remplace par une balise br,
          htmlspecialchars pour que les codes ne soient pas interprete-->
      </p>
    </div>
    <?php
  }
  ?>
</div>

<!--Second bloc: ajouter des news-->
<div id="formnews">

<?php
if (isset($_GET['reussite']) AND ($_GET['reussite']==1)) //Si la variable
  est bien definie dans l'url et qu'elle vaut 1
  echo "L'article a bien ete enregistre!";
if ($_SESSION['MdP'] == "MotDePasse") //On verifie si le mot de passe est
  bon, si c'est le cas, on affiche le formulaire. Dans le cas contraire
  , on n'affiche que le mot de passe est faux.
{
  ?>

```

```

<!-- Creation du formulaire-->
<form action="enreg_news.php" method="post">
    <p>
        Titre : <input type="Text" name="titre" /><br> <!--
            Demande du titre de l'article-->
        Contenu : <br><!-- Demande du contenu de l'article-->
<textarea name="message" rows="8" cols="45">
Tapez votre article ici.</textarea>
        <br>
        <input type="submit" value="Valider" />
    </p>
</form>
<?php
}
else
    echo "Le mot de passe est incorrect<br/>";
?>
</div>

<!-- Chargement du pied de page-->
<?php include("piedDePage.php"); ?>

</body>

</html>

```

## 2.3 enreg\_news.php

Les données entrées dans la page précédente sont envoyées à ce formulaire qui les traite et les enregistre dans la base de données. On redirige ensuite automatiquement la page vers le formulaire précédent.

```

<?php
//Recuperation des donnees, sans mise en forme
$bdd = new PDO('mysql:host=mysql2.alwaysdata.com;dbname=
    essaissitesweb_news', '67162', 'penrchalper');//Changer ici pour
    coller avec les identifiants necessaires, connexion a la base de
    donnees

$sauv = $bdd->prepare('INSERT INTO articles(titre, contenu, date_ecriture
) VALUES(:titre, :contenu, NOW())'); //On prepare a rentrer de
    nouveaux champs dans la table article dans la base
$sauv->execute(array( //Recuperation des donnees
    'titre'=>$_POST['titre'],
    'contenu'=>$_POST['message'],
));

header('Location: formul_news.php?reussite=1');
?>

```

## 2.4 contact.php

Cette page propose un formulaire permettant d'envoyer directement un mail à une personne (dont l'adresse est défini dans le script suivant). Cela permet d'automatiser l'envoi de message sans passer par un système de messagerie.

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

<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="style2.css" />
<title> Essais php – Conrad Hillairet et Alexandre Vieira</title>
</head>

<body>

<!--Chargement de l'entete-->
<?php include("entete.php"); ?>
<!--Chargement du menu -->
<?php include("menu.php"); ?>

<!--Definition du corps de la page-->
<div id="corps">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<?php
if (isset($_GET['reussite'])) //Si la variable est bien definie dans l'
    URL
{
    if ($_GET['reussite']==1)
        echo "Le message a bien ete envoye!";
    else
        echo "Le message n'a pas pu etre envoye.";
}
?>
<!--Creation du formulaire-->
<form action="envoi_mail.php" method="post">
    <p>
        Sujet du message : <input type="Sujet" name="sujet" /><br>
        <!--Demande du sujet du message-->
        Adresse mail de contact : <input type="Adresse" name="
            adresse" /><br>
        <!--Demande d'une adresse pour pouvoir
            repondre quand meme-->
        Message : </br>
        <!--Demande du contenu de l'article-->
<textarea name="message" rows="8" cols="45">
Tapez votre message ici.</textarea>
        <br>
        <input type="submit" value="Valider" />
    </p>
</form>
</div>

<!--Chargement du pied de page-->
```

```
<?php include("piedDePage.php"); ?>

</body>

</html>
```

