

Tables des matières

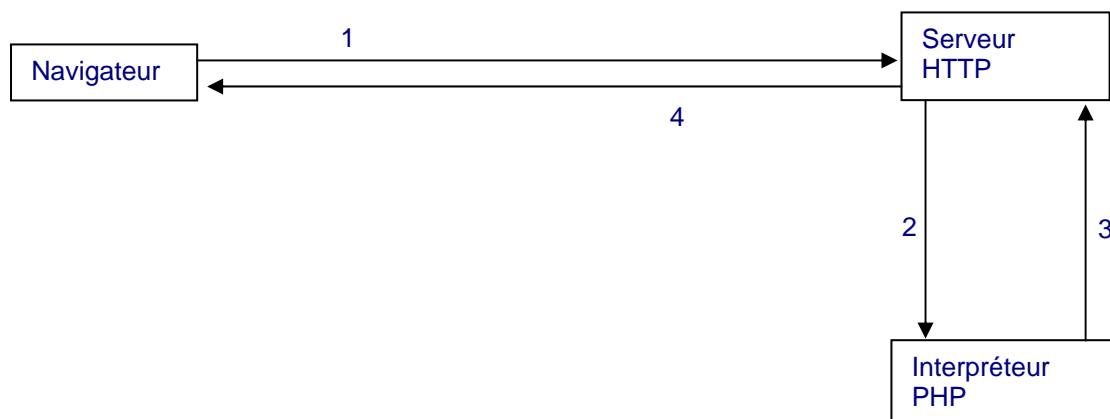
1. Introduction.....	2
2. Environnement de travail.....	2
3. Premiers scripts PHP	3
a. Le tout premier.....	3
b. Une page dynamique.....	3
4. Formulaires et interactivité	5
a. Principe	5
b. Contrôle de type texte.....	5
c. Cases à cocher	6
d. Boutons radios et listes déroulantes.....	8
e. Une rangée de boutons	11
f. Get et Post.....	12
5. Accéder à une base de données.....	13
a. Introduction	13
b. Affichage des données de la base.....	14
c. Mise en forme des données.....	15
d. Sélection de données	17
e. Mise à jour des données.....	20
6 Variables de session	23
a. Introduction	23
b. Utilisation de variables.....	23
c. Divers	24

1. Introduction

Le langage HTML permet la description de pages Web statiques, c'est-à-dire dont le contenu est figé. Ce contenu est fixé lors de la création de la page. Toute modification nécessite une intervention avec un éditeur de texte (ou un logiciel de conception de pages HTML).

La programmation PHP (Personal Home Page tools) est une technique permettant de générer une page HTML à l'aide d'un script. Le contenu de la page est fixé par le script et peut donc être différent d'une exécution à l'autre. A titre d'exemple, une page HTML générée par un script PHP pourrait afficher le contenu d'une table relationnelle et présenter des informations constamment à jour.

Principe de fonctionnement :



1. Le navigateur indique l'URL `http://www.monSite.monScript.php`. Ce lien désigne le script php sur le serveur (il s'agit en fait d'un fichier texte mêlant HTML et code PHP).

2. Le serveur HTTP (Apache par exemple) transmet le fichier à un interpréteur PHP (module intégré ou programme indépendant).

3. L'interpréteur PHP construit une page HTML à la volée. Pour la construire, il peut notamment faire appel à une base de données. Cette page est retournée au serveur HTTP.

4. Le serveur HTTP transmet la page construite précédemment au navigateur. Celui-ci reçoit donc du texte HTML comme d'habitude. Il l'interprète et affiche le résultat.

2. Environnement de travail

L'ensemble des scripts présentés ici ont été testés sous Windows XP avec le serveur Apache et Internet Explorer 6. N'importe quel serveur HTTP supportant PHP peut être utilisé.

Le SGBD utilisé est MySQL. Il est donc possible d'utiliser un autre SGBDR. Il existe notamment des possibilités d'accès à une base de données via ODBC.

La version de PHP utilisée est la version 4.3.3. L'installation de Apache, PHP et MySQL peut être réalisée très simplement à l'aide de l'outil "EasyPhp" que l'on trouve aisément sur Internet.

On suppose par la suite que le répertoire racine du serveur HTTP est `c:\wwwroot` et que le sous répertoire `c:\wwwroot\php` est destiné à contenir les scripts.

On suppose également que le navigateur, le serveur HTTP et le SGBDR sont placés sur le même poste.

3. Premiers scripts PHP

a. Le tout premier

Saisissez le script ci-dessous (php01.php) :

```
<html><head><title>Premier script php</title></head>
<body>
<?php
    echo "Cette page est générée par un script php";
?>
</body>
</html>
```

Remarques

- Le code PHP est encadré par les balises <?php et ?>. On peut également utiliser <?phpphp et ?>.
- L'instruction echo génère simplement la chaîne de caractères qui la suit sur la page finale.

Test du script

Ce texte doit être enregistré sous c:\wwwroot\php\php01.php. Lancer un navigateur et pointer sur <http://localhost/php/php01.php>. Afficher le source de la page pour voir...

b. Une page dynamique

Le second script produit une page dynamique. Le contenu de la page HTML construite dépend du poste client (navigateur utilisé et adresse IP) et de la date de consultation.

Il illustre notamment l'utilisation des variables d'environnement CGI. Ces variables sont mises à disposition du script par le serveur HTTP. Elles concernent le serveur (nom, version du logiciel, ...) ou le client (navigateur, adresse IP, ...).

// php02.php

```
<html>
<head><title>Second script php</title></head>
<body>
<?php
    // ceci est un commentaire
    $jour=date("d");
    $mois=date("m");
    $an=date("Y");
    echo "Bonjour, ici Big Brother...<br /><br />";
    echo "Nous sommes le ".$jour."/ ".$mois."/ ".$an."<br />";
    echo "Votre adresse IP est '$_SERVER["REMOTE_ADDR"]."<br />";
    echo "Vous utilisez le navigateur ".$_SERVER["HTTP_USER_AGENT"]."<br />";
?>
</body>
</html>
```

Il est clair que ce script ne produit pas la même page html pour tout le monde !

Remarques :

- Le nom d'une variable locale commence toujours par \$. Les variables ne se déclarent pas.
- Les variables d'environnement sont regroupées dans le « tableau associatif » \$_SERVER.
- Une chaîne de caractères peut être indifféremment encadrée par des guillemets ou des simples quotes.

SIO - Introduction à la programmation PHP (version PDO)

- Le point (.) est l'opérateur de concaténation de chaînes.
- Il est même possible de définir des fonctions (function).
- On peut également inclure un fichier contenant des fonctions PHP (include).

