|  |
| --- |
| ***introduction PHP – 10 séances de 4h*** |

(Support proposé par C. Dabancourt, inspiré aussi par des exemples et tutoriels Internet)

[PHP Séance n°1 – introduction & CMS 4](#_toc124)

[Exercice n°1 : principe du PHP 4](#_toc126)

[Exercice n°2 : les programmes de base 4](#_toc132)

[Exercice n°3 : les CMS 5](#_toc155)

[Exercice n°4 : faire un site avec SPIP 5](#_toc157)

[Création du site 5](#_toc158)

[Structuration et données du site : 5](#_toc160)

[La mise en forme du site (squelette) 5](#_toc166)

[Exercice n°4 : insérer une page php dans Spip 6](#_toc168)

[Exercice n°5 : vous pouvez essayer de faire le même travail avec Joomla 6](#_toc171)

[PHP Séance n°2 – introduction 7](#_toc174)

[Exercice n°1 : les types de variables simples 7](#_toc185)

[Exercice n°2 : les chaînes de caractères (les fonctions) 7](#_toc223)

[Exercice n°3 : les conditions 7](#_toc234)

[Exercice n°4 : les boucles (for et while) 8](#_toc268)

[Exercice n°5 : les tableaux (foreach) 8](#_toc293)

[Exercice n°6 : les tableaux associatifs 8](#_toc321)

[Exercice n°7 : les fonctions 9](#_toc376)

[Exercice n°8 : les variables d’environnement 9](#_toc407)

[Exercice n°9 : les dates 9](#_toc427)

[Exercice n°10 : portée des variables (include) 9](#_toc440)

[Exercice n°11 : lire/écrire un nombre dans un fichier texte 9](#_toc472)

[Exercice n°12 : deux pages : GET 9](#_toc504)

[Exercice n°13 : deux pages : POST 10](#_toc569)

[Exercice n°14 : deux pages : le jeu « plus grand-plus petit » 10](#_toc572)

[PHP Séance n°3 – PHP et Mysql 11](#_toc636)

[Exercice n°0 : connexion-déconnexion à une BDD mysql (1 seule table) 11](#_toc638)

[Exercice n°1 : select dans un tableau – 1 page 11](#_toc669)

[Exercice n°2 : gestion des erreurs SQL 12](#_toc679)

[Exercice n°3 : Insert – 2 pages 12](#_toc683)

[Exercice n°4 : Delete – 1 page 12](#_toc689)

[Exercice n°5 : Update – 2 pages 13](#_toc694)

[Exercice n°6 : les Magic Quotes 13](#_toc697)

[Exercice n°7 : session simple (le panier !) 13](#_toc701)

[Exercice n°8 : protéger des pages - session 14](#_toc710)

[PHP Séance n°4 - PROJET – 4h 15](#_toc778)

[Exercice n°1 : structure d’une application (pour déploiement/maintenance facilité) 15](#_toc779)

[Exercice n°2 : projet à réaliser. Trombinoscope 15](#_toc793)

[PHP Séance n°6A – devoir = 2h (version 2008) 17](#_toc814)

[Devoir papier : écrire 3 algorithmes 17](#_toc815)

[Devoir machine : écrire 2 algorithmes 17](#_toc823)

[PHP Séance n°6B = 2h 18](#_toc828)

[Exercice n°1 : Utilisation de la classe fpdf 18](#_toc829)

[Installez et testez la classe fpdf et ses exemples 18](#_toc830)

[Utilisation de la classe fpdf 18](#_toc832)

[Exercice n°2 : Utilisation de la classe fpdf 18](#_toc839)

[PHP Séance n°5 – 4h 19](#_toc842)

[Exercice n°1 : Yahoo YUI et Google AJAX Libraries API 19](#_toc844)

[Exercice n°2 : Ajax pour envoyer un nom de fichier. 19](#_toc858)

[Exercice n°3 : deux pages : le jeu « plus grand-plus petit » avec Ajax 19](#_toc863)

[Exercice n°4 : reprendre l’application « trombinoscope ». 20](#_toc866)

[PHP Séance n°7 – Transaction & ADODB 21](#_toc868)

[Exercice n°1 : la banque – les transactions 21](#_toc869)

[Exercice n°2 : Analyser le corrigé du TP6 21](#_toc879)

[Exercice n°3 : Analyser le corrigé du TP3 : l’adapter à ADODB 21](#_toc885)

[Exercice n°4 : reprendre le corrigé du trombinoscope : l’adapter à ADODB 21](#_toc894)

[Exercice n°5 : projet – gestion de bateaux 22](#_toc897)

[PHP Séance n°8 – SMARTY 23](#_toc913)

[Exercice n°0 : Tester SMARTY 23](#_toc934)

[Exercice n°1 : Tester les exemples SMARTY 23](#_toc941)

[Exercice n°2 : reprendre trombinoscope avec SMARTY 24](#_toc952)

[Exercice n°3 : projet – gestion de bateaux avec SMARTY 24](#_toc954)

[PHP Séance n°9 – Devoir 4h 25](#_toc956)

[Le projet gestion de … 25](#_toc957)

[Le sujet 25](#_toc958)

[La réalisation : écrire les 5 pages php. 25](#_toc960)

[PHP Séance n°10 – utilisation d’outils PHP 26](#_toc962)

