Réaliser un site vitrine (2)

Dans le cours précédent, nous avons pensé *page* et vu les langages de présentation avec le HTML et le CSS ainsi qu'un peu de graphisme avec Gimp et Inkscape ; dans cette deuxième partie, nous allons penser *site* et voir aussi les langages de programmation PHP et Javascript.
Nous avons fait une page-type que nous avons mise de côté.

Introduction

La page comprend un entête, un corps et un pied. Certaines parties sont invariables et sont faites en HTML. Certaines parties varient et sont faites en PHP ou en javascript.

Le site comprend une navigation différente selon la page, un formulaire de contact et une connexion à la base de données, ce sont les parties dynamiques et interactives.

Le **PHP** est un langage interprété à la volée par le serveur avant d'être présenté au navigateur, ce qui permet d'insérer des données dynamiques et personnalisées provenant d'une base de données. C'est un langage de script, c'est-à-dire que son code est mélangé au code HTML dans la page.

Le **Javascript** est un langage interprété à la volée par le navigateur dans le cas qui nous occupe et peut déceler des erreurs dans les données insérées dans un formulaire avant que l'internaute ne les soumette au serveur. *Il existe aussi du javascript côté serveur depuis la création de Nodejs mais nous n'en parlerons pas ici.*

Les parties communes

Les fichiers communs sont regroupés dans un répertoire nommé *commun* afin de pouvoir facilement s'orienter dans les fichiers.

```php
<?php
include_once ($_SERVER['DOCUMENT_ROOT']."/commun/conf.php");
```

Dans ce document, le fichier commun/conf.php est appelé et inclus. Il contient toutes les données de connexion à la base de données. Ainsi il est possible désormais de lancer des requêtes d'insertion, de mise à jour ou d'effacement de données.

Le répertoire *commun* contient un fichier *html* nommé *index.html* ou *index.php* et qui est vide. Il est présent pour empêcher les internautes de naviguer sur ces fichiers qui contiennent parfois des informations sensibles.
Les autres fichiers communs sont le fichier pour se connecter à la base de données et qui contient tous les paramètres de connexion, le fichier *menu.php* qui présente un menu interactif qui change de couleur selon la page courante, le fichier qui affiche l'entête de page et celui qui affiche le pied de page.

La navigation du site

La navigation du site se fait au moyen de liens vers les différentes pages du site. Cette navigation pourrait se faire avec du HTML statique mais ici nous choisirons de la faire avec des éléments

dynamiques. Le PHP nous aidera à reconnaître la page courante et avec la feuille de style, nous modifierons l'aspect du lien conduisant à cette page pour, ainsi, mieux l'identifier.

```
<?php
echo '<nav role="navigation"><a name="menu"></a><ul class="nav nav-pills">';
if (preg_match("/index\.php/i", $_SERVER['REQUEST_URI']) OR $_SERVER['REQUEST_URI']=="http://exemple.fr/")
{
echo '<li role="presentation" class="active"><a href="#">Accueil</a></li> ';
}
else {
echo '<li role="presentation" class="non-active"><a href="index.php">Accueil</a></li> ';
}
```

Ici le PHP affiche d'abord du HTML avec la commande *echo*, puis l'adresse de la page courante est analysée et selon le résultat, on affiche une classe *non-active* et un dièse en guise de lien, ou alors on affiche une classe CSS *active* et un lien . Dans le premier cas, le nom du lien de navigation est celui de la page courante et dans l'autre cas c'est celui d'une autre page dont nous conservons le lien pour pouvoir y accéder en cliquant dessus. Ce procédé permet aussi de savoir sur quelle page nous sommes.

L'inconvénient de cette méthode est qu'une personne aveugle ne sera pas avertie sur quelle page, elle se trouve. Pour y remédier, il suffit d'ajouter un attribut aria-describedby qui sera lu par les navigateurs audio :

```
echo '<li role="presentation" class="active" aria-describedby="cette page courante est la page d'accueil" ><a
href="#">Accueil</a></li> ';
```

