

Travaux dirigés & pratiques : PHP & JS

Table des matières

1	TD PHP	2
1.1	TD PHP Bases (©J. Engels)	3
1.2	TD PHP Structures de contrôle (©J. Engels)	4
1.3	TD PHP Chaînes de caractères (©J. Engels)	5
1.4	TD PHP Tableaux (©J. Engels)	7
1.5	TD PHP Formulaires (©J. Engels)	10
1.6	TD PHP Fonctions (©J. Engels)	11
2	TP PHP	12
2.1	TP PHP 0 - Environnement de travail	13
2.2	TP PHP 1 - Premières pages	15
2.3	TP PHP 2 - Génération HTML	17
2.4	TP PHP 3 - Manipulation de tableaux	19
2.5	TP PHP 4 - Formulaires	21
2.6	TP PHP 5 - Fonctions, formulaires	22
2.7	TP PHP 6 - Création de fichiers	24
2.8	TP PHP 7 - Conversion JSON	25

1 TD PHP

1.1 TD PHP Bases (©J. Engels)

Exercice 1. Parmi les variables suivantes, lesquelles ont un nom valide :

mavar, \$mavar, \$var5, \$_mavar, \$_5var, \$__élément1, \$hotel4*

Exercice 2. Donner les valeurs de `$x`, `$y`, `$z` à la fin du script suivant :

```
<?php
$x="PostgreSQL";
$y="MySQL";
$z=&$x;
$x="PHP 5";
$y=&$x;
?>
```

Exercice 3. Lire les valeurs des variables du script de l'exercice 2 à l'aide du tableau `$GLOBALS`.

Exercice 4. Déterminer le numéro de version de PHP, le nom du système d'exploitation de votre serveur ainsi que la langue du navigateur du poste client.

Exercice 5. Donner la valeur de chacune des variables pendant et à la fin du script suivant et vérifier l'évolution du type de ces variables :

```
$x="PHP7";
$a[0]=&$x;
$y=" 7e version de PHP";
$z=$y*10;
$x.=$y;
$y*=$z;
$a[0]="MySQL";
```

Exercice 6. Donner la valeur de chacune des variables à la fin du script :

```
$x="7 personnes";
$y=(integer) $x;
$x="9E3";
$z=(double) $x;
```

Exercice 7. Donner la valeur booléenne de chacune des variables :

```
$a="0";
$b="TRUE";
$c=FALSE;
$d=($a OR $b);
$e=($a AND $c);
$f=($a XOR $b);
```

1.2 TD PHP Structures de contrôle (©J. Engels)

Exercice 1. Rédiger une expression conditionnelle pour tester si un nombre est à la fois un multiple de 3 et de 5.

Exercice 2. Écrire une expression conditionnelle utilisant les variables `$age` et `$sexe` dans une instruction `if` pour sélectionner une personne de sexe féminin dont l'âge est compris entre 21 et 40 ans et afficher un message de bienvenue approprié.

Exercice 3. Effectuer une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée d'un nombre pair suivi de deux nombres impairs. Exemple de résultat obtenu :

- 194,285,494
- 435,759,162
- 237,292,768
- 366,533,397

Résultat obtenu en 4 coups.

Exercice 4. Créer et afficher des numéros d'immatriculation automobile (pour Paris, par