

# TP 3 de Programmation Web - Master 1

## Informatique : PHP

Auteur : Olivier GLÜCK, Université Lyon 1

### Objectifs

- Installation et configuration de PHP
- Apprentissage de la programmation Web côté serveur en PHP
- Interfaçage avec une base de données

### Pré-requis

Protocole HTTP, formulaires HTML, principes de la programmation web côté serveur, langage PHP

### NB

Ce TP se déroule sur 4h30 (moitié séance 4 + séance 5). Il est impératif de travailler de façon importante entre chaque séance. Pour ce faire, il est nécessaire de conserver vos fichiers d'une séance sur l'autre.

## 1. Installation d'Apache

Dans ce TP, on vous demande d'installer et de configurer un serveur Apache sur votre poste de travail. Pour contacter votre serveur Web, vous utiliserez le navigateur Mozilla ou Netscape avec l'adresse `http://localhost/` ou `http://192.168.9.xx/`.

0- Pensez à faire la remise à zéro de la configuration des machines avant de commencer le TP...

1- Configurez `eth0` sur le réseau du miroir (192.168.9.9) en vous branchant sur le commutateur central ; vous prendrez l'adresse IP **192.168.9.xx** où **xx** est le dernier octet de l'adresse IP inscrite sur l'étiquette de votre machine. Vérifiez que vous arrivez à « ping » le miroir .

2- Installation des packages

Pour ce TP, les packages qui nous intéressent sont :

<b>apache</b>	1.3.28-2	Versatile, high-performance HTTP server
<b>apache-common</b>	1.3.28-2	Support files for all Apache webserver
<b>apache-doc</b>	1.3.31-1	Apache webserver docs
<b>php4</b>	4.3.4-4	A server-side, HTML-embedded scripting language
<b>mysql-server</b>	3.23.49-8.4	mysql database server binaries
<b>php4-mysql</b>	4.3.4-4	MySQL module for php4

Tapez les commandes suivantes pour installer apache ; vérifiez pour chaque commande qu'il n'y a pas de message d'erreur.

```
ping 192.168.9.9
```

```
export ftp_proxy=http://192.168.9.9:3128/
```

```
export http_proxy=http://192.168.9.9:3128/
```

```
rm -Rf /etc/apache /var/www;
```

```
apt-get update; apt-get remove --purge apache apache-doc apache-common;
```

```
apt-get install nom_pkg
```

3- Pour vérifier que le package est correctement installé, faire un `dpkg -l` et vérifiez que la ligne correspondant au package commence par `i`

Des fichiers (comme par exemple les logos permettant la construction de votre site) sont disponibles sur le serveur de la salle TP. Vous pouvez les récupérer en montant **192.168.9.9:/partage** par NFS (en lecture seule).

A la fin de la séance, n'oubliez pas de sauvegarder vos fichiers pour la séance suivante, soit en utilisant une disquette ou clé USB, soit en copiant vos fichiers dans `/root` et en utilisant `restore`.

## 2. Mise en place de PHP et premiers tests

### **Manipulation**

Installez les packages permettant l'utilisation de `apache`, `php4` et `mysql` (cf. 1). Configurez le serveur `apache` pour qu'il interprète correctement les fichiers ayant pour extension `.php`. Vous pourrez utiliser l'utilitaire `apache-modconf` pour permettre le chargement du module `php4`.

### **Manipulation**

Créez un répertoire `/root/PHP` dans lequel vous placerez l'ensemble de vos fichiers pour ce TP. Utilisez la directive `Alias` pour que `http://localhost/tpphp` corresponde au répertoire `/root/PHP`. Créez, dans ce répertoire, un premier fichier `phpinfo.php` permettant d'afficher tous les paramètres de la configuration actuelle de PHP sur votre serveur. Vérifiez que tout fonctionne correctement.

### **Question et manipulation**

Quelles sont les informations qui sont listées quand vous invoquez `phpinfo.php` ? Où se trouve le fichier de configuration de PHP ? Regardez le source de la page affichée dans votre navigateur après une requête vers `phpinfo.php`. Qu'en concluez-vous ?

### **Question et manipulation**

Quel processus exécute le code PHP sur le serveur ? Combien de processus de ce type sont en cours d'exécution sur votre serveur ? Ajoutez une boucle infinie dans `phpinfo.php` et vérifiez avec `top` votre réponse à la première question. Au bout de combien de temps le processus arrête-t-il d'exécuter la boucle infinie ? Que se passe-t-il alors dans le navigateur ? Où cela se configure-t-il (donnez le nom de la directive correspondante) ?

## 3. Une petite application de gestion d'Images

On vous propose de réaliser une petite application de gestion et d'affichage d'images (ou photos). Cette application, principalement écrite en PHP, sera interfacée avec une base de données `mysql` nommée `photos_base` contenant deux tables :

- une table `mes_photos` contenant les informations relatives à chaque photo publiée sur votre site :
  - o `nom_photo` (nom relatif du fichier) de type `VARCHAR(20) BINARY NOT NULL`,
  - o `categorie` (catégorie à laquelle la photo appartient) de type `VARCHAR(20) BINARY NOT NULL`,
  - o `titre` (permet d'associer un titre à la photo) de type `VARCHAR(50) BINARY`,
  - o `hauteur` (hauteur en pixels de la photo) de type `SMALLINT NOT NULL`,
  - o `largeur` (largeur en pixels de la photo) de type `SMALLINT NOT NULL`,
  - o `date` (date de la prise photo) de type `DATE DEFAULT '0000-00-00'`,
  - o `liste_mots` (mots-clés associés à la photo avec comme séparateur le caractère espace) de type `VARCHAR(100)`
- une table `mes_categories` contenant les informations relatives aux catégories d'appartenance des photos :
  - o `categorie` (nom de la catégorie) de type `VARCHAR(20) BINARY NOT NULL`,
  - o `chemin` (chemin absolu du répertoire contenant les photos de la catégorie) de type `VARCHAR(50) BINARY NOT NULL`,
  - o `passwd` (code d'accès aux photos de la catégorie) de type `VARCHAR(20) BINARY DEFAULT NULL`

L'application devra :

- proposer un formulaire à l'utilisateur lui permettant de choisir les photos qu'il souhaite voir afficher dans son navigateur en fonction soit d'une catégorie, soit d'une liste de mots-clés, soit de la date de prise de la photo, soit d'une combinaison de ces trois critères ;
- gérer l'authentification lorsque l'accès à une catégorie nécessite un mot de passe ;
- gérer l'affichage des photos demandées en utilisant éventuellement le titre et la date de la photo ;

## Manipulation

Créez la base `photos_base` et les tables `mes_photos` et `mes_categories`. La clé primaire de la table `mes_photos` sera le couple (`nom_photo`, `categorie`) et celle de `mes_categories` sera (`chemin`). Il est très fortement conseillé de mettre les commandes SQL de création de chacune des tables dans un fichier `.sql` afin de ne pas avoir à tout retaper s'il faut recommencer...