4. Formulaires et interactivité

a. Principe

Une première page HTML (statique) est envoyée au client. Cette page contient un formulaire, c'est à dire un ensemble de champs pouvant être saisis par l'utilisateur.

En cliquant sur un bouton de validation, l'utilisateur provoque l'envoi des données saisies au serveur HTTP qui les transmet à un script PHP. Le script construit alors une seconde page HTML (dynamique) dont le contenu sera fonction de la saisie de l'utilisateur.

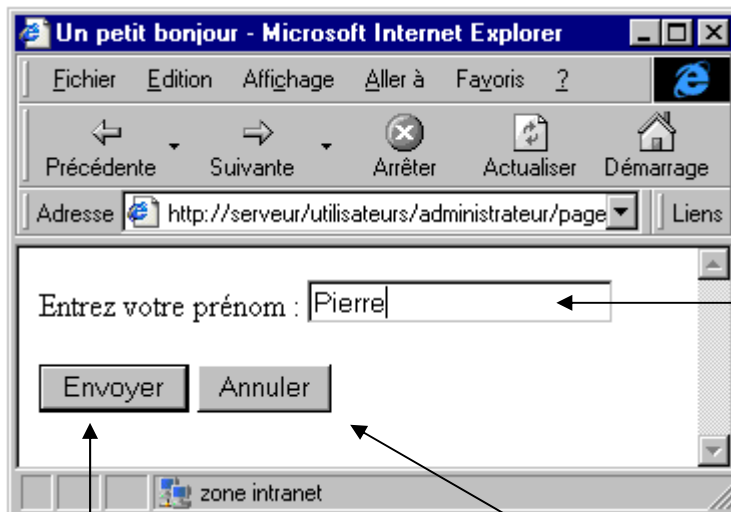
b. Contrôle de type texte

L'exemple ci dessous illustre l'utilisation d'un champ de saisie.

Page HTML de départ (page01_03.html) :

```
<html>
<head><title> Un petit bonjour </title></head>
<body>
<form action="../Scripts /php03.php" method="post">
Entrez votre prénom :
<input type="text" name="prenom" size="20" />
<p />
<input type="submit" value="Envoyer" />
<input type="reset" value="Annuler" />
</form>
</body>
</html>
```

Ce code HTML produit la page suivante :



L'utilisateur peut saisir dans la zone de texte. Cette saisie sera récupérée dans le programme php03 (variable \$_POST["prenom"]).

Un clic sur ce bouton lance le programme php03 et lui envoie l'information saisie.

Un clic sur ce bouton annule la saisie et remet à blanc la zone de texte. Cette réinitialisation est gérée par le navigateur, il n'y a rien à faire !

SIO - Introduction à la programmation PHP (version PDO)

Le script ci-dessous traite l'information saisie et construit la seconde page HTML.

// php03.php

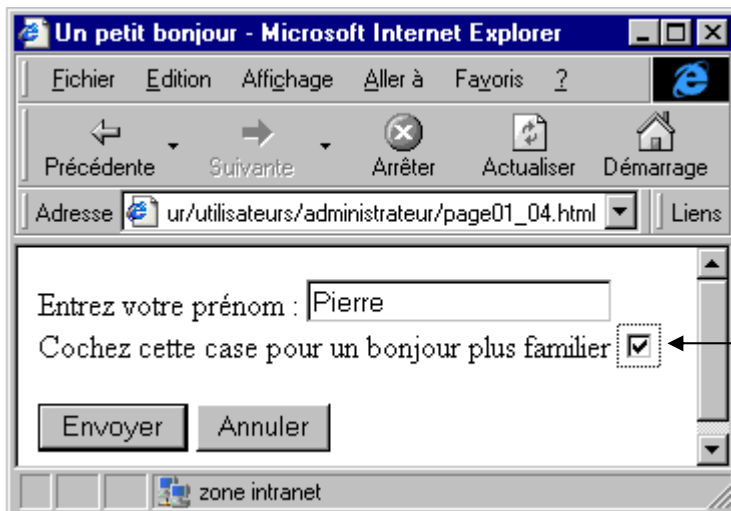
```
<html>
<head>
<title>Un petit bonjour</title>
</head>
<body>
<?php
    echo "Bonjour ".$_POST["prenom"];
?>
</body>
</html>
```

Remarques :

- En remplaçant le type "text" par le type "password" pour la zone de saisie dans page01_03.html, le prénom saisi par l'utilisateur n'apparaît pas lors de la frappe. Le script php continue à fonctionner de la même manière. Le type "textarea" permet la création d'un champ multilignes. Le type "hidden" permet la création d'un champ caché, ce qui permet par exemple d'envoyer une donnée à un script PHP.
- Le tableau \$_POST permet de récupérer les valeurs transmises par la méthode POST, le tableau \$_GET est l'équivalent pour la méthode GET.
- Il est possible de préciser une cible (frame ou fenêtre) différente à l'aide de l'attribut target.

c. Cases à cocher

La page HTML suivante (page01_04.html) offre à l'utilisateur la possibilité de choisir un bonjour plus "familier".



Dans le script appelé (php04.php), la fonction `empty($_POST["familier"])` retournera faux si la case n'est pas cochée.

SIO - Introduction à la programmation PHP (version PDO)

```
<html>
<head><title> Un petit bonjour </title></head>
<body>
<form action="../Scripts/php04.php" method="post">
Entrez votre prénom :
<input type="text" name="prenom" size="20" />
<br />
Cochez cette case pour un bonjour plus familier
<input type="checkbox" name="familier">
<p />
<input type="submit" value="Envoyer" />
<input type="reset" value="Annuler" />
</form>
</body>
</html>
```

Le script ci-dessous gère cette nouvelle possibilité.

// php04.php

```
<html>
<head>
<title>Un petit bonjour</title>
</head>
<body>
<?php
    if (empty($_POST["familier"]))
    {
        echo "Bonjour ".$_POST["prenom"];
    }
    else
    {
        echo "Salut ".$_POST["prenom"];
    }
?>
</body>
</html>
```