[Exercice n°1 : PHPCheckStyle 26](#_toc964)

[Exercice n°2 : insérer googleMap en PHP 26](#_toc966)

[Exercice n°3 : AjaxTerm : un xterm via le web 26](#_toc969)

[Exercice n°4 : GLPI 26](#_toc971)

[Exercice n°5 : Nagios 26](#_toc974)

# PHP Séance n°1 – introduction & CMS

(13 exercices corrigés) vous pouvez consulter <http://fr.php.net/tut.php>

## Exercice n°1 : principe du PHP

* [installer un éditeur de texte](http://fr.wikipedia.org/wiki/PHP)
* [installer WAMP ou LAMP ou EasyPHP ou MAMP](http://fr.wikipedia.org/wiki/PHP)
* [Le navigateur](http://fr.wikipedia.org/wiki/PHP)
* [Mettre le fichier « .php » dans C:\wamp\www ou /var/www/ ou autre...](http://fr.wikipedia.org/wiki/PHP)

## Exercice n°2 : les programmes de base

Quelques outils pour commencer :

* En local (tout sur votre ordinateur : plus rapide !) :
  + Un serveur WEB Apache avec le module PHP
  + Un éditeur de texte (VI, Quanta+, Notepad++, Dreamwaver disponible à l’iut))
  + Un navigateur
* Sur un Serveur Distant (en production !)
  + Un serveur WEB distant avec PHP et un serveur FTP
  + Un éditeur de texte
  + Un client FTP (pour déposer les fichiers php)
  + Un navigateur

Sous windows, vous avez des programmes **WAMP** (Windows-Apache-Mysql[[1]](#footnote-2)-Php), les pages sont dans le dossier C:\wamp\www\.

Sous linux, on parle de **LAMP** (Linux-Apache-Mysql-Php), les pages sont dans le dossier /var/www/

A faire : installer le système linux pour tester le programme « bonjour.php » suivant :

<html><body>

<?php

**echo** " bonjour " **;**

**print** " tout le monde "**;**

?>

</body></html>

Attention !!! : [Fichier][ouvrir][c:\wamp\www\ tp1\e0.php] **≠** « http://localhost/ tp1/e0.php »

Regardez le « code source » de la page web…

## Exercice n°3 : les CMS

## Exercice n°4 : faire un site avec SPIP

### Création du site

Écrire l'adresse : http://localhost/spip/ecrire/

### Structuration et données du site :

Faire un site qui reprend ce cours « Introduction PHP »

Vous créerez une rubrique par séance : 10 en tout.

Chaque rubrique possèdera une sous rubrique par exercice.

Chaque sous-rubrique possèdera au moins un article (l’énoncé) et des fichiers ou des liens s’ils sont disponibles…

N’oubliez pas de publier les articles

### La mise en forme du site (squelette)

Installer les « plugins » associés à SPIP.

## Exercice n°4 : insérer une page php dans Spip

Il vous sera utile d’inclure des pages php dans votre article spip.

<http://www.eklesia.net/spip.php?article47>

## Exercice n°5 : vous pouvez essayer de faire le même travail avec Joomla

# PHP Séance n°2 – introduction

Vous pouvez consulter la documentation, et particulièrement l’« Index des fonctions »

Rappel :

<html><body>

<?php

echo " bonjour " ;

print " tout le " ;

printf ("%s" , " monde ") ;

?>

</body></html>

## Exercice n°1 : les types de variables simples

Les variables sont définies par le caractère « $ ». Elles ne sont pas typées.

Définir et afficher les variables suivantes :

<html>

<body>

<?php

$a= 1 ; // un commentaire

$b= 21.55 ;

//$c= 22,66 ; la virgule ne marche pas !

$d= $a + $b ;

$e= " attention \n";

$f= " à vous";

$g= $e.$f ;

$h= " le resultat = ". $a+$b ;

$i= $a<$b ;

echo $e.$f;

echo $a."<br />";

echo $b."<br />";

echo $c."<br />";

echo $d."<br />";

echo $e."<br />";

echo $f."<br />";

echo $g."<br />";

echo $h."<br />";

$h= " le resultat = ". $a ;

echo $h."<br />";

$h= " le resultat = ". {$a+$b} ;

echo $h."<br />";

$h= " le resultat = ". ($a+$b) ;

echo $h."<br />";

echo $i."<br />";

?>

</body>

</html>

## Exercice n°2 : les chaînes de caractères (les fonctions)

<html>

<body>

<?php

$ch1 = " Au début, ils rient de vous, ensuite ils vous combattent, et à la fin, vous gagnez <br>";

$ch1 = str\_replace ('gagnez','<b>GAGNEZ</b>',$ch1);

$ch1 = str\_replace ('<b>de vous</b>','',$ch1);

echo $ch1 ;

?>

</body>

</html>

## Exercice n°3 : les conditions

Structure conditionnelle :

<html>

<body>

<?php

$a = 6;

$note = 14;

$pair = $a%2;

if ($pair == 0){

echo "La variable est paire \n </br>"; // \n sert à aller à la ligne dans un paragraphe

}

else if ($pair == 1){

echo "La variable est impaire \n </br>"; // </br> sert à aller à la ligne

}

if ($note < 8){

echo "Mauvais";