Récupérez sur le serveur 192.168.9.9 le fichier `BD_TP_PHP.tgz` (par NFS ou ssh). Importez le fichier `photos.txt` dans la table `mes_photos` et ajoutez les catégories correspondantes dans la table `mes_categories`. Pour l'importation de `photos.txt`, utilisez la commande suivante :

```
mysql> LOAD DATA LOCAL INFILE "photos.txt" INTO TABLE mes_photos;
```

Créez un utilisateur php de mysql qui vous permettra de vous connecter à la base avec tous les droits ; pour cela, vous taperez les commandes suivantes :

```
bash$ mysql mysql
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO php@localhost IDENTIFIED BY 'mot_de_passe' WITH GRANT OPTION;
```

## Manipulation

Dans un fichier, nommé `photos.php`, créez un formulaire contenant deux boutons de soumission portant le même nom mais une valeur distincte. La soumission du formulaire doit provoquer l'exécution de `photos.php` (vous pourrez utiliser `$_SERVER['PHP_SELF']`). Dans ce script, écrivez une fonction `aff_table()` permettant d'afficher le contenu d'une table et prenant en paramètres le nom d'une base, le nom d'une table, le nom d'hôte hébergeant la base de données, le nom de connexion et le mot de passe permettant l'accès à la base. Complétez le script pour que les deux boutons du formulaire permettent chacun l'affichage d'une table (resp. `mes_photos` et `mes_categories`). Les affichages liés à une soumission du formulaire devront se faire en-dessous de celui-ci.

## Manipulation

Ajoutez au formulaire précédent :

- un menu déroulant contenant la liste des noms de catégories ; ce menu devra être construit après une interrogation de la table `mes_categories` ;
- un bouton de soumission (portant toujours le même nom mais avec comme valeur « Voir les photos »)

## Manipulation

Modifiez le script pour qu'un click sur « Voir les photos » (après la sélection d'une catégorie) affiche l'ensemble des images de la catégorie sélectionnée. Pour l'instant, ne tenez pas compte du mot de passe éventuel d'accès à la catégorie. Vous afficherez les photos dans un tableau sur 4 colonnes en renseignant correctement les attributs de la balise `<img>` (`src`, `alt`, `width`, `height`) ; l'attribut `alt` devra contenir le titre de la photo ainsi que sa date. Testez votre script en utilisant la base de photos qui vous a été fournie. Voici un aperçu de ce que vous devez obtenir :

## Formulaire d'affichage d'images

Voir table photos

Voir table categories

Sélectionnez une catégorie

Animaux  
Bouffe  
Famille  
Monuments

Voir les photos

### Voici les photos que vous avez sélectionnées !

Il y a 4 photos dans la catégorie Animaux (BD\_TP\_PHP/Animaux, toto1) :



BD\_TP\_PHP/Animaux/DSC00011.jpg "Perroquet" 2004-06-06 h=200 l=150 "perroquet oiseau pirate"  
BD\_TP\_PHP/Animaux/DSC01212.jpg "Chien" 2003-06-06 h=200 l=150 "chien lydie"  
BD\_TP\_PHP/Animaux/DSC01422.jpg "Canard" 2004-06-06 h=200 l=150 "canard versailles"  
BD\_TP\_PHP/Animaux/DSC01446.jpg "Chevre" 2004-06-06 h=200 l=150 "chevre versailles"

### Manipulation

Ajoutez maintenant la gestion de l'authentification si un code d'accès à la catégorie est nécessaire.

### Manipulation

Utilisez une session PHP pour conserver les mots de passe des catégories déjà accédées et ainsi éviter à l'utilisateur de ressaisir systématiquement le mot de passe. S'il vous reste du temps, vous pouvez améliorer votre application en enrichissant le formulaire pour que le choix des photos puisse également se paramétrer en fonction de la date de prise de la photo, des mots-clés associés aux photos, ... Vous pourrez ajouter un bouton permettant de voir la liste des mots-clés disponibles (pour faire du « exact\_match »). Vous pourrez également proposer une interface permettant d'ajouter des photos dans la table mes\_photos... Donnez libre court à votre imagination.

# BDW1 - TD 6

## Programmation Web - PHP

UCBL - Département Informatique de Lyon 1 – 2018

L'objectif de ce TD est de vous familiariser avec la syntaxe HTML et PHP. Pour cela, nous allons nous mettre en place des fonctions qui vous permettront de construire l'espace de jeu utile au projet.

Nous considérons le fichier *mesFonctions.php* dont les premières lignes sont les suivantes :

```
<?php
\* Declaration des constantes *\
define( 'RIEN' , 'vide' );
define( 'DALLE', 'dalle' );
define( 'FIN' , 'carotte' );
define( 'TROU', 'trou' );

\* Declaration des variables globales *\
$dim = 5;           // dimension du plateau
$numCase = 0;       // compteur d'affectation d'un numero de dalle
$plateau = array(); // espace de jeu
$schemin = array(); /* coordonnees de dalles dans le plateau
                      correspondant au chemin */

...
?>
```

Une fois toutes les fonctionnalités implémentées, vous serez en mesure de générer un espace de jeu composé d'un chemin lui-même composé de dalles numérotées aboutissant à une dalle finale représentant une carotte. Trois chemins en spirale sont représentés sur la figure 1.