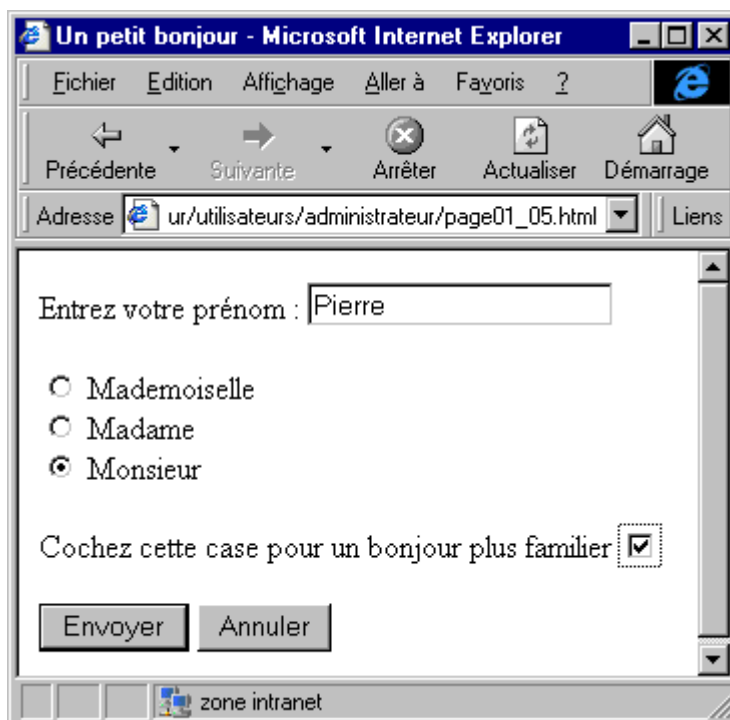
La syntaxe du langage est très proche de celle du langage C et ne sera donc pas décrite ici dans le détail.

d. Boutons radios et listes déroulantes

Ajoutons cette fois une formule de politesse !

Page01_05.html

```
<html>
<head><title> Un petit bonjour </title></head>
<body>
<form action="../../Scripts/php05.php" method="post">
Entrez votre prénom :
<input type="text" name="prenom" size="20" />
<p />
<input type="radio" name="politesse" value="1" checked /> Mademoiselle<br />
<input type="radio" name="politesse" value="2" /> Madame<br />
<input type="radio" name="politesse" value="3" /> Monsieur<br />
<p />
Cochez cette case pour un bonjour plus familier
<input type="checkbox" name="familier" />
<p />
<input type="submit" value="Envoyer" />
<input type="reset" value="Annuler" />
</form>
</body>
</html>
```



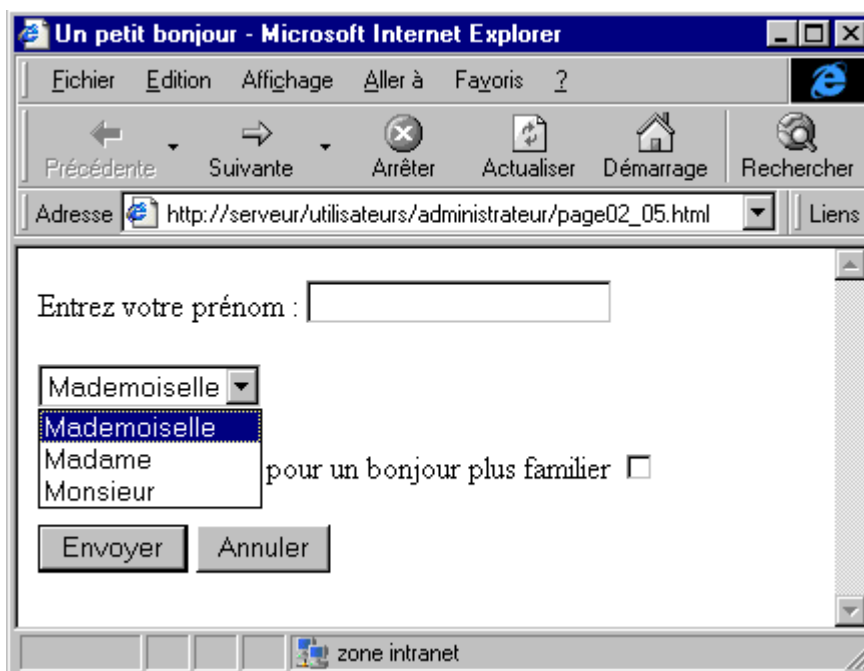
SIO - Introduction à la programmation PHP (version PDO)

Le formulaire est traité par le script contenu dans php05.php :

```
<html>
<head>
<title>Un petit bonjour</title>
</head>
<body>
<?php
    $message="";
    if (!empty($_POST["familier"]))
    {
        $message = "Salut ";
    }
    else
    {
        $message = "Bonjour ";
    }
    switch($_POST["politesse"])
    {
        case 1 : $message = $message."Mademoiselle ";
                break;
        case 2 : $message = $message."Madame ";
                break;
        case 3 : $message = $message."Monsieur ";
                break;
    }
    $message = $message.$_POST["prenom"];
    echo $message;
?>
</body>
</html>
```

Remarque : la valeur renvoyée est obligatoirement 1, 2 ou 3.

Le formulaire ci-dessous (page_02_05.html)...



The screenshot shows a Microsoft Internet Explorer window titled "Un petit bonjour - Microsoft Internet Explorer". The address bar displays "http://serveur/utilisateurs/administrateur/page02_05.html". The form contains a text input field labeled "Entrez votre prénom :". Below it is a dropdown menu with "Mademoiselle" selected. To the right of the dropdown is a checkbox labeled "pour un bonjour plus familier". At the bottom are two buttons: "Envoyer" and "Annuler". The status bar at the bottom indicates "zone intranet".

... obtenu par le source HTML suivant ...

SIO - Introduction à la programmation PHP (version PDO)

```
<html>
<head><title> Un petit bonjour </title></head>
<body>
<form action="../../Scripts/php05.php" method="post">
Entrez votre prénom :
<input type="text" name="prenom" size="20" />
<p />
<select name="politesse" size="1">
<option selected value=1> Mademoiselle</option>
<option value=2> Madame</option>
<option value=3> Monsieur</option>
</select>
<p />
Cochez cette case pour un bonjour plus familier
<input type="checkbox" name="familier" />
<p />
<input type="submit" value="Envoyer" />
<input type="reset" value="Annuler" />
</form>
</body>
</html>
```

... peut être géré par le même script PHP (php05.php).

En indiquant : `<select name="politesse" size="3">`, on obtient une interface de type "liste déroulante" en lieu et place du "menu déroulant" pour le choix de la politesse. La gestion du formulaire reste cependant identique. Si le nombre d'éléments est supérieur à la taille indiquée par "size", la liste est munie d'un ascenseur vertical (page03_05.html).

En indiquant : `<select name="politesse" size="3" multiple>`, on obtient une liste déroulante dans laquelle il est possible de choisir plusieurs valeurs !