}

else if ($note >= 8 && $note < 10){

echo "Pas bon";

}

else if ($note >= 10 && $note < 12){

echo "Correct";

}

else if ($note >= 12 && $note < 16){

echo "Bon";

}

else if ($note >=16 && $note <= 20){

echo "Tres bien";

}

?>

</body>

</html>

## Exercice n°4 : les boucles (for et while)

Écrire dans un tableau HTML la table de multiplication de 10x10. Vous utiliserez la boucle « for » et la boucle « while » :

<html>

<body>

<?php

echo "<table border=1>";

$i = 1;

while ($i <= 10){

echo "<tr>";

$j = 1;

while ($j <= 10){

$result = $i \* $j;

echo "<td> $result </td>";

$j++;

}

echo "</tr>";

$i++;

}

echo "</table>";

?>

</body>

</html>

## Exercice n°5 : les tableaux (foreach)

Dans un fichier « e5.php », reprendre l’exercice précédent en stockant tous les résultats de la table de multiplication préalablement dans un tableau :

<html>

<body>

<?php

/\*\*\*\*\*\*\*Definition du tableau\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for($i = 1; $i <= 10; $i++)

for ($j = 1; $j <= 10; $j++)

$multiplication[$i][$j]= $i \* $j;

/\*\*\*\*\*\*\*Affichage du tableau\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo "<table border=1>";

foreach($multiplication as $tab){

echo "<tr>";

foreach($tab as $result){

echo "<td> $result </td>";

}

echo "</tr>";

}

echo "</table>";

print\_t($multiplication);

?>

</body>

</html>

## Exercice n°6 : les tableaux associatifs

Dans un fichier « e6a.php », reprendre l’exercice précédent en stockant tous les résultats de la table de multiplication préalablement dans un tableau :

<html>

<body>

<?php

//$multiplication[10][10];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for($i = 1; $i <= 10; $i++)

for ($j = 1; $j <= 10; $j++){

$li="ligne".$i;

$co="colonne".$j;

$multiplication[$li][$co]= $i \* $j;

}

print\_r($multiplication);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

echo "<table border=1>";

foreach($multiplication as $tab){

echo "<tr>";

foreach($tab as $result){

echo "<td> $result </td>";

}

echo "</tr>";

}

echo "</table>";

?>

</body>

</html>

<html>

<body>

<?php

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for($i = 1; $i <= 10; $i++)

for ($j = 1; $j <= 10; $j++)

$tab["pain"] = 1;

$tab["viande"] = 25;

$tab["pates"] = 3;

$tab["sac"] = 15;

$tab["disque"] = 35;

arsort($tab);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

foreach($tab as $course => $prix){ // course est l'indice

echo "$course - $prix <br>"; // prix la est la valeur

}

?>

</body>

</html>

## Exercice n°7 : les fonctions

<html>

<body>

<?php

function estPair($nb) {

if (($nb%2) == 0 ) {

return TRUE;

}

}

function estImpair($nb) {

if (($nb%2) == 1 ) {

return TRUE;

}

}

$num = 6;

if (!is\_int($num)){

echo "Impossible.";

}

else if (estPair($num) == TRUE ){

echo "Le numéro est pair.";

}

else if (estImpair($num) == True ){

echo "Le numéro est impair.";

}

?>

</body>

</html>

## Exercice n°8 : les variables d’environnement

<html>

<body>

<?php

$a = $\_SERVER["SERVER\_NAME"];

$b = $\_SERVER["REMOTE\_ADDR"];

$c = $\_SERVER["DOCUMENT\_ROOT"];

$d = $\_SERVER["SCRIPT\_NAME"];

echo $a."<br/>";

echo $b."<br/>";

echo $c."<br/>";

echo $d."<br/>";

phpinfo();

?>

</body>

</html>

## Exercice n°9 : les dates

<html>

<body>

<?php

$dt = new DateTime('1789-7-14');

echo "Révolution: ".$dt->format('Y'); //1ère manière de faire

<br>

echo "Révolution: {$dt->format('Y-M')}"; //2ème manière de faire

?>

</body>

</html>

## Exercice n°10 : portée des variables (include)

exo10a.php

<?php // ne pas mettre l'entete <html> dans ce fichier !

echo "le fichier 'exo10a.php' est ici.<br/>";

function A()

{

echo "Fonction A";

}

?>

exo10b.php

<html>

<body>

<?php

echo "le fichier 'exo10b.php' est ici.<br/>";

include 'exo10a.php';

A();

?>

</body>

</html>

exo10c.php

<?php

echo "le fichier 'exo10c.php' est ici.<br/>";

include 'exo10b.php';

?>

… et taper http://localhost/exo10c.php

## Exercice n°11 : lire/écrire un nombre dans un fichier texte

<html>

<body>

<?php

// trouver un nb au hasard

$nb = rand(1, 100); // tire un nombre au hasard entre 1 et 100

if (!$handle = fopen("fichier11.txt", 'a+')) {

echo "Impossible d'ouvrir le fichier ($filename)";

exit;

}

// écrire le nb dans le fichier ET le retour à la ligne "\n"

if (fwrite($handle, $nb."\n") === FALSE) {

echo "Impossible d'écrire dans le fichier ($filename)";

exit;

}

echo "L'écriture de ($nb) dans le fichier (fichier11.txt) a réussi<br/><br/>";

fclose($handle);

// lire le nb dans le fichier

$handle = @fopen("fichier11.txt", "r");

if ($handle) {

while (!feof($handle)) {

$buffer = fgets($handle, 4096);

echo "lecture dans le fichier :".$buffer."<br/>";

}

fclose($handle);

}

?>

</body>

</html>

## Exercice n°12 : deux pages : GET

exo12a.php

<!doctype html public "-//W3C//DTD HTML 4.0 //EN">

<html>

<body>

<form action="exo12b.php" method="GET">

Saisissez une valeur.