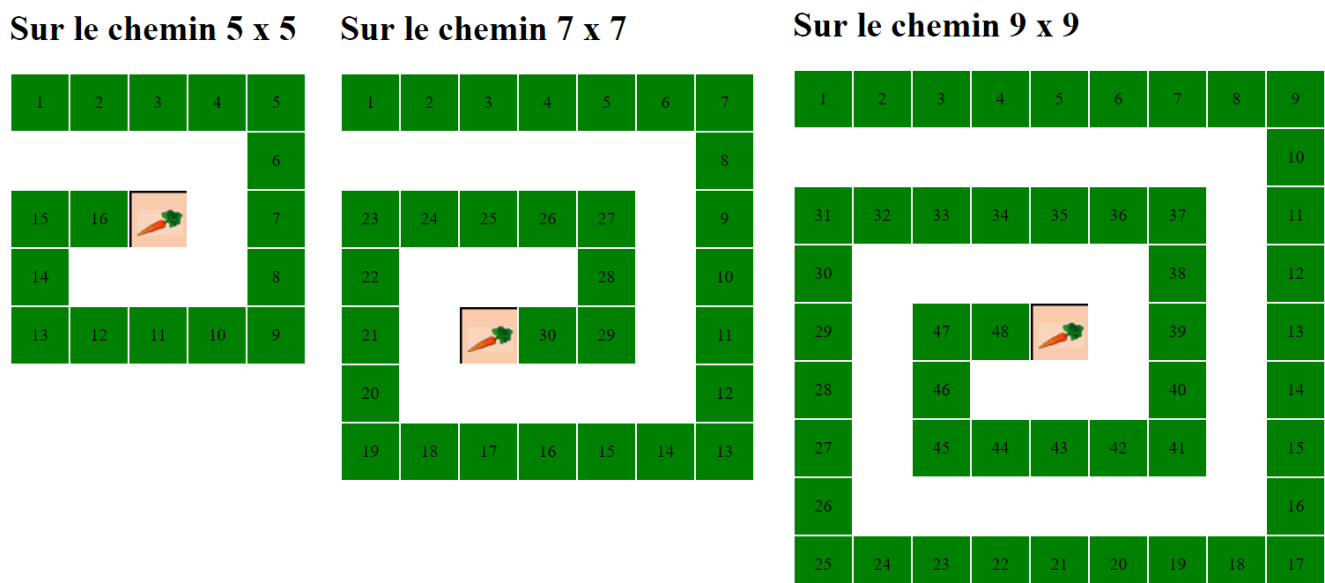


FIGURE 1 – Trois chemins pour des valeurs de \$dim valant 5, 7 et 9.



## Exercice 1 : Construction d'un chemin en spirale

Écrire les fonctions PHP répondant aux besoins spécifiés dans les questions.

1. Définir la fonction *initPlateau()* qui initialise la variable globale *\$plateau* en un tableau à 2 dimensions de taille  $\$dim \times \$dim$ , rempli par défaut avec la valeur de la constante RIEN.
2. Définir le prédicat *estLibre(\$x,\$y)* qui retourne faux si les coordonnées ( $\$x, \$y$ ) ne correspondent pas à un élément du plateau ou si les coordonnées correspondent à une valeur dans le plateau qui n'est pas la valeur de la constante RIEN. Sinon la fonction retourne vrai.
3. Définir le prédicat *estAuBord(\$x,\$y,\$sens)* qui, étant donnée la cellule courante ( $\$x, \$y$ ) dans le plateau et le sens d'avancement  $\$sens$ , retourne vrai, :
  - si la valeur de  $\$y$  correspond à la dernière colonne du plateau, pour  $\$sens$  qui vaut 'est' ;
  - si la valeur de  $\$x$  correspond à la dernière ligne du plateau, pour  $\$sens$  qui vaut 'sud' ;
  - si la valeur de  $\$y$  correspond à la première colonne du plateau, pour  $\$sens$  qui vaut 'ouest' ;
  - si la valeur de  $\$x$  correspond à la première ligne du plateau, pour  $\$sens$  qui vaut 'nord' ;Sinon la fonction retourne faux.
4. Définir le prédicat *avancerEstPossible(\$x,\$y,\$sens)* qui, à partir de la cellule courante ( $\$x, \$y$ ), retourne vrai si les deux cases suivantes selon la direction  $\$sens$  sont libres ou si la case suivante (toujours selon la direction  $\$sens$ ) est au bord du plateau et libre.
5. Définir la fonction *avance(\$x,\$y,\$sens)* qui, étant donnée la cellule courante ( $\$x, \$y$ ) dans le plateau, retourne un tableau associatif de clés 'x' et 'y' de valeurs respectives correspondantes aux coordonnées de la cellule :
  - à droite de ( $\$x, \$y$ ), pour  $\$sens$  qui vaut 'est' ;
  - au dessous de ( $\$x, \$y$ ), pour  $\$sens$  qui vaut 'sud' ;
  - à gauche de ( $\$x, \$y$ ), pour  $\$sens$  qui vaut 'ouest' ;
  - au dessus de ( $\$x, \$y$ ), pour  $\$sens$  qui vaut 'nord' ;
6. Définir la fonction *sensSuivant(\$sens)* qui retourne le point cardinal qui se trouve à tribord (droite) du point cardinal  $\$sens$ . Donc pour 'est' la fonction retournera 'sud', pour 'sud' la fonction retournera 'ouest'...
7. Définir le prédicat *peutTournerTribord(\$x,\$y,\$sens)* qui, étant donnée la cellule courante ( $\$x, \$y$ ) dans le plateau, retourne vrai s'il existe au moins deux cases libres adjacentes dans la direction se trouvant à tribord (par la droite) du sens de direction  $\$sens$ .
8. Définir la fonction *affecte(\$x,\$y,\$val)* qui :
  - si  $\$val$  correspond à la valeur de la constante 'DALLE', affecte :
    - les coordonnées ( $\$x, \$y$ ) dans le chemin  $\$chemin$  (La place de la dalle dans le chemin sera géré *via*  $\$numCase$ ).
    - le numéro de case dans la case ( $\$x, \$y$ ) du plateau.
  - sinon, affecte la valeur  $\$val$  aux coordonnées ( $\$x, \$y$ ) dans le plateau
9. Définir la fonction *chemine(\$x,\$y,\$sens)* qui permet de construire un chemin à travers le plateau selon le principe : "A partir de la cellule initiale ( $\$x, \$y$ ), on avance dans la direction  $\$sens$  tant que possible. Si ce n'est plus possible, on cherche toujours à tourner à tribord (droite). Quand il n'est plus possible d'avancer et de tourner à tribord, la dernière dalle du chemin est marquée dans le plateau par la valeur de la constante 'FIN'".