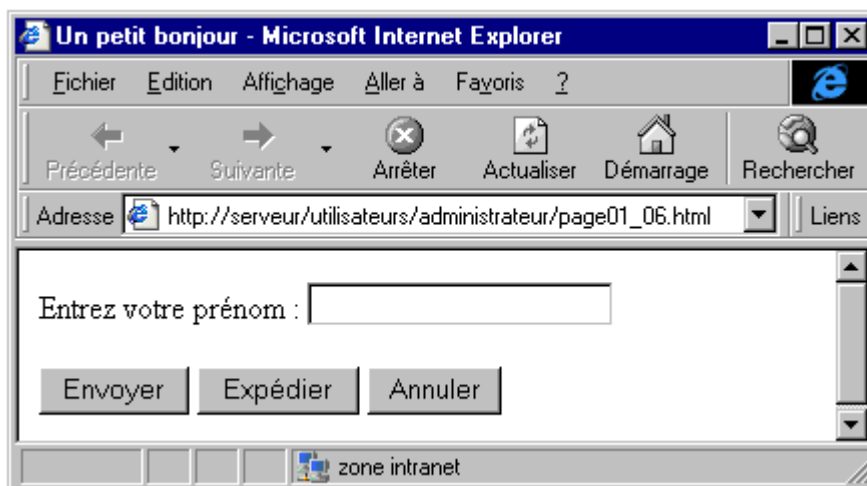
e. Une rangée de boutons

Il est possible de faire figurer plusieurs boutons "submit" sur le même formulaire. Il est alors nécessaire de leur associer une variable qui sera utilisée dans le script pour déterminer lequel des boutons a été cliqué.

Le code suivant (page01_06.html) ...

```
<html>
<head><title> Un petit bonjour </title></head>
<body>
<form action="../Scripts/php06.php" method="post">
Entrez votre prénom :
<input type="text" name="prenom" size="20" />
<p />
<input type="submit" name="validation" value="Envoyer" />
<input type="submit" name="validation" value="Expédier" />
<input type="reset" value="Annuler" />
</form>
</body>
</html>
```

... produit la page ci-dessous ...



... dont le formulaire peut être géré par le script php suivant :

// php06.php

```
<html>
<head>
<title>Un petit bonjour</title>
</head>
<body>
<?php
    echo "Bonjour ".$_POST["prenom"];
    echo "<br />";
    echo "Vous avez cliqué sur le bouton ".$_POST["validation"];
?>
</body>
</html>
```

Remarque :

Un bouton de formulaire peut être de type "submit" (envoyer le formulaire), "reset" (annuler la saisie), ou simplement "button" (faisant appel à un script javascript par exemple).

f. Get et Post

Il existe plusieurs méthodes pour envoyer des données à un script PHP à l'aide d'un formulaire.

Nous avons jusqu'ici utilisé la méthode "post". Celle-ci envoie littéralement le formulaire au serveur HTTP. Celui-ci extrait alors les données transmises pour les exploiter par le biais de la variable PHP `$_POST`.

La méthode "get", quant à elle, rassemble les données à transmettre dans la chaîne de requête, c'est-à-dire l'URL demandée.

Modifiez la page "page01_03.html" pour utiliser la méthode "get" au lieu de "post", modifiez le script "php03.php" pour utiliser `$_GET` au lieu de `$_POST` et observez l'URL affichée par "php03.php".

Celle-ci est de la forme : `http://localhost/php/php03.php?prenom=Pierre`. La méthode "get" transmet bien les données dans la chaîne de requête.

> Conséquences intéressantes

- Il est possible d'entrer directement les données à transmettre dans l'URL du navigateur (essayez `http://localhost/php/php03.php?prenom=Christine`).

- Il est possible de transmettre des valeurs prédéterminées à un script PHP de la manière suivante :

```
<html>
<head>
<title>Bonjour Paul</title>
</head>
<body>
<a href="../Scripts/php03.php?prenom=Paul">Cliquez ici...</a>
</body>
</html>
```

> Transmission de plusieurs valeurs

Lorsqu'il y a plusieurs valeurs à transmettre, elles sont séparées par le caractère & dans la chaîne de requête construite.

Exemple : `http://localhost/php/php03.php?prenom=Christine&nom=Gaubert`.

> Caractères spéciaux

- Il est possible de transmettre des espaces en les remplaçant par le caractère +.

Exemple : `http://localhost/php/php03.php?prenom=Christine&nom= Gaubert+Macon`.

- Tout caractère peut être remplacé par "%code" où code représente son code ascii exprimé en hexadécimal.

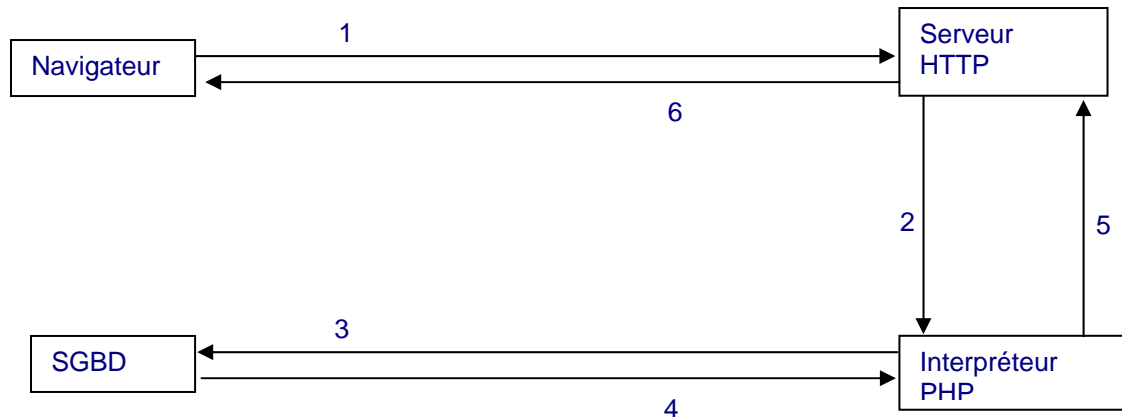
Exemple : <http://localhost/php/php03.php?prenom=Christine&nom=Gaubert%20Macon>.

5. Accéder à une base de données

a. Introduction

La technologie PHP permet de faire interagir une page HTML et une base de données (il y a bien entendu d'autres solutions).

Le schéma de principe de l'application devient donc le suivant (nous sommes en plein dans l'architecture client/serveur !) :



1. Le navigateur indique l'URL `http://www.monSite.monScript.php`. Ce lien désigne le script PHP sur le serveur.

2. Le serveur HTTP (Apache par exemple) l'exécution de l'interpréteur PHP en lui transmettant le script.

3. L'interpréteur PHP s'exécute et envoie une requête SQL au SGBD.

4. Le SGBD exécute la requête reçue et transmet le résultat à l'interpréteur PHP.

5. L'interpréteur PHP reçoit le résultat de la requête et construit une page HTML à la volée. Cette page est retournée au serveur HTTP.

6. Le serveur HTTP transmet la page construite précédemment au navigateur. Celui-ci reçoit donc du texte HTML comme d'habitude. Il l'interprète et affiche le résultat.