<input type="text" name="valeur" size=20 maxlength=40">

<br>

<input type="submit" name="valider" value="Valider">

</form>

<?php

//On recupere les variables passer dans le tableau $\_GET[] :

$valeur = $\_GET["valeur"];

//On fais un print des variables obtenues :

Print "Valeur: $valeur <br> ";

?>

</body>

</html>

exo12b.php

<!doctype html public "-//W3C//DTD HTML 4.0 //EN">

<html>

<body>

<?php

//On recupere les variables passer dans le tableau $\_GET[] :

$valeur = $\_GET["valeur"];

//On fais un print des variables obtenues :

Print "Valeur: $valeur <br> ";

?>

<form action="exo12c.php" method="get">

<!-- On va cacher la valeur saisie "reçue" pour la renvoyer à la page c -->

<input type="hidden" name="valeur" value= <?php echo "$valeur"; ?> >

<input type="submit" name="valider" value="Valider">

</form>

</body>

</html>

exo12c.php

<!doctype html public "-//W3C//DTD HTML 4.0 //EN">

<html>

<body>

<?php

//On recupere les variables passer dans le tableau $\_GET[] :

$valeur = $\_GET["valeur"];

//On fais un print des variables obtenues :

Print "Valeur: $valeur <br> ";

?>

<form action="exo12a.php" method="get">

<!-- On va cacher la valeur saisie "reçue" pour la renvoyer à la page a -->

<input type="hidden" name="valeur" value= <?php echo "$valeur" ?> >

<input type="submit" name="valider" value="Valider">

</form>

</body>

</html>

## Exercice n°13 : deux pages : POST

Idem avec « POST » (**method**=”**post**”>).

if(isset($\_POST['nom'])) $nom=$\_POST['nom'];  
else      $nom="";

## Exercice n°14 : deux pages : le jeu « plus grand-plus petit »

<!doctype html public "-//W3C//DTD HTML 4.0 //EN">

<html>

<head>

<title>Jeu</title>

</head>

<body>

<form action="jeu.php" method="get">

<input type="submit" name="nouveau" value="Nouveau Jeu">

</form>

<hr/>

<br/>

<?php

$nouveau = $\_GET["nouveau"];

$jouer = $\_GET["jouer"];

$valeur = $\_GET["valeurJouee"];

$nombre = $\_GET["solution"];

$i = $\_GET["tentative"];

if ( $nouveau == "Nouveau Jeu" || !isset($\_GET["solution"])){

$nombre = rand(0,100);

$i = 1;

}

if ( $jouer == "Jouer"){

if ( $valeur < $nombre && $i<10){

echo "Le nombre est plus grand que ".$valeur;

}

else if ( $valeur > $nombre && $i<10){

echo "Le nombre est plus petit que ".$valeur;

}

else if ( $valeur == $nombre){

echo "Vous avez gagné, le nombre est bien $nombre.";

} else{

echo "Vous avez perdu, vous avez fait 10 tentatives.<br>";

echo "Le nombre était $nombre.";

echo "Une nouvelle partie commence...";

$nombre = rand(0,100);

$i = 0;

}

$i++;

}

?>

<br/><br/>

<form action="jeu.php" method="get">

Saisissez une valeur comprise entre 0 et 100:

<input type="text" name="valeurJouee" value=<?php echo "$valeur" ?> >

<br>

<input type="hidden" name="tentative" value=<?php echo "$i" ?> >

<input type="hidden" name="solution" value=<?php echo "$nombre" ?> >

<br>

<input type="submit" name="jouer" value="Jouer">

</form>

</body>

</html>

# PHP Séance n°3 – PHP et Mysql

(BDD mysql en PHP et Sessions)

## Exercice n°0 : connexion-déconnexion à une BDD mysql (1 seule table)

Créez la BDD « **seance3** » avec la table « **produit** » :

|  |  |  |  |
| --- | --- | --- | --- |
| NP | Produit | Couleur | QS |
| Integer | Varchar(20) | Varchar(20) | Integer |

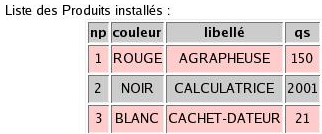
Avec la documentation, créer un fichier « fonctionSQL.php » qui contient la fonction de connexion et la fonction deconnexion. Tester.