## Exercice 2 : Affichage d'un chemin en spirale

1. Définir la fonction *printChemin()* qui à partir du plateau construit un tableau HTML où chaque cellule du plateau correspond à une case du tableau. En fonction de la valeur de la cellule du plateau, vous affecterez à la case la classe correspondant à la valeur de la cellule. Attention, dans le cas où la valeur de la cellule est numérique, il faudra affecter la classe 'dalle'.
2. Définir un fichier *monStyle.css*. Pour chaque valeur de constante définie dans le fichier *mesFonctions.php*, déclarer un sélecteur de classe. Au sélecteur de cellule de tableau, associer une hauteur et une largeur de 50px. A chaque sélecteur de classe, associer un fond comme suit :
  - fond blanc pour le sélecteur *.vide*
  - fond vert pour le sélecteur *.dalle*
  - fond avec image "carotte.png" pour le sélecteur *.carotte*
  - fond noir pour le sélecteur *.trou*
3. Définir le fichier *monJeu.php* qui permet l’affichage d’un chemin en spirale. Ce fichier référencera le fichier *monStyle.css* et inclura le fichier *mesFonctions.php*.

### Exercice 3 : Affichage de trou dans le chemin

1. Dans le fichier *mesFonctions.php*, définir la fonction *placeTrou(\$case)* qui permet d’affecter la valeur de la constante 'TROU' dans le plateau aux coordonnées correspondant au numéro de case \$case.

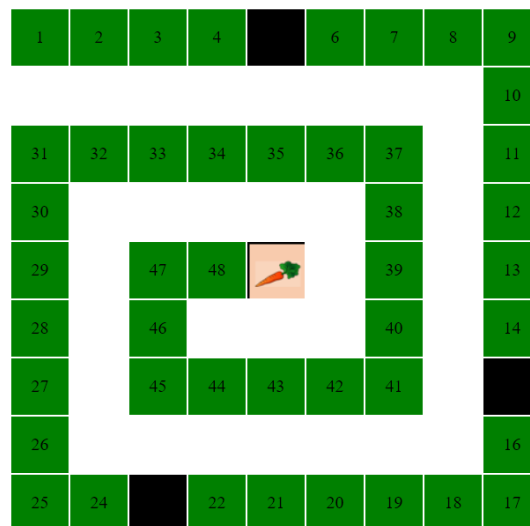


FIGURE 2 – Chemin 9x9 avec des trous aux cases 5, 15, 23

# BDW1 - TD 7

## Programmation Web HTML/CSS/PHP

UCBL - Département Informatique de Lyon 1 – 2018

On considère la base de données suivante :

*CARTE*(valeur, couleur, points, image)

*MAIN*(joueur, valeur, couleur)

permettant de stocker les un jeu de cartes permettant de jouer à la belote. Les valeurs sont réparties de la manière suivante :

valeur	points	points atout
7	0	0
8	0	0
9	0	14
10	10	10
valet	2	20
dame	3	3
roi	4	4
as	11	11

### Exercice 1 : Manipulation des cartes

On suppose qu'une connexion a été préalablement définie et stockée dans la variable globale `$c`.

1. Ecrire une fonction PHP qui permet d'insérer les cartes dans la table *CARTE* pour les 4 couleurs ('coeur', 'carreau', 'pique', 'trèfle') avec leurs points (valeurs normales non atout). On suppose disposer d'images .PNG représentant les cartes dans le répertoire 'imgCartes' se trouvant au même niveau que le fichier PHP que vous êtes en train d'écrire. Les noms des fichiers sont définis de la forme suivante "<couleur>\_\_<valeur>.PNG".
2. Ecrire une fonction PHP qui prend en argument une couleur d'atout et qui met à jour la table *CARTE* pour associer les valeurs d'atout aux cartes.
3. Ecrire une fonction PHP *distribueCarte* qui prend en paramètre un nombre de cartes *\$nbCarte* et qui insère dans la table *MAIN* *\$nbCarte* cartes choisies aléatoirement dans *CARTE* et ce, pour 4 joueurs : 'joueur1', 'joueur2', 'joueur3' et 'joueur4'. Il sera nécessaire de s'assurer qu'une même carte n'est pas distribuée à plusieurs joueurs.

### Exercice 2 : Affichage d'une carte

1. Définir une fonction qui prend en paramètre une valeur de carte et une couleur de carte et qui retourne le code HTML permettant de créer une carte avec le contenu de la figure au verso.



2. Définir un CSS permettant d'afficher une carte.
3. Modifier votre fonction d'affichage d'une carte et le CSS pour cacher les cartes par défaut et les afficher quand la souris la survole.



## - TD n°1 -

### Mise en place et premiers tests

Tous les scripts PHP que vous allez tester doivent être servis par un serveur web proprement configuré, sans quoi vous ne pourrez que lire la source du script PHP sans l'interpréter. Vous utiliserez donc un espace de publication web sur votre partition sur Venus.

#### Exercice 1.

1. Si cela n'a pas déjà été fait lors des TP en Technologie Web de 1ère année, créez un répertoire nommé `public.html` dans votre partition sur Venus, et assurez-vous qu'il est bien accessible en lecture par tout le monde (commande `chmod o+rx public.html`).
2. Créez dans ce répertoire un fichier `test.php` contenant le script suivant :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html14/strict.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Page de test</title>
  </head>
  <body>
    <h1>Page de test</h1>
    <?php echo "Hello World! \n<br />Bonjour à tous!"; ?>
  </body>
</html>
```

L'utilisation de l'éditeur de texte **Emacs** est fortement recommandée pour la réalisation de vos pages `.html` et vos scripts `.php`. Dans la plupart des cas, vos scripts PHP seront inscrits dans un squelette de document HTML, comme ci-dessus.

3. Pour visualiser le résultat de votre script dans une fenêtre de navigateur, tapez son URL du type `http://venus/~nom_utilisateur/test.php`.
4. Affichez le code source de la page dans votre navigateur. Que constatez-vous ? A quoi servent la balise `<br />` et la séquence `\n` ? Testez leur utilité en modifiant la chaîne de caractères précédente.
5. Que se passe-t-il si vous passez au navigateur une URL du type `file://~nom_utilisateur/test.php` ?

#### Exercice 2.

1. Ecrivez dans un fichier `info.php` le script suivant :

```
<?php
phpinfo();
?>
```
2. Visualisez le résultat de votre script dans un navigateur. Quelles sont les informations fournies concernant votre serveur ?

## Variables, fonctions, structures de contrôle, tableaux

### Exercice 3. Variables d'environnement

1. Créez un script dans lequel vous appellerez la fonction `echo`, ainsi que les variables d'environnement `$_SERVER['PHP_SELF']`, `$_SERVER['SCRIPT_FILENAME']` et `$_SERVER['SCRIPT_URI']` afin d'afficher le nom du fichier du script en cours d'exécution, son adresse absolue et son URI.
2. Quelle est la variable prédéfinie qui contient l'adresse IP du client qui demande la page courante ? Quelle est celle qui contient les langues acceptées par le navigateur du poste client ? Affichez leurs valeurs dans une page web avec la fonction `echo`.

**Exercice 4.** Affectez à une variable `$a` une valeur entière quelconque. Affichez (avec `echo`) la chaîne de caractères `J'ai X ans`, où `X` est remplacé par la valeur de `$a`. Vous procéderez de deux façons :

- en utilisant la concaténation de chaînes de caractères
- en passant 3 arguments à la fonction `echo`.

**Exercice 5.** Que fait le script suivant ?