Base de données exemple

Nous travaillerons avec une base de données très simple dont voici le schéma :

service (code (char 3), designation (char 30))

employe (matricule (char 4), nom (char 25), prenom (char 20), cadre (char 1), service# (char 3))

Cette base est implantée sous MySql.

Les connexions s'effectuent avec l'utilisateur root sans mot de passe.

b. Affichage des données de la base

Pour afficher des données issues de la base, il faut créer une page HTML dynamique affichant le résultat de l'exécution d'une requête. Le script PHP suivant affiche la liste des employés.

// php07.php

```
<html>
<head>
<title>Liste des employés</title>
</head>
<body>
<?php
// Connection au serveur
try {
    $dns = 'mysql:host=localhost;dbname= empScePhp ';
    $utilisateur = 'root';
    $motDePasse = '';
    $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
} catch ( Exception $e ) {
    echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
    die();
}
// Création de la requête
$requete = $pdo->query("SELECT * FROM employe");

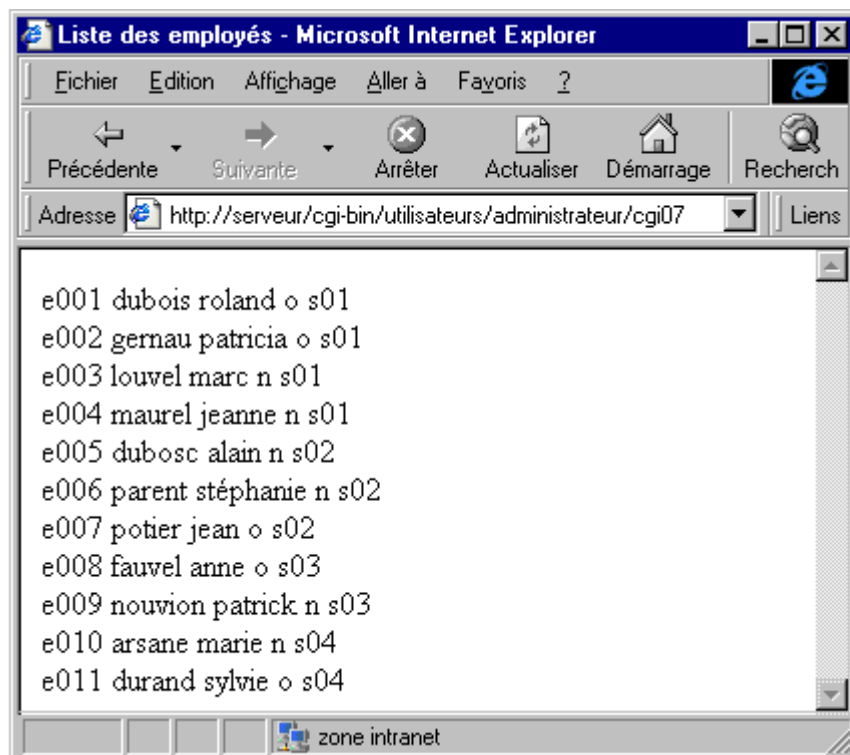
// Traitement du résultat de cette requête
while( $ligne = $requete->fetch() ) {
    {
        echo $ligne["matricule"];echo " ";
        echo $ligne["nom"];echo " ";
        echo $ligne["prenom"];echo " ";
        echo $ligne["cadre"];echo " ";
        echo $ligne["service"];echo "<br />";
    }
unset($pdo);
unset($requete);
?>
</body>
</html>
```

Après la connexion à la base de données, le script construit "à la volée" la page HTML contenant les informations lues dans la table employé.

La fermeture de la connexion se fait à l'aide de la fonction unset(\$pdo).

SIO - Introduction à la programmation PHP (version PDO)

Ce programme produit le résultat suivant :

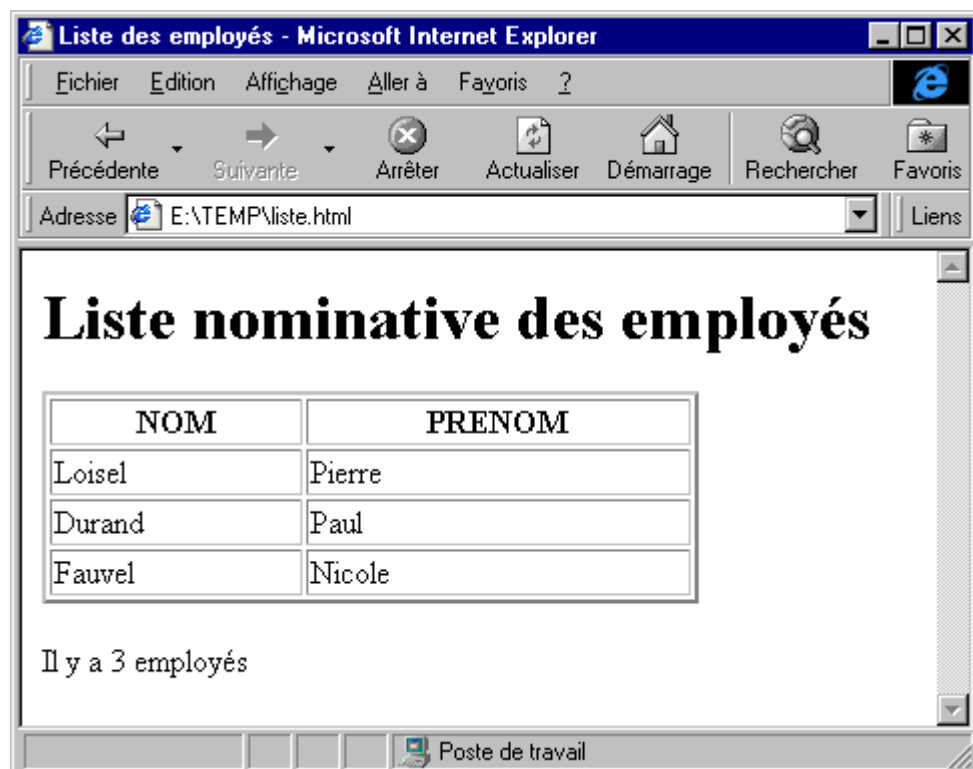


c. Mise en forme des données

Il est bien entendu possible de faire un petit effort de présentation. Le script php08.php affiche la liste des employés (nom et prénom) dans un tableau formaté en HTML.

La méthode la plus simple est sans doute de commencer par écrire une page html affichant le type de résultat souhaité (à l'aide d'un éditeur ou d'un générateur).

Cette page ...