$link = **mysql\_connect**('localhost', 'mysql\_user', 'mysql\_password');  
if (!$link) {  
    die('Connexion impossible : ' . mysql\_error());  
}  
echo 'Connecté correctement';  
**mysql\_close**($link);

Exemple de code extrait de la documentation

## Exercice n°1 : select dans un tableau – 1 page

Avec la documentation, créer dans le fichier « fonctionSQL.php » la fonction qui retourne un tableau contenant la liste des noms de la table « produit ». Tester. Afficher la table « produit » dans un tableau html grâce à la page « select.php ».



mysql\_select\_db("mydb");

$result = mysql\_query("SELECT id, name FROM mytable");  
  
while ($row = mysql\_fetch\_array($result, MYSQL\_NUM)) {  
   printf("ID : %s  Nom : %s", $row[0], $row[1]);  
}  
mysql\_free\_result($result);

$result = mysql\_query("SELECT id, name FROM mytable");  
  
while ($row = mysql\_fetch\_array($result, MYSQL\_ASSOC)) {  
   printf("ID : %s  Nom : %s", $row["id"], $row["name"]);  
}  
mysql\_free\_result($result);

Exemple de code extrait de la documentation

## Exercice n°2 : gestion des erreurs SQL

Avec la documentation, modifiez le fichier « fonctionSQL.php » pour afficher un message d’erreur en cas de disfonctionnement. Tester (par exemple, en coupant le serveur mysqld).

echo mysql\_errno($link) . ": " . mysql\_error($link) . "\n";

Exemple de code extrait de la documentation

## Exercice n°3 : Insert – 2 pages

Avec la documentation, créer dans le fichier « fonctionSQL.php » la fonction qui permet d’insérer un nouveau tuple dans la table « produit ». Tester.

La saisie se fera dans le fichier « insert.php ».

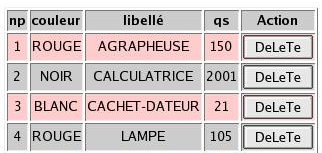
$sql = "INSERT INTO infos\_tbl(id, nom, prenom,email, icq, titre, url) VALUES('','$nom','$prenom','$email','$icq','$titre','$url')";

**mysql\_query**($sql) **or die**('Erreur SQL !'.$sql.'<br>'.**mysql\_error**());   
echo 'Vos infos on été ajoutées.';

Exemple de code pour insérer en php sous mysql

## Exercice n°4 : Delete – 1 page

Créer la page « delete.php » inspirée de la page « select.php », qui contiendra une colonne de plus avec un bouton permettant de supprimer la ligne du bouton cliqué.



Indication importante : pour un formulaire avec un tableau, le nom des variables sera indiqué avec « [ ] », par exemple : (il sera ensuite récupéré en get ou post sous forme de tableau).

**<input type=**”**text**” name=”nom[]” value=”valeur1”>

## Exercice n°5 : Update – 2 pages

Créer la page « update.php » inspirée de la page « select.php », qui contiendra une colonne de plus avec un bouton permettant de modifier les informations de la ligne du bouton cliqué, puis de valider ou d’annuler les changements.



## Exercice n°6 : les Magic Quotes

Pour éviter les « SQL-Injection ».

Voir : <http://www.phpfrance.com/tutoriaux/index.php/2005/04/17/29-les-magic-quotes>

(ou les fichier pdf joint).

## Exercice n°7 : session simple (le panier !)

Pour passer d’une page à l’autre des valeurs sans mettre de formulaires avec des variables cachées (hidden), vous avez la possibilité d’utiliser la variable d’environnement $\_SESSION.

Indication : sous free, vous avez la possibilité de tester cela en créant un dossier « session » à la racine de votre site.

<?php  
// page1.php  
  
session\_start();  
  
echo 'Bienvenue à la page numéro 1';  
  
$\_SESSION['favcolor'] = 'green';  
$\_SESSION['animal']   = 'cat';  
$\_SESSION['time']     = time();

?>

Suivie par

<?php  
// page2.php  
  
session\_start();  
  
echo 'Bienvenue sur la page numéro 2<br />';  
  
echo $\_SESSION['favcolor']; // green  
echo $\_SESSION['animal'];   // cat  
echo date('Y m d H:i:s', $\_SESSION['time']);  
  
// Vous pourriez utiliser la constante SID ici, tout comme dans la page page1.php  
echo '<br /><a href="page1.php">page 1</a>';  
?>

Exemple de code extrait de la documentation

## Exercice n°8 : protéger des pages - session

<?php

include("secret.php");

echo "A";

?>

<br><br>

<a href="b.php">goto b (non securise)</a>

<?php

echo "B";

?>

<br><br>

<a href="index.php">goto index (securise)</a>

<?php

include("secret.php");

echo "toto";

?>

<br><br>

<a href="a.php">goto a</a>

<?php

session\_start();

function c(){

//echo "<br>vous etes connecté<br>";

$\_SESSION['TIMEOUT']=time()+20;

//echo "prochaine deconnexion : ";

//echo time()+20;

}

function d(){

echo "<br>vous etes deconnecté<br>";

session\_unset();

echo "<form action='index.php' method='POST'>

mot de passe : <input type='text' name='mdp'>

<br><br>

<input type='submit' name='action' value='Valider'>

</form>

</center>

<br>";

exit();

}

if (isset($\_POST["mdp"])){ // on vient de saisir le MDP !!

if ($\_POST["mdp"]=="abc"){

c();

return;

}else{

d();

}

}

if (isset($\_SESSION['TIMEOUT'])){ // on vient de saisir le MDP !!