```
<?php
$dept=75 ;
if ($dept == 75) echo "Paris" ;
if ($dept == 78) echo "Yvelines" ;
if ($dept == 91) echo "Essonne" ;
if ($dept == 92) echo "Hauts de Seine" ;
?>
```

Réécrivez un script produisant le même résultat à l'aide une instruction `switch ... case`.

**Exercice 6.** Recopiez dans un script la fonction suivante :

```
function mystere($n)
{
    $val=1 ;
    for ($i=1 ; $i<=$n ; $i++)
        $val*=$i ;
    return $val ;
}
```

Appelez cette fonction en lui passant la valeur 4 en argument et affichez le résultat. Que fait cette fonction ?

### Exercice 7.

1. A l'aide de la fonction prédéfinie `date`, affichez dans une page web la date d'aujourd'hui sous le format suivant :  
`Aujourd'hui, nous sommes le 15/09/2006 ; il est 10:16:18`
2. Ecrire une fonction prenant en argument une année `$a` et retournant `true` si l'année est bissextile, et `false` sinon.  
**Rappel :** une année est bissextile si elle est divisible par 4, à l'exception des années divisibles par 100 qui ne sont bissextiles que si elles sont également divisibles par 400 (ex : 2000 est bissextile, mais 1900 ne l'est pas).

## Exercice 8.

1. Recopiez le script suivant et visualisez-le dans une page web

```
<?php
$clients = array(
    "client 1"=>array("nom 1"=>"Dulong","ville 1"=>"Paris","age 1"=>"35"),
    "client 2"=>array("nom 2"=>"Leparc","ville 2"=>"Lyon","age 2"=>"47"),
    "client 3"=>array("nom 3"=>"Dubos","ville 3"=>"Tours","age 3"=>"58"));
echo "<table border='1' width='100%'> \n
<tr> <th> Client </th> <th> Nom </th> <th> Ville </th> <th> Age </th>
</tr>\n";
foreach($clients as $cle=>$tab)
{
    echo "<tr><td><b>$cle</b></td>\n";
    foreach($tab as $key=>$val)
        echo "<td> $key : <b>$val</b> </td>\n";
    echo "</tr>\n";
}
echo "</table>";
?>
```

2. Ajoutez dans le tableau un client référencé comme client n°7, de nom Duval, domicilié à Marseille et âgé de 24 ans.
3. Modifiez le script précédent de telle sorte que le tableau associé à chaque client ne contienne que le nom, la ville et l'âge sans préciser de clefs particulières. Simplifiez le code afin d'obtenir l'affichage suivant :

Client	Nom	Ville	Age
client 1	Dulong	Paris	35
client 2	Leparc	Lyon	47
client 3	Dubos	Tours	58
client 7	Duval	Marseille	24

**Exercice 9.** Ecrivez une fonction PHP prenant en argument un nombre entier strictement positif  $n$ , et produisant (avec `echo`) un tableau HTML contenant la table de multiplication de 1 à  $n$ . Affichez la table pour  $n$  égal à 6. Le tableau devra avoir l'allure suivante :

x	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	8	10	12
3	3	6	9	12	15	18
4	4	8	12	16	20	24
5	5	10	15	20	25	30
6	6	12	18	24	30	36

## Installation d'un serveur local sur votre ordinateur personnel

Si vous souhaitez continuer vos scripts PHP chez vous, il est fortement conseillé d'installer sur votre ordinateur personnel un serveur local simulant votre serveur distant. Cette démarche vous évitera d'avoir à transférer, pour chacun de vos tests, tous vos fichiers sur le serveur distant de votre hébergeur.

Il existe sur le web des packages complets pour Windows ou Mac, qui permettent d'installer en une seule opération tous les éléments constituant un serveur local (serveur web Apache, interpréteur de code PHP, base de données MySQL 5, phpMyAdmin...).

1. Sur windows, télécharger et installer le package **WAMP5 1.6.4a** disponible sur

<http://www.wampserver.com>

dans le dossier **C:\wamp5** par exemple. Tous les scripts devront alors être enregistrés dans le sous-dossier **www**, dans lequel vous pourrez créer plusieurs sous-dossiers correspondant à vos différents sites. Lors du redémarrage du serveur, ils apparaîtront dans la rubrique **Vos projets** de la page d'accueil de Wampserveur.

2. Sous Mac OS X, le logiciel **Apache 1.3** est déjà installé par défaut. Pour démarrer le serveur, il suffit d'activer le service **Partage Web personnel** dans la rubrique **Partage des Préférences Système**. Le module PHP correspondant au serveur web Apache inclus dans Mac OS X est téléchargeable sur

<http://www.entropy.ch/software/macosx/php>

Il vous suffit alors de suivre les instructions d'installation du package **PHP 5.1.4** pour **Apache 1.3**.

### Exercice 1 : Premiers pas en PHP

Voici un exemple de code PHP.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
4 lang="en">
5 <head>
6 <title> <?php echo 'Premiers pas en PHP'; ?> </title>
7 <meta http-equiv="Content-Type" content="text/html;
8 charset=UTF-8" />
9 </head>
10 <body>
11 <h1> Mes premiers pas en PHP </h1>
12 <?php $temps=2; echo '<p> Je débute depuis ' ; echo $temps;
13 echo ' heures... </p>'; ?>
14 <p> Mais cela a l'air intéressant <?php echo '!' ?> </p>
15 <?php echo '
16 <h2> Vive le PHP </h2>
17 <p> Les pages vont pouvoir être dynamiques! </p>
18 ' ; ?>
19 <p> Encore quelques paragraphes </p>
20 echo '<p> Avant dernier paragraphe </p>';
21 <p> Voilà, c'est terminé! </p>
22 </body>
23 </html>
```

Répondre aux questions suivantes :

1. Donner dans ce fichier les parties correspondant à du code PHP et à du code XHTML.
2. Si ce fichier s'appelle exo1.php, et s'il est stocké sur le site Web [www.exemple.org](http://www.exemple.org) dans le répertoire PremierTP, comment "exécuter" ce fichier ? Que donne son exécution ? Ceci est-il valide ? Pourquoi ? Corriger alors le problème.
3. Remplacer les trois instructions echo de la ligne 10 par une seule.