Le formulaire de contact

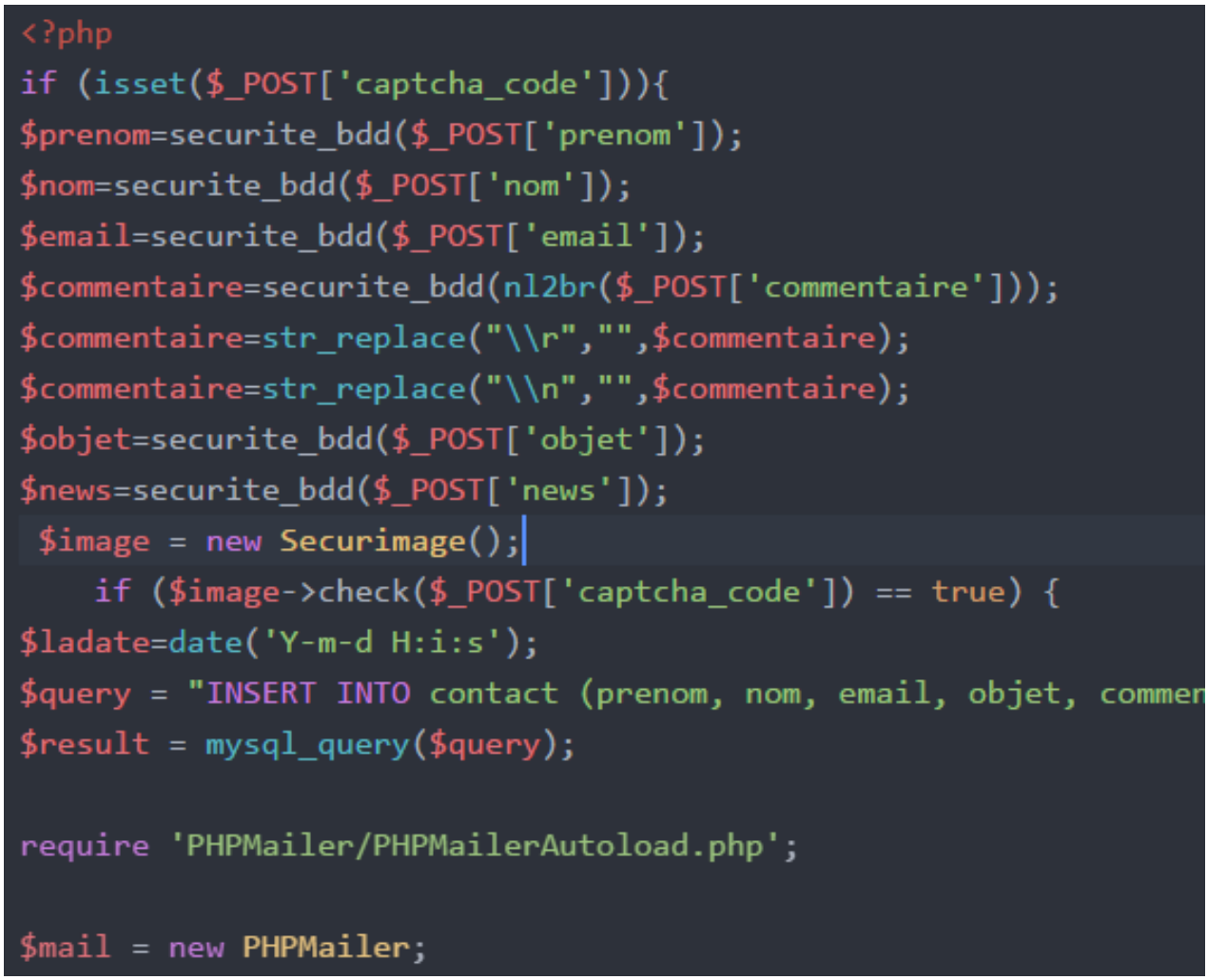
Pour créer un formulaire de contact, on utilise les balises <form> </form>, entre lesquelles on insère des champs comme prénom, nom. Avant la fin du formulaire, un bouton de soumission est ajouté.

Toutes les données du formulaire seront envoyées à la page courante, *contact.php*, pour y être traitées.

```
  <form class="form-horizontal" name="commentform" action="contact.php" method="post" onsubmit="return
verif_formulaire()">
   <div class="form-group">
      <div class="col-md-4">
<label for="prenom">Prénom</label>
         <input type="text" class="form-control" id="first_name" name="prenom" placeholder="Prénom"/>
      </div>
      <div class="col-md-4">
<label for="last_name">Nom de famille</label>
         <input type="text" class="form-control" id="last_name" name="nom" placeholder="Nom de famille"/>
      </div>
         <button type="submit" value="Submit" class="btn btn-warning pull-right">Envoyer</button>
      </div>
   </div>
</form>
```

Pour traiter le fichier, le code est écrit dans le même fichier, contact.php, on décèle si le formulaire a été rempli en testant la valeur du champ *captcha_code* :

```
if (isset($_POST['captcha_code'])){
$prenom=securite_bdd($_POST['prenom']);
$nom=securite_bdd($_POST['nom']);
$email=securite_bdd($_POST['email']);
$commentaire=securite_bdd(nl2br($_POST['commentaire']));
$commentaire=str_replace("\\r","",$commentaire);
$commentaire=str_replace("\\n","",$commentaire);
$objet=securite_bdd($_POST['objet']);
$news=securite_bdd($_POST['news']);
 $image = new Securimage();
   if ($image->check($_POST['captcha_code']) == true) {
$ladate=date('Y-m-d H:i:s');
$query = "INSERT INTO contact (prenom, nom, email, objet, commentaire, news, moment) VALUES ('$prenom', '$nom', '$email', '$objet', '$commentaire', '$news', '$ladate')";
$result = mysql_query($query);
require 'PHPMailer/PHPMailerAutoload.php';
$mail = new PHPMailer;
//ici le programme après avoir testé les valeurs et entré les données dans la base, envoie le contenu dans un email
}
```



Le captcha est l'élément visuel dans un formulaire (accompagné souvent d'une version audio) qui permet de savoir que l'utilisateur est un humain et non une machine. La plupart du temps l'image représente un texte qu'il faut reproduire. Le programme envoie les données dans une fonction qui teste la valeur pour être sûr qu'elle corresponde à ce qu'on attend.

```
if (isset($_POST['captcha_code'])){
```

Il peut alors être intéressant d'insérer un code javascript pour afficher un panneau qui annonce que le formulaire a bien été reçu et que nous y répondrons le plus rapidement possible.

```
echo  "alert( \"Merci de votre message, nous y répondrons le plus rapidement possible\") ;  " ;
```

Les antislashes (\) sont là pour que le programme prennent les guillemets littéralement et non pour la fermeture de la commande *echo.*

Conclusion

Les pages web sont envoyées au navigateur en langage HTML, CSS et Javascript mais, en amont, un langage de script de type PHP, peut en transformer le contenu.

Pour concevoir un site, le développeur commence par créer une page type, un patron, un gabarit, il identifie dedans les parties communes à toutes les pages et qui seront appelées dans chaque page par une ligne de code. Ainsi les modifications seront faites sur un seul fichier. Ensuite il suffit de dupliquer ce qui reste de la page-type et d'y ajouter chaque contenu.