$heure=$\_SESSION['TIMEOUT'];

if ($heure>=time()){

c();

return;

}else{

d();

}

}

d();

?>

# PHP Séance n°4 - PROJET – 4h

## Exercice n°1 : structure d’une application (pour déploiement/maintenance facilité)

L’idée consiste à ne pas avoir de fichier à la racine du projet. Seulement les fichiers « index.php » qui affichera la page désirée.

Structure de l’application : (non exhaustif)

/projet/etc/ ‘ le(s) fichier(s) de configuration

/projet/module/ ‘ les pages « php »

/projet/css/

/projet/js/

/projet/tmp/

/projet/images/

/projet/lib/ ‘ les librairies spécifiques utilisées

Créer un projet qui permet de naviguer entre 2 pages « a.php » et « b.php », d’afficher une image.

## Exercice n°2 : projet à réaliser. Trombinoscope

1) l’administrateur

- il dépose des images dans un répertoire « photo/vrac/ » (au moyen d’un fichier .zip), ou un tableau d’images.

- les groupes disponibles seront définis par l’administrateur directement dans un fichier de configuration.

2) l’étudiant

- lors de son premier passage, il doit donner

- son mail

- son groupe

- et cliquer sur sa photo

- la photo est identifiée, et déplacée dans un répertoire « photo/numéro du groupe/ »

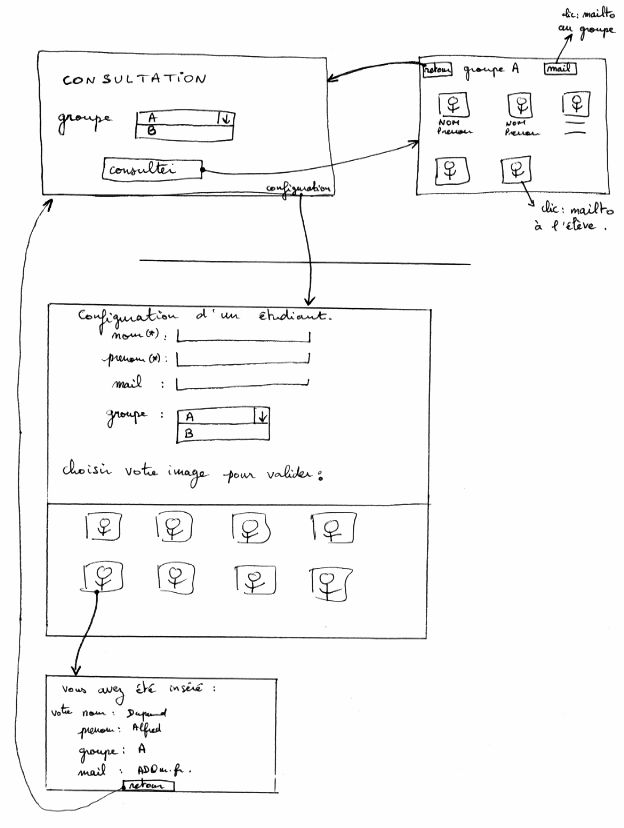
- lors des passages suivants :

- il peut visualiser le trombinoscope (choix par groupe, par nom, par prénom)

2.1 faire un dessin **précis** du site (avec des flèches pour indiquer le passage d’information, le nom et le numéro associé de chaque page)

2.2 définir la table SQL et la créer dans une BDD « trombinoscope ».

2.3 définir et écrire les pages php.



# PHP Séance n°6A – devoir = 2h (version 2008)

## Devoir papier : écrire 3 algorithmes

EXO1 : Écrire deux pages en php. La première permet à l’utilisateur de saisir 3 nombres.

La deuxième page affiche le plus grand des trois nombres.

EXO2 : Expliquez la différence entre "get" et "post".

EXO3 : Écrire une page qui affiche 1500 nombres au hasard. Les nombres (le plus petit et le plus grand) seront aussi affichés sur la première ligne, avant d’afficher les 1500 nombres.

Rq : la fonction « rand » génère une valeur aléatoire

## Devoir machine : écrire 2 algorithmes

EXO1 : Définir le programme « exo1.php » qui affiche en majuscule les fichiers du dossier courant.

EXO2 : Écrire un programme « exo2.php » qui permet de saisir dans un menu déroulant (de 1 à 8) le nombre de champs texte à saisir. Les saisir, valider et les sauvegarder dans une table.

(BDD=  « devoir1 », table =  « exo2 », champ= « nom » (varchar(20)).

# PHP Séance n°6B = 2h

## Exercice n°1 : Utilisation de la classe fpdf

### Installez et testez la classe fpdf et ses exemples

Faire le tutoriel donné avec fpdf.

### Utilisation de la classe fpdf

Faire une page php qui demande :

* Le nom du serveur MySql
* Le nom de la base de données
* Le nom d’utilisateur et son mot de passe
* Le nom de la table

La validation entraine la création d’un fichier pdf qui contient les données du tableau

## Exercice n°2 : Utilisation de la classe fpdf

Ecrire la page php qui génère le trombinoscope au format PDF.

# PHP Séance n°5 – 4h

AJAX avec Google et Yahoo

## Exercice n°1 : Yahoo YUI et Google AJAX Libraries API

Écrire 5 pages php qui utilise les technologies fourines par googe et yahoo.