## 2.5 envoi\_mail.php

Ce script php permet de récupérer les données du formulaire précédent et envoi tout cela par mail à une adresse défini dans ce script.

```
<?php
$sujet=$_POST['sujet']; //On recupere le sujet
$message=$_POST['message']; //De meme avec le message
$message="\r\n". 'Adresse de contact: '; //On concatene l'adresse ou
    contacter la personne.
$message=$_POST['adresse'];
$message="\r\n";

if (mail('alexandre.vieira@insa-rouen.fr', $sujet, $message)) //Envoi de
    l'e-mail
    header('Location: contact.php?reussite=1'); // Si OK, retour a la
        page de contact, affiche un message pour dire que c'est bon
else
    header('Location: contact.php?reussite=0'); // De meme, message
        signalant echec
?>
```

## Conclusion

Ce projet nous a permis d'aborder plusieurs points :

- l'apprentissage d'un nouveau langage de programmation, qui rappelle par certains aspects les langages de script et le C
- de voir d'une autre manière la programmation par objet et par la même occasion approcher les bases de données

Ce projet nous sera sans aucun doute utile dans l'avenir, pour différents petits projets personnels ou même dans le cadre de nos études.

## A Bibliographie

HEURTEL Olivier, *PHP5 - Développer un site Web dynamique et interactif*, Collection Ressources Informatiques, ENI Editions, 2004

NEBRA Mathieu, *Concevez votre site web avec PHP et MySQL*, 2013, disponible [ici](#)

## B Crédit d'illustration

Page 6 : schéma trouvé [ici](#)

## C Annexes

### C.1 Feuilles CSS

```
/*-----Configuration de la page -----*/
body
{
    width: 700px;
    margin: auto;

    /* Pour
        centrer notre page */
    margin-top: 20px;

    /* Pour ne pas
        coller en haut*/
    margin-bottom: 20px;

    /* Pour ne
        pas coller en bas*/
    background-image: url("images/beige4.jpg");
}

/*-----Configuration du corps de la page ----- */
/*Des blocs */
#news
{
    margin-bottom: 20px;
    background-color: #dece97;
    width : 700px;
    height: 500px;
    overflow: auto;
    text-align: justify;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
}
#addnews
{
    margin-bottom: 20px;
    background-color: #dece97;
    width : 700px;
    height: 110px;
    overflow: hidden;
    text-align: center;
    border:solid;
    border-radius:5px;
```



```

        box-shadow:1px 1px 3px gray;
    }
#formnews
{
    margin-bottom: 20px;
    background-color: #dece97;
    width : 700px;
    height: 250px;
    overflow: hidden;
    text-align: center;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
}
/*Style des paragraphes */
#news p
{
    color: #202020;
    text-indent: 30px;
}
/*Style des titres h1 */
#news h1
{
    color: #112e4b;
    text-align: center;
    font-variant: small-caps;
    margin-bottom: 3px;
}
/*Style des titres h2 */
#news h2
{
    color: #a97633;
    text-align: center;
    margin-bottom: 1px;
    text-decoration: none;
    font-size:10px;
}
/*Style des titres h3 */
#news h3
{
    color: #202070;
    text-align: center;
    text-decoration: none;
}
/*Style d'une news */
.une_news
{
    width:600px;
    background-color: #dece97;
    margin-bottom: 40px;
    margin-top: 40px;
    font-size: 10px;
    margin: auto;
}

```

```

/*----- Configuration de l'en-tete
----- */
#en-tete
{
    height:150px;
/*
    Hauteur en-tete */
    width:700px;
/*
    Largeur en-tete */
    background-image: url("images/banniere6.jpg"); /* La
    banniere */
    margin:3em auto;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
    text-align:center;
    overflow:hidden;
    white-space:nowrap;
    margin-bottom:10px;
}

/*----- Configuration du menu
----- */
/*le bloc*/
#menu
{
    width: 700px;
    height: 50px;
    margin-bottom:30px;
}
/*element menu*/
.element_menu
{
    width: 330px;
    background-color: #112e4b;
    margin-bottom: 20px;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
    font-size: 13px;
}
/*Style des titres dans le menu */
.element_menu h3 , .element_menu h3 a
{
    color: #f3a600 ;
    text-align: center;
    font-variant: small-caps;
}