SIO - Introduction à la programmation PHP (version PDO)

... est obtenue par le source HTML ci-dessous :

```
<html>
<head>
<title>Liste des employés</title>
</head>

<body>
<h1>Liste nominative des employés</h1>
<p /><table border=2 width="75%">
  <tr><th>NOM</th><th>PRENOM</th></tr>
  <tr><td>Loisel</td><td>Pierre</td></tr>
  <tr><td>Durand</td><td>Paul</td></tr>
  <tr><td>Fauvel</td><td>Nicole</td></tr>
</table>
<p />
Il y a 3 employés.
</body>
</html>
```

Ce source HTML peut être généré par le script suivant :

// php08.php

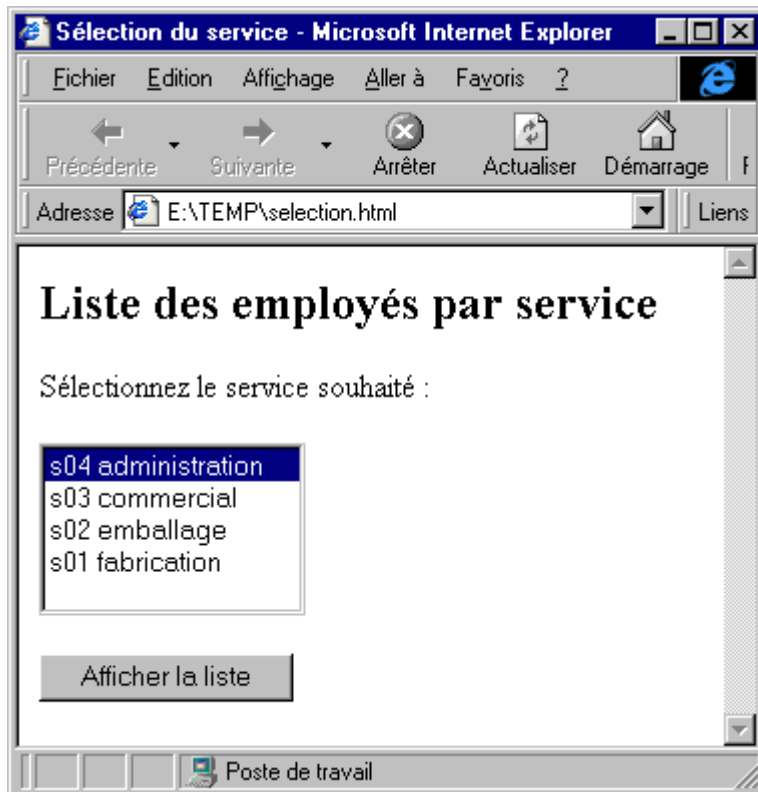
```
<html>
<head>
  <title>Liste des employés</title>
</head>
<body>
<?php
try {
  $dns = 'mysql:host=localhost;dbname= empScePhp ';
  $utilisateur = 'root';
  $motDePasse = '';
  $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
  } catch ( Exception $e ) {
    echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
    die();
  }
  $nb=0; // Initialisation du compteur du nombre de personnes
  echo "<h1>Liste nominative des employés</h1>";
  echo '<p /><table border="2" width="75%">';
  echo "<tr><th>NOM</th><th>PRENOM</th></tr>";
  // Création de la requête
  $requete = $pdo->query("SELECT nom,prenom from employe;"); // Requête SQL

  // Traitement du résultat de cette requête
  while( $ligne = $requete->fetch() ) {
    // On affiche le contenu de la ligne
    echo "<tr><td>".$ligne["nom"]."</td><td>".$ligne["prenom"]."</td></tr>";
    $nb=$nb+1; // On incrémente de compteur de 1
  } // FINTANTQUE
  echo "</table><p />";
  echo "Il y a ".$nb." employés.";
}
unset($pdo);
unset($requete);
?>
</body>
</html>
```


d. Sélection de données

Un premier script affiche la liste des services. L'utilisateur sélectionne le service souhaité et un second script recherche et affiche les employés de ce service.

Interface souhaitée pour la sélection du service :



Le code HTML correspondant à cette page est le suivant :

```
<html>
<head>
<title>Sélection du service</title>
</head>
<body>
<form action="..\Scripts/php10.php" method="post">
<h2>Liste des employés par service</h2>
<p />Sélectionnez le service souhaité :<p />
<select name="service" size="5">
  <option selected value = "s04">s04 administration</option>
  < option value = "s03">s03 commercial</option>
  < option value = "s02">s02 emballage</option>
  < option value = "s01">s01 fabrication</option>
</select>
<p /><input type="submit" value="Afficher la liste"><p />
</form>
</body>
</html>
```

L'attribut value dans les balises option va permettre de faciliter la recherche des employés dans la base de données. Cette recherche se fera dans le script php10.php cité dans l'attribut action du formulaire.

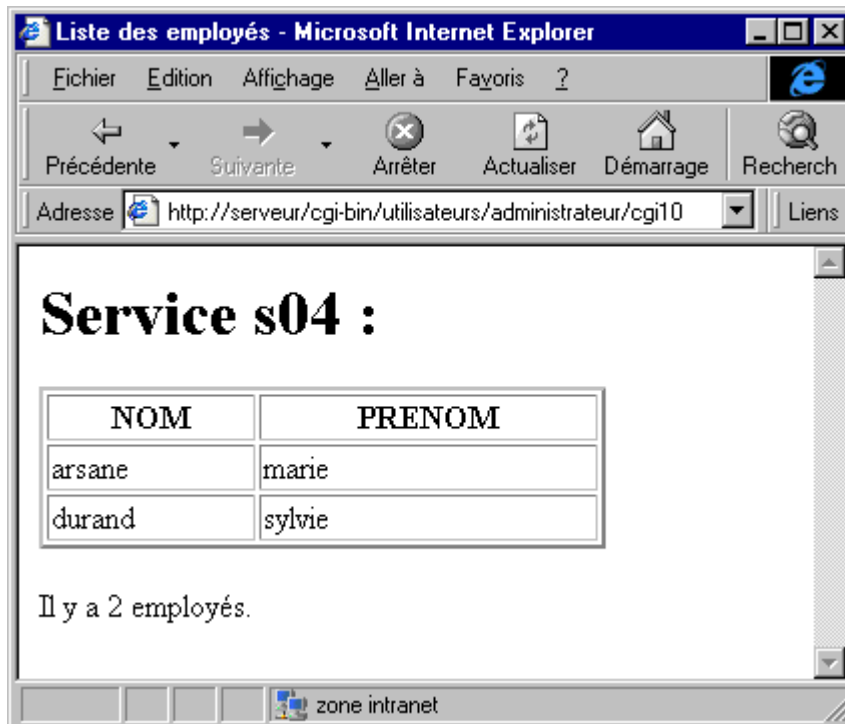
SIO - Introduction à la programmation PHP (version PDO)