* Images à déplacer
* saisie de texte
* graphisme (sous forme de camebert)
* petit traitement de texte
* … au choix…

<http://blog.barbayellow.com/2006/09/17/quelle-librairie-javascript-utiliser/>

<http://dojocampus.org/explorer/#Dojo>

<http://code.google.com/apis/ajaxlibs/>

<http://developer.yahoo.com/yui/>

## Exercice n°2 : Ajax pour envoyer un nom de fichier.

On veut envoyer un fichier du poste client vers le serveur.

Avant d’envoyer le fichier (qui peut-être très gros), on veut vérifier par Ajax si le nom du fichier n’existe pas déjà dans le dossier « /tmp/ » du projet.

Si le fichier existe déjà : on le signale à l’utilisateur.

Sinon, on ne dit rien…

## Exercice n°3 : deux pages : le jeu « plus grand-plus petit » avec Ajax

Au début, le jeu place un nombre aléatoire dans un fichier. Le joueur dispose de 10 essais pour trouver le nombre.

## Exercice n°4 : reprendre l’application « trombinoscope ».

Utilisez les technologies Ajax pour rendre plus conviviale le site de trombinoscope (Faites preuve d’imagination).

# PHP Séance n°7 – Transaction & ADODB

## Exercice n°1 : la banque – les transactions

Créer une BDD « banque », une table innoDB « compte » avec 5 clients…

Faire trois pages php :

DAB

Guichet (cheque et liquide)

Achat carte bleue

Le but : faire fonctionner les trois pages en même temps, pour empêcher les conflits d’écritures simultanées sur la table « compte ».

## Exercice n°2 : Analyser le corrigé du TP6

Pour sécuriser un site, vous pouvez « cacher » les pages appelées.

Tester le code suivant : « tp6-num.zip » (avec et sans numéro)

Tester le code suivant « tp7-num-session.zip » (avec numéro & temps de connexion limité).

## Exercice n°3 : Analyser le corrigé du TP3 : l’adapter à ADODB

Installer & Tester « tp7-adodb-tp3-bis.zip »

**ADOBD (Active Data Objects Data Base)** permet d'effectuer simplement des requêtes en SQL sous MySQL, PostgreSQL, Interbase, Informix, Oracle, MS SQL 7, Foxpro, Access, ADO, SAP DB, **SQLite**, Sybase, DB2 pour des applications Web faite en PHP.

sitographie :

<http://fr.wikipedia.org/wiki/ADOdb>

<http://phplens.com/phpeverywhere/adodb_french>

## Exercice n°4 : reprendre le corrigé du trombinoscope : l’adapter à ADODB

- à la fin, installer le trombinoscope avec un autre SGBDR… (en moins de 5 minutes)

## Exercice n°5 : projet – gestion de bateaux

On veut gérer un port de plaisance qui loue des places pour amarrer les bateaux.

Il y a environ 150 emplacements. Chacun pouvant accepter un type de bateau :