### Exercice 2 : Inclusion d'en-tête et pied de page

Le langage PHP permet d'inclure des fichiers dans d'autres. Ceci permet alors de décomposer

un code XHTML ou PHP en plusieurs parties logiques et d'insérer ensuite ces différentes parties

dans un fichier PHP. Utiliser l'inclusion de fichiers pour séparer, dans le fichier précédent, le corps

du document du reste. Quel est l'intérêt d'une telle décomposition ?

### Exercice 3 : Affichage des carrés de tous les nombres entre 1 et 30\*

Créer une page PHP permettant d'afficher, sous forme de liste non ordonnée, les carrés des nombres de 1 à 30 selon le format :

- 12 = 1



- 22 = 4

...

- 302 = 900

Utiliser d'abord l'instruction echo avec des apostrophes puis avec des guillemets.

Remarque : Pour mettre du texte en exposant en XHTML, il est possible d'utiliser la balise sup.

Créer ensuite une fonction prenant en paramètre un nombre n et retournant la chaîne de caractères  $n^2 = m$ . Utiliser cette fonction pour produire la même page Web.

#### Exercice 4 : Table de multiplication\*

Créer un script PHP permettant d'afficher la table de multiplication donnée par la figure 3.3.

	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Figure 3.3 – Table de multiplication

L'opérateur modulo (symbole %) permet de donner le reste de la division entière. Pour déterminer

si une ligne est paire ou impaire, il suffit alors de regarder la valeur du numéro de ligne modulo 2.

#### Exercice 5 : Initiation aux tableaux

Créer un tableau ayant pour valeur les noms des douze mois de l'année. Quelles sont alors les

clés ? Parcourir ensuite le tableau pour afficher les mois de l'année sous forme de liste ordonnée.

Créer un tableau associant à chaque mois de l'année le nombre de jours du mois. (On supposera

que l'année n'est pas bissextile.) Quelles sont alors les clés ? Les valeurs ? Afficher, sous forme de

tableau, le nombre de jours de chaque mois.

#### Exercice 6 : Tableau à deux dimensions

On définit le tableau suivant :

```
1 $personnes = array(  
2 'mdupond'=> array('prenom' => 'Martin', 'nom' => 'Dupond',  
'age' => 25, 'ville' => 'Paris'),  
3 'jm'=> array('prenom' => 'Jean', 'nom' => 'Martin', 'age' =>  
20, 'ville' => 'Villetaneuse'),  
4 'toto'=> array('prenom' => 'Tom', 'nom' => 'Tonge', 'age' =>  
18, 'ville' => 'Epinay'),  
5 'arn'=> array('prenom' => 'Arnaud', 'nom' => 'Dupond', 'age'  
=> 33, 'ville' => 'Paris'),  
6 'email'=> array('prenom'=>'Emilie', 'nom'=>'Ailta',  
'age'=>46, 'ville'=>'Villetaneuse'),  
7 'dask' => array('prenom'=>'Damien',  
'nom'=>'Askier', 'age'=>7, 'ville'=>'Villetaneuse')  
8 );
```

Question 6.1 : Quelles sont les clés du tableau \$personnes et leur type ? De quel type sont les

valeurs de ce tableau ? Quelle est la valeur associée à 'toto' ?

Question 6.2 : Comment accéder à la valeur 33 dans le tableau ? À la valeur 'Epinay' ? Au tableau contenant les valeurs 'Damien', 'Askier', 7, 'Villetaneuse' ?

Question 6.3 : Écrire une fonction permettant d'afficher le tableau dans son ensemble.

Ajouter

une première ligne contenant les clés 'prenom', 'nom', 'age' et 'ville'. Ajouter ensuite un fichier CSS afin d'obtenir le tableau donné par la figure 3.4.

prenom	nom	age	ville
Martin	Dupond	25	Paris
Jean	Martin	20	Villetaneuse
Tom	Tonge	18	Epinay
Arnaud	Dupond	33	Paris
Emilie	Ailta	46	Villetaneuse
Damien	Askier	7	Villetaneuse

Figure 3.4 – Tableau d'informations de personnes

Question 6.4 : Écrire une fonction permettant d'afficher sous forme de tableau (en utilisant toujours le CSS), les informations des personnes habitant dans une ville donnée en paramètre.

Par exemple, si la fonction est appelée avec le tableau \$personnes défini précédemment et la ville

'Epinay', le tableau affiché doit alors contenir uniquement la ligne relative à toto.

## Exercice 7 : Paramètres dans l'url

Question 7.1 : Appeler une page PHP en passant dans l'url un paramètre de nom pseudo et ayant pour valeur un des pseudonymes du tableau \$personnes. Faire en sorte que la page affiche

le pseudo et les informations associées contenues dans le tableau \$personnes. Améliorer le script

pour que ce dernier affiche Désolé, votre pseudonyme n'apparaît pas dans la liste si le pseudonyme n'est pas une clé du tableau \$personnes.

Question 7.2 : Créer un formulaire permettant à l'utilisateur de saisir le pseudonyme à rechercher

afin de faciliter la saisie pour l'utilisateur. Mettre ensuite directement le formulaire dans le script

PHP créé précédemment afin de pouvoir effectuer facilement plusieurs recherches. Faire en sorte

que le champ de saisie du pseudonyme contienne la dernière valeur saisie.

## PROGRAMMATION PHP

### - Travaux dirigés -

#### 1. Principales balises HTML

1.1. Créer 2 pages HTML reproduisant les tableaux suivants :

Numéro	Nom
1	Dupont, Charles
2	Dubois, Jules

Liste des étudiants	
Numéro	Nom
1	Dupont, Charles
2	Dubois, Jules

*Rappel* : les balises HTML nécessaires pour commencer un tableau, une ligne de tableau et une cellule sont respectivement `<table>`, `<tr>` et `<td>`.