Il reste à écrire le script php09.php permettant de le générer à partir de la base de données :

Script php09.php

```
<html>
<head>
<title>Selection du service</title>
</head>
<body>
<?php
try {
    $dns = 'mysql:host=localhost;dbname=empSce';
    $utilisateur = 'root';
    $motDePasse = '';
    $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
    } catch ( Exception $e ) {
        echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
        die();
    }
    echo '<form action="../../Scripts/php10.php" method="post">';
    echo "<h2>Liste des employes par service</h2>";
    echo "<p />Selectionnez le service souhaite :<p />";
    echo '<select name="service" size="5">';
    $requete = $pdo->query("select * from service order by 1;"); // Requête SQL
    // Traitement du résultat de cette requête
    $ligne = $requete->fetch();
    echo '<option selected value = "' . $ligne["code"] . '">' . $ligne["code"]
. ' ' . $ligne["designation"] . '</option>'; // ne pas changer de ligne
    while( $ligne = $requete->fetch() ) {
        echo '<option value = "' . $ligne["code"] . '">' . $ligne["code"]
. ' ' . $ligne["designation"] . '</option>'; // ne pas changer de ligne
    }
    echo "</select>";
    echo '<p /><input type="submit" value="Afficher la liste"><p />';
    echo "</form>";
    unset($pdo);
    unset($requete);
?>
</body>
</html>
```

Le script php10.php, appelé depuis ce formulaire, va rechercher les employés du service sélectionné et les afficher dans un tableau HTML.

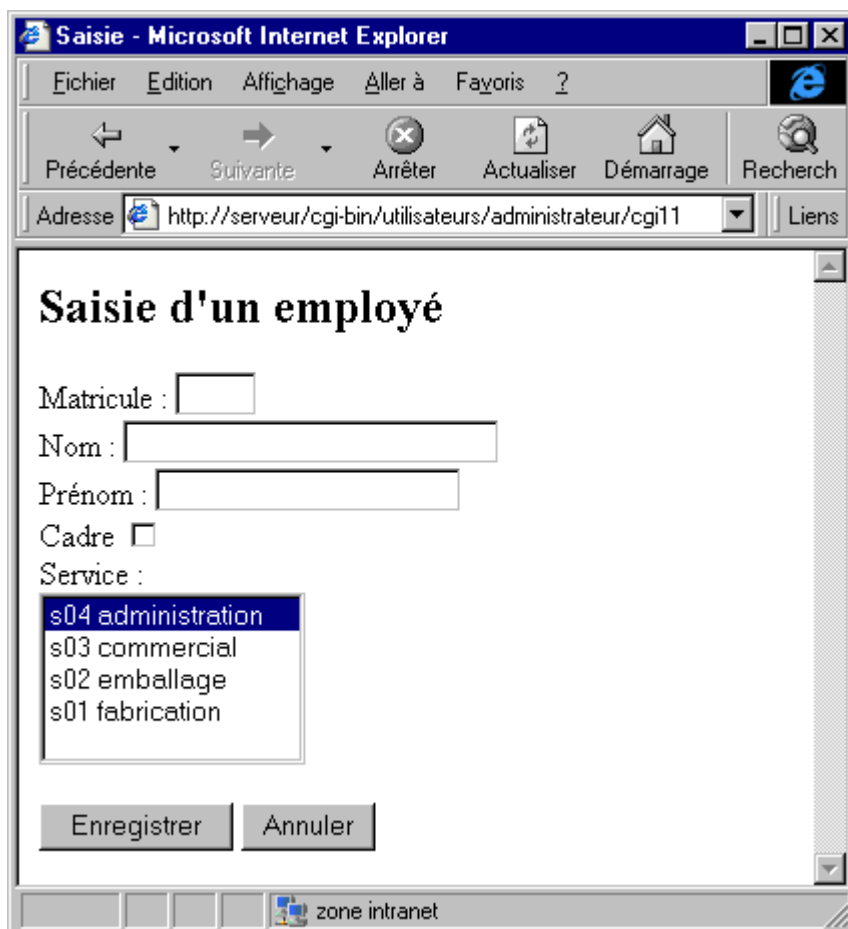


Script php10.php

```
<html>
<head>
<title>Liste des employes</title>
</head>
<body>
<?php
try {
    $dns = 'mysql:host=localhost;dbname=empSce';
    $utilisateur = 'root';
    $motDePasse = '';
    $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
    } catch ( Exception $e ) {
        echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
        die();
    }
    $nb=0;
    echo "<h1>Liste nominative des employes</h1>";
    echo "<p /><table border='2' width='75%'>";
    echo "<tr><th>NOM</th><th>PRENOM</th></tr>";
    $requete = $pdo->query("select nom,prenom from employe where service='".$$_POST["service"]."'");
    // Traitement du résultat de cette requête
    while( $ligne = $requete->fetch() ) {
        echo "<tr><td>".$ligne["nom"]."</td><td>".$ligne["prenom"]."</td></tr>";
        $nb++;
    }
    echo "</table><p />";
    echo "Il y a ".$nb." employes.";
    unset($pdo);
    unset($requete);
?>
</body>
</html>
```

e. Mise à jour des données

Le formulaire ci-dessous permet la saisie des informations concernant un nouvel employé.



The screenshot shows a Microsoft Internet Explorer window titled "Saisie - Microsoft Internet Explorer". The address bar displays "http://serveur/cgi-bin/utilisateurs/administrateur/cgi11". The main content area is titled "Saisie d'un employé" and contains the following form elements:

- Matricule :
- Nom :
- Prénom :
- Cadre ☐
- Service :
 - s04 administration
 - s03 commercial
 - s02 emballage
 - s01 fabrication

At the bottom of the form are two buttons: "Enregistrer" and "Annuler". The status bar at the bottom of the browser window shows "zone intranet".

Lorsque l'utilisateur clique sur le bouton enregistrer, le script php12.php est lancé.

Ce formulaire est généré par le script php11.php présenté page suivante.