Type A : moins de 5 mètres

Type B : moins de 10 mètres

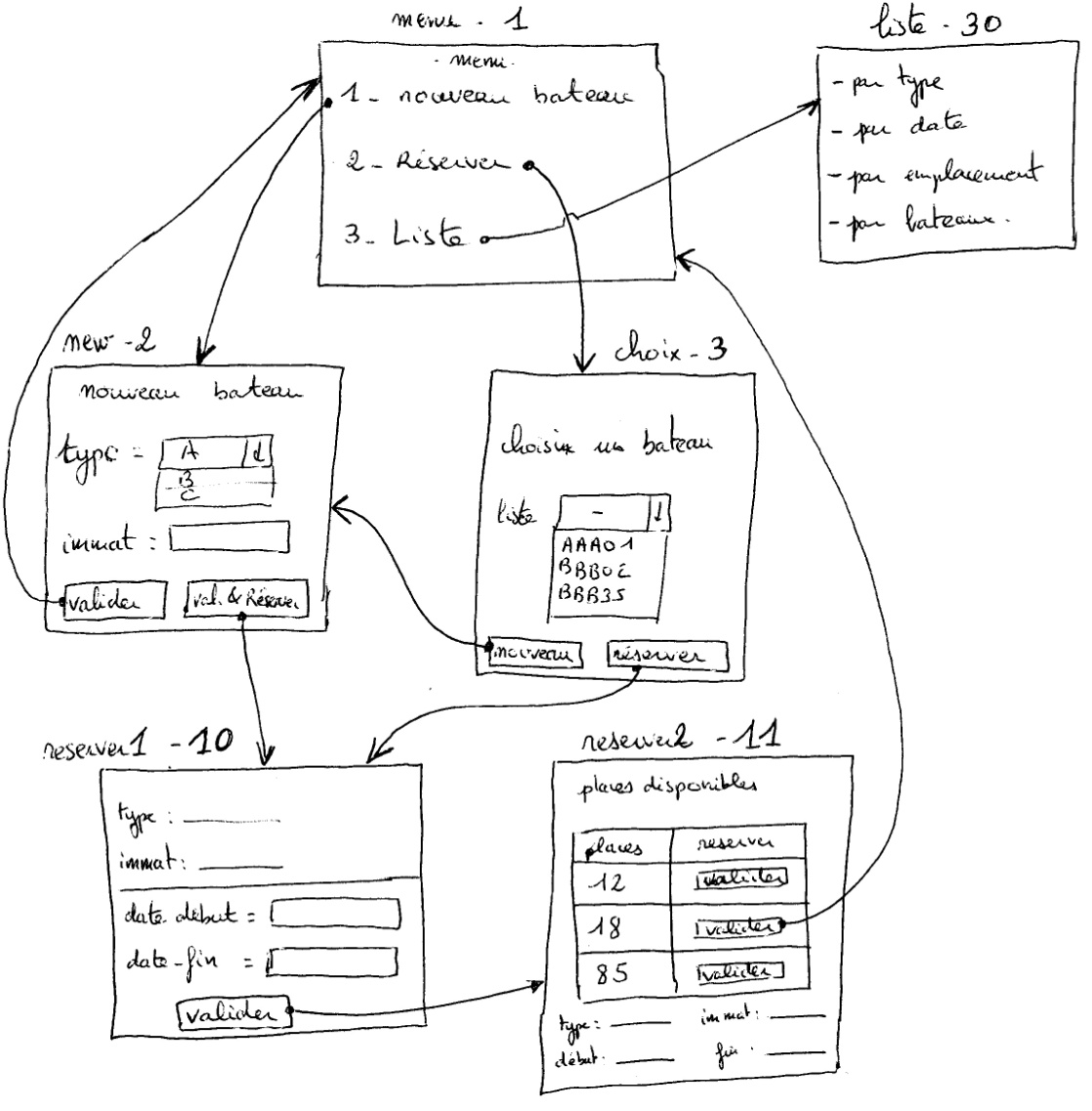
Type C : moins de 17 mètres

Les propriétaires des bateaux font des réservation à partir d’une date pour un nombre de jours données (par exemple, à partir du 10/6/2006 pour 3 jours : l’emplacement sera réservé le 10/6, 11/6 et 12/6, l’emplacement pourra être loué à un autre client à partir du 13/6…).

Les réservations se font exclusivement avec le type de bateaux.

Chaque bateau est identifié par une immatriculation de 3 caractères suivis de 2 chiffres.

Il sera possible dans le futur, d’obtenir l’agrément pour les plus gros bateaux de type D.



# PHP Séance n°8 – SMARTY

<http://fr.wikipedia.org/wiki/Smarty>

Le Modèle MVC « a.php » et « a.tpl »

1 include(\_SMARTY);

2 $smarty = new Smarty;

3 $smarty->template\_dir = './module';

$smarty->compile\_dir = './temp/';

$smarty->config\_dir = './temp/';

$smarty->cache\_dir = './temp/';

4 // le CODE PHP CALCULE les VALEURS

5 $smarty->assign('**tab**',$tableau);

6 $smarty->display("$page.tpl");

<HTML><BODY>

CONSULTER

{html\_options options=**$tab** }

</BODY></HTML>

## Exercice n°0 : Tester SMARTY

Télécharger le projet « tp-smarty.zip »

Copier le dossier « tp-smarty » dans « www »

Créer la BDD « trombi »

Utiliser le script SQL « tp-smarty\Base-de-donnees.sql » pour la BDD « trombi »

Modifier la configuration dans « tp-smarty\etc\variable.inc »

Tester

## Exercice n°1 : Tester les exemples SMARTY

Dans le projet « tp-smarty.zip », créer deux fichiers :

« tp-smarty\module\c.php » et « tp-smarty\module\c.tpl »

Et ajouter dans « tp-smarty\etc\config.inc » l’association « num=3  page=c »

Lire la documentation en Français sur le site de SMARTY

Dans chaque exemple, vous avez 2 parties : **le code php** ET **le code Smarty**. Faire du copier-coller des exemples dans, respectivement, « c.php » et « c.tpl »

Regardez le résultat dans « http://localhost/tp-smarty/index.php?num=3 »

… et recommencez avec divers exemples…

## Exercice n°2 : reprendre trombinoscope avec SMARTY

à la fin, installer le trombinoscope avec une autre BDD… (en moins de 5 minutes)

## Exercice n°3 : projet – gestion de bateaux avec SMARTY

Reprendre l’exercice de la séance n°7 en utilisant SMARTY.

# PHP Séance n°9 – Devoir 4h

## Le projet gestion de …

### Le sujet

### La réalisation : écrire les 5 pages php.

# PHP Séance n°10 – utilisation d’outils PHP

Regardez les applications ci-dessous et installez-testez celles que vous voulez.

## Exercice n°1 : PHPCheckStyle

Configurer et utiliser cet outil pour rendre plus standard votre code php.

## Exercice n°2 : insérer googleMap en PHP

Un exercice simple et sympathique :

<http://fardeen.biz/index.php/tutoriel-application-google-map-avec-php/>

## Exercice n°3 : AjaxTerm : un xterm via le web

<http://nawer.freecontrib.org/index.php?2006/05/24/203-ajaxterm-un-terminal-a-distance-dans-un-simple-navigateur>

## Exercice n°4 : GLPI

Gestion Libre du parc informatique

Voir fichier zip

## Exercice n°5 : Nagios

Gestion du réseau

<http://fr.wikipedia.org/wiki/Nagios>

1. Mysql : Serveur de Base de données [↑](#footnote-ref-2)