```

```

}
/*Style des liens dans le menu au passage de la souris */
.element_menu h3 a:hover
{
    background-color: #ffffff;
    color: #f3a600;
    font-weight: bold;
}

/*-----Configuration pied de page-----*/
/*le bloc*/
#pied-de-page
{
    height:30px;
    width:700px;
    margin:3em auto;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
    text-align:center;
    position:relative;
    white-space:nowrap;
    overflow:hidden;
    background-color:#112e4b;
}
/*les paragraphes*/
.piedDePage
{
    font-size: 10px;
    font-variant: small-caps;
}

```

```

/*-----Configuration de la page -----*/
body
{
    width: 700px;
    margin: auto;

    margin-top: 20px;
    margin-bottom: 20px;

    /* Pour centrer notre page */
    /* Pour ne pas coller en haut*/
    /* Pour ne pas coller en bas
    */
    background-image: url("images/beige4.jpg");
}

```

```

/*-----Configuration du corps de la page ----- */
/*Du bloc */
#corps
{
    margin-bottom: 20px;
    background-color: #dece97;
    width : 700px;
    height: 290px;
    overflow: auto;
    text-align: center;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
}
/*Style des paragraphes */
#corps
{
    color: #202020;
    text-indent: 30px;
}
/*Style des titres h1 */
#corps h1
{
    color: #640c00;
    text-align: center;
    margin-bottom: 10px;
}
/*Style des titres h2 */
#corps h2
{
    color: #a97633;
    text-align: center;
    font-variant: small-caps;
    margin-bottom: 6px;
    text-decoration: none;
}
/*Style des titres h3 */
#corps h3
{
    color: #202070;
    text-align: center;
    text-decoration: none;
}

/*----- Configuration de l'en-tete ----- */
#en-tete
{

```

```

        height:150px;
                                                    /*
            Hauteur en-tete */
width:700px;
                                                    /*

        Largeur en-tete */
background-image: url("images/banniere6.jpg");    /* La
        banniere */
margin:3em auto;
border:solid;
border-radius:5px;
box-shadow:1px 1px 3px gray;
text-align:center;
white-space:nowrap;
overflow:hidden;
margin-bottom:10px;
}

/*----- Configuration du menu
        -----*/
/*le bloc*/
#menu
{
        width: 700px;
        height: 50px;
        margin-bottom:30px;
}
/*element menu*/
.element_menu
{
        width: 330px;
        background-color: #112e4b;
        margin-bottom: 20px;
        border:solid;
        border-radius:5px;
        box-shadow:1px 1px 3px gray;
        font-size: 13px;
}
/*Style des titres dans le menu */
.element_menu h3 , .element_menu h3 a
{
        color: #f3a600 ;
        text-align: center;
        font-variant: small-caps;
}
/*Style des liens dans le menu au passage de la souris */
.element_menu h3 a:hover
{
        background-color: #ffffff;
        color: #f3a600;
        font-weight: bold;
}

```

```

}

/*----- Configuration pied de page -----*/
/*le bloc*/
#pied-de-page
{
    height:30px;
    width:700px;
    margin:3em auto;
    border:solid;
    border-radius:5px;
    box-shadow:1px 1px 3px gray;
    text-align:center;
    position:relative;
    white-space:nowrap;
    overflow:hidden;
    background-color:#112e4b;
}
/*les paragraphes*/
.piedDePage
{
    font-size: 10px;
    font-variant: small-caps;
}

```

## C.2 entete.php et piedDePage.php

```

<div id="en-tete">

</div>

```

```

<div id="menu">

    <div style="float:left" class="element_menu">
        <h3><a href="index.php">News</a></h3>
    </div>

    <div style="float:right" class="element_menu">
        <h3><a href="contact.php">Contact</a></h3>
    </div>

</div>

```

```

<div id="pied-de-page">

<p class="piedDePage" >
    Essais Php – Conrad Hillairet et Alexandre Vieira – GM3
</p>

```

|</div>