SIO - Introduction à la programmation PHP (version PDO)

Script php11.php

```
<html>
<head>
<title>Saisie</title>
</head>
<body>
<?php
try {
    $dns = 'mysql:host=localhost;dbname=empSce';
    $utilisateur = 'root';
    $motDePasse = '';
    $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
    } catch ( Exception $e ) {
        echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
        die();
    }
    echo '<form action="../Scripts/php12.php" method=post>';
    echo "<h2>Saisie d'un employe</h2>";
    echo 'Matricule : <input type="text" name="matricule" size="3" /><br />';
    echo 'Nom : <input type="text" name="nom" size="25" /><br />';
    echo 'Prenom : <input type="text" name="prenom" size="20" /><br />';
    echo 'Cadre <input type="checkbox" name="cadre" /><br />';
    echo 'Service :<br />';
    echo '<select name="service" size="5">';
    $requete = $pdo->query("select * from service order by designation desc;");
    // Traitement du résultat de cette requête
    $ligne = $requete->fetch() ;
    echo '<option selected value = "'. $ligne["code"].'">'. $ligne["code"].' ' . $ligne["designation"];
    echo "</option>";
    while( $ligne = $requete->fetch() ) {
        echo '<option value = "'. $ligne["code"].'">'. $ligne["code"].' ' . $ligne["designation"] . '</option>';
    }
    echo "</select>";
    echo '<p /><input type="submit" value = "Enregistrer" />
        <input type="reset" value="Annuler" /><p />';
    echo '</form>';
    unset($pdo);
    unset($requete);
?>
</body>
</html>
```

SIO - Introduction à la programmation PHP (version PDO)

Le script php12.php, lancé depuis le formulaire précédent, permet l'insertion du nouvel employé dans la base de données. Un message d'erreur est affiché en cas de problème (clé primaire dupliquée par exemple).

Script php12.php

```
<html>
<head>
  <title>Ajout d'un employe</title>
</head>
<body>
<?php
try {
  $dns = 'mysql:host=localhost;dbname=empSce';
  $utilisateur = 'root';
  $motDePasse = '';
  $pdo = new PDO( $dns, $utilisateur, $motDePasse , array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_WARNING));
  } catch ( Exception $e ) {
    echo "Connection à la Base MySQL empSce impossible : ", $e->getMessage();
    die();
  }
  if (empty($_POST["cadre"])) {
    $charCadre='n';
  } else {
    $charCadre='o';
  }
  $matricule=$_POST["matricule"];
  $nom=$_POST["nom"];
  $prenom=$_POST["prenom"];
  $cadre=$_POST["cadre"];
  $service=$_POST["service"];
  $sql = "insert into employe(matricule,nom,prenom,cadre,service) VALUES (:matricule, :nom, :prenom,
:cadre, :service)";
  $res=$pdo->prepare($sql);
  $ok = $res->execute(array(
    ":matricule"=>$matricule,
    ":nom"=>$nom,
    ":prenom"=>$prenom,
    ":cadre"=>$cadre,
    ":service"=>$service
  ));
  if ($ok) {
    echo "L'employe a ete correctement ajoute";
  } else {
    echo "Attention, l'ajout de l'employe a echoue !!!";
  }
  unset($pdo);
  unset($requete);
?>
</body>
</html>
```

Il est bien entendu possible d'utiliser toute la puissance du langage SQL : delete, update, create...

6 Variables de session

a. Introduction

Il est possible de conserver des valeurs d'une page à l'autre en utilisant le mécanisme des sessions. Une session est un ensemble d'informations relatives à un utilisateur connecté sur le site. Ces informations sont enregistrées sur le serveur et existent donc quelque soit la page PHP courante. Toute page souhaitant accéder à ces informations de session doit commencer par :

```
<?php
    session_start();
?>
```

Attention, ces lignes doivent être les premières du fichier.

Chaque session possède un identifiant qui conserve la même valeur au fil des pages PHP. Cette valeur de type chaîne de caractères peut être utilisée de la manière suivante :

```
<?php
    session_start();
?>
<html>
<body>
<?php
    echo session_id();
?>
</body>
</html>
```

b. Utilisation de variables

Lorsqu'une session est créée, il est possible de déclarer une variable de session et de lui donner une valeur. Voici l'exemple d'une page "sessions2.php" qui définit deux variables de session et comporte un lien vers une page "sessions3.php" :

```
<?php
    session_start();
?>
<html>
<body>
<?php
    session_register("nom");
    $_SESSION["nom"]="Dupont";
    session_register("prenom");
    $_SESSION["prenom"]="Paul";
?>
<a href="sessions3.php">sessions3.php</a>
</body>
</html>
```

La page "sessions3.php" peut ensuite utiliser les deux variables définies :

```
<?php
    session_start();
?>
<html>
<body>
<?php
    echo 'Nom : ' . $_SESSION["nom"] . '<br />';
    echo 'Prénom : ' . $_SESSION["prenom"];
?>
</body></html>
```

c. Divers

- Pour savoir si une variable a été définie, on peut utiliser la fonction "isset" (ceci est valable pour toute variable, pas uniquement pour les variables de session) :

```
if (!isset($_SESSION["nom"]))
{
    session_register("nom");
    $_SESSION["nom"]="Dupont";
}
```

Remarque : pour les variables de session, il est également possible d'utiliser :

```
if (session_is_registered("prenom"))
{
    echo "la variable prenom est définie";
}
```

- Pour supprimer une variable de session :

```
session_unregister($_SESSION["nom"]);
```

- Pour supprimer toutes les variables de la session courante :

```
session_unset();
```


SIO - Introduction à la programmation PHP (version PDO)

Script crebase_empsce SQL

```
--
-- Base de données: empsce
--

-----

--
-- Structure de la table employe
--

DROP TABLE IF EXISTS employe;
CREATE TABLE IF NOT EXISTS employe (
  matricule char(4) NOT NULL,
  nom char(25) DEFAULT NULL,
  prenom char(20) DEFAULT NULL,
  cadre char(1) DEFAULT NULL,
  service char(3) DEFAULT NULL,
  PRIMARY KEY (matricule)
);

--
-- Contenu de la table employe
--

INSERT INTO employe (matricule, nom, prenom, cadre, service) VALUES
('e001', 'dubois', 'roland', 'o', 's01'),
('e002', 'gernau', 'patricia', 'o', 's01'),
('e003', 'louvel', 'marc', 'n', 's01'),
('e004', 'maurel', 'jeanne', 'n', 's01'),
('e005', 'dubosc', 'alain', 'n', 's02'),
('e006', 'parent', 'sptéphanie', 'n', 's02'),
('e007', 'potier', 'jean', 'o', 's02'),
('e008', 'fauvel', 'anne', 'o', 's03'),
('e009', 'nouvion', 'patrick', 'n', 's03'),
('e010', 'arsane', 'marie', 'n', 's04');

-----

--
-- Structure de la table service
--

DROP TABLE IF EXISTS service;
CREATE TABLE IF NOT EXISTS service (
  code char(3) NOT NULL,
  designation char(30) DEFAULT NULL,
  PRIMARY KEY (code)
);

--
-- Contenu de la table service
--

INSERT INTO service (code, designation) VALUES
('s01', 'fabrication'),
('s02', 'emballage'),
('s03', 'commercial'),
('s04', 'administration');
```

SIO - Introduction à la programmation PHP (version PDO)

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```