1.2. Créer des pages HTML reproduisant les formulaires suivants (les boutons sont inactifs pour le moment) :

##### CONVERSION pouces→cm

Saisissez une valeur en pouces (inches) :

Convertir en cm

*conversion1.html*

##### CONVERSIONS multiples

Valeur à convertir :

Type de conversion :

pouces→cm  
 cm→pouces  
 Celsius→Fahrenheit  
 Fahrenheit→Celsius

Convertir

*conversion2.html*

##### CONVERSIONS multiples

Valeur à convertir :

Type de conversion :

pouces→cm  
 cm→pouces  
 Celsius→Fahrenheit  
 Fahrenheit→Celsius

Convertir

*conversion3.html*

*Rappel* : la balise HTML pour commencer un formulaire est `<form>`, les éléments de formulaires sont ici créés avec `<input type="text">` et `<input type="submit">` pour les boutons d'envoi. La liste déroulante est créée avec la balise `<select>`, ses items avec `<option>`.

Le caractère « flèche à droite » (→) est créé grâce au code `&rarr;`;

1.3. Créer une page HTML *covoiturage.htm* reproduisant le formulaire ci-contre. Dans cet exemple, les utilisateurs sont invités à saisir quelques informations dans un but de covoiturage.

*Rappel* : Un bouton radio est créé avec `<input type="radio">`, et l'exclusion mutuelle entre les 2 boutons radios est réalisée en leur donnant mêmes noms.

Le caractère « plus ou moins » (±) est créé grâce au code `&plusmn;`

##### Renseignements généraux

Nom

Prénom

Tolérance

±5 min

±15 min

>15 min

Villes dés

Orchies

Covoiturage fumeur ☒ Oui ☐ Non

Envoyer

## 2. Traitement de formulaires HTML avec PHP

- 2.1. Modifier le formulaire *conversion1.html* pour qu'un script de traitement nommé *conversion1\_tr.php* soit appelé lorsque l'utilisateur clique sur le bouton « Convertir en cm ». La méthode de transmission des données sera GET dans un premier temps.
- 2.2. Écrire le script de traitement *conversion1\_tr.php* qui se contente d'afficher la valeur saisie dans le formulaire *conversion1.html*. Tester ce script avec plusieurs saisies de valeurs en pouces. Noter que la valeur est transmise en paramètre à l'URL du script (dans la barre d'adresse du navigateur).
- 2.3. Compléter *conversion1\_tr.php* pour qu'il affiche cette fois le résultat de la conversion en centimètres de la valeur saisie dans *conversion1.html*. Exemple d'affichage :  
 Résultat de la conversion : 2 pouces = 5.08 centimètres.
- 2.4. Modifier directement la valeur du paramètre dans l'URL (sans revenir au formulaire).  
 Quel est le lien HTML permettant d'appeler la page correspondante ?  
 Conclure quant aux avantages et inconvénients de la méthode GET.
- 2.5. Utiliser maintenant la méthode POST pour transmettre les données du formulaire *conversion1.html* à son script de traitement *conversion1\_tr.php*.  
 Quelles différences constatez-vous par rapport à l'utilisation de GET ?  
 Conclure quant aux avantages et inconvénients de la méthode POST.

## 3. Structures de contrôle

- 3.1. Affiner le script de traitement *conversion1\_tr.php* pour afficher un message utilisant correctement le pluriel. Exemples :  
 Résultat de la conversion : 0.25 pouce = 0.635 centimètre.  
 Résultat de la conversion : 1.25 pouce = 3.175 centimètres.  
 Résultat de la conversion : 2.25 pouces = 5.715 centimètres.
- 3.2. Écrire le script *conversion3\_tr.php* qui traite le formulaire *conversion3.html* en affichant le résultat de la conversion demandée. On rappelle ici que 1 pouce = 2,54 cm et qu'une température en degrés Celsius  $^{\circ}\text{C}$  s'exprime en degré Fahrenheit par  $^{\circ}\text{F} = ^{\circ}\text{C} \times 9/5 + 32$ . Exemples pour une saisie de 2.25 :  
 Résultat de la conversion : 2.25 pouces = 5.715 centimètres.  
 Résultat de la conversion : 2.25 centimètres = 0.88583 pouce.  
 Résultat de la conversion : 2.25 Celsius = 36.05 Fahrenheit.  
 Résultat de la conversion : 2.25 Fahrenheit = -16.5278 Celsius.
- 3.3. Écrire un script *cm2inch.php* qui affiche la table de conversion des centimètres (de 1 à 5) en pouces sous forme textuelle (cf. ci-dessous).
- 3.4. Compléter *cm2inch.php* en *cm2inch\_tab.php* pour que l'affichage se fasse sous forme de tableau HTML.

### Conversion cm→pouces

1 pouce = 2.54 centimètres  
 2 pouces = 5.08 centimètres  
 3 pouces = 7.62 centimètres  
 4 pouces = 10.16 centimètres  
 5 pouces = 12.7 centimètres

*cm2inch.php*

### Conversion cm→pouces

1 pouce	=	2.54 centimètres
2 pouces	=	5.08 centimètres
3 pouces	=	7.62 centimètres
4 pouces	=	10.16 centimètres
5 pouces	=	12.70 centimètres

*cm2inch\_tab.php*

#### 4. Tableaux

- 4.1. Écrire le script *covoiturage\_tr.php* qui traite les données du formulaire *covoiturage.html* en affichant simplement les valeurs saisies, grâce à la fonction `print_r()`. La méthode de transmission des données doit être POST.

Interpréter le résultat obtenu.

- 4.2. Réécrire ce script de façon à produire exactement le même résultat mais sans utiliser `print_r`. Pour cela, on parcourra le tableau `$_POST` grâce à une boucle `foreach`.

Comment les villes desservies sont-elles transmises en paramètre et quel problème pose leur affichage ?

- 4.3. Modifier ce script pour afficher correctement les villes desservies.
- 4.4. Modifier ce script pour afficher de manière personnalisée la liste des valeurs saisies (notamment les villes choisies), par exemple sous la forme :

```

nom -> Losson
prenom -> Olivier
tolerance -> 15
villes -> (1,3)
fumeur -> Non
envoyer ->
    
```

Champ	Type	Attributs	Extra
<b>num_etu</b>	int(10)	UNSIGNED	auto_increment
<b>nom_etu</b>	varchar(50)		
<b>date_naiss</b>	date		
<b>sexe</b>	char(1)		

#### 5. Pages dynamiques et Bases de données

La base utilisée dans cette partie est constituée d'une seule table *etudiants* dont la structure MySQL est donnée ci-contre.

- 5.1. Écrire un script *afficher\_noms\_etu.php* qui affiche simplement les noms de tous les étudiants :

```

Dupont, Charles
Dubois, Jules
Favier, Isabelle
...
    
```

- 5.2. Écrire un script *afficher\_tab\_etu.php* qui affiche toutes les données de la table *etudiants* sous forme de tableau HTML :

N°	Nom étudiant	Date de naissance	Sexe
1	Dupont, Charles	1991-03-18	M
2	Dubois, Jules	1992-02-11	M
3	Favier, Isabelle	1989-02-02	F
4	Gros, Sophie	1990-06-21	F

- 5.3. *Rappel* : la lecture d'une ligne de résultat d'une requête se fait avec la fonction `mysql_fetch_row()`, `mysql_fetch_array()`, `mysql_fetch_object()` ou `mysql_fetch_assoc()`.

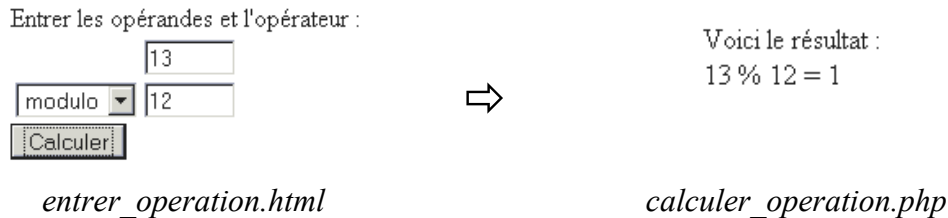
- 5.4. Modifier le script précédent en *afficher\_etu\_annee.php* pour permettre d'afficher uniquement les étudiants nés dans une année donnée. Cette année sera fournie dans l'URL de la page, comme si elle avait été transmise en utilisant la méthode GET. Exemple de *afficher\_etu\_annee.php?annee=1990* :

N°	Nom étudiant	Date de naissance	Sexe
4	Gros, Sophie	1990-06-21	F
6	Humbert, Odile	1990-03-19	F
8	Martin, Julie	1990-04-25	F



## 6. Exercices supplémentaires

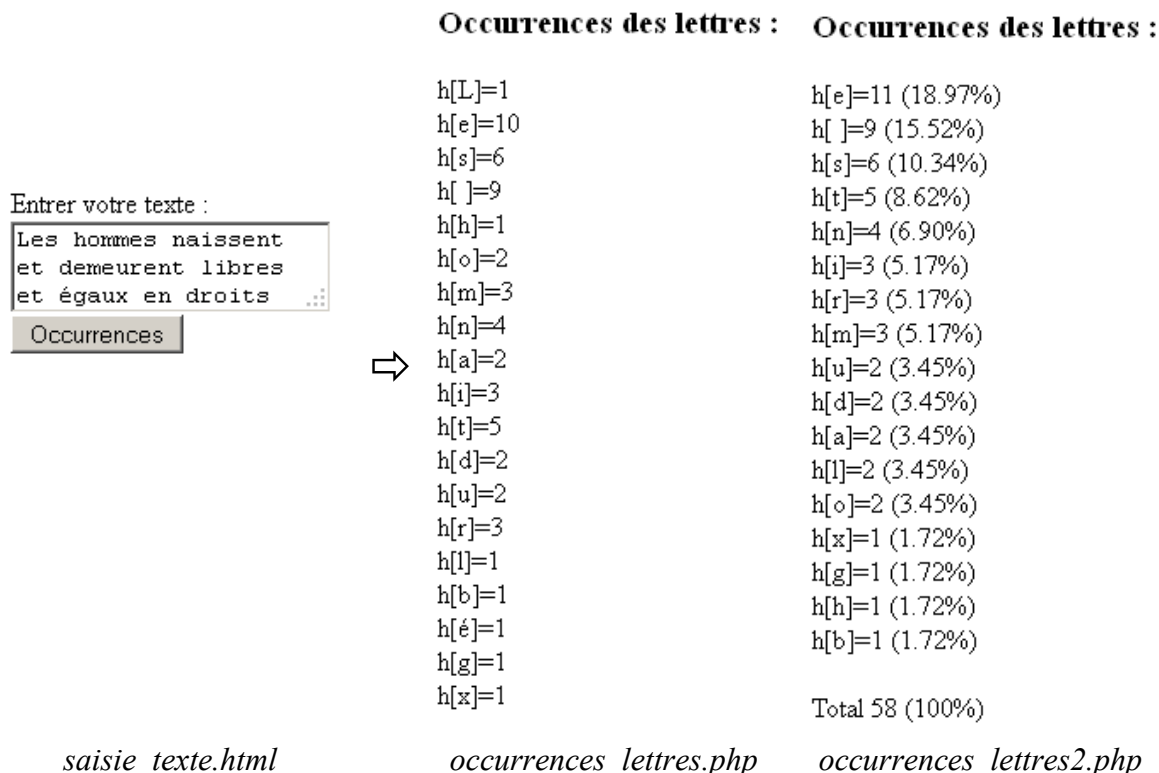
- 6.1. Écrire un formulaire *entrer\_operation.html* et son script de traitement *calculer\_operation.php*, qui permettent de calculer et d'afficher le résultat d'une opération. Les opérations possibles sont : addition, soustraction, multiplication, division et modulo.



- 6.2. Écrire un script *histogramme.php* qui calcule et affiche l'histogramme d'un tableau de valeurs (nommé *tab*) transmis en paramètre.

Comment *tab* est-il transmis au script et comment est-il lu par ce script ?

- 6.3. S'inspirer du script précédent pour écrire un script *occurrences\_lettres.php* qui affiche les occurrences de chaque lettre présente dans un texte saisi par l'utilisateur dans un formulaire *saisie\_texte.html*.



- 6.4. Améliorer le script précédent en *occurrences\_lettres2.php* afin d'ignorer les accents et les majuscules, ainsi que d'afficher les occurrences de lettres dans l'ordre décroissant et les pourcentages correspondants.

**Remarque :** Il est possible d'ignorer les lettres accentuées et les majuscules grâce aux instructions :

```
$replace = "àáâãäåæçèéêëìíîïðñòóôõöùüý";
$by = "aaaaaeieiiioooouuuu";
$texte = strtolower(strtr($texte, $replace, $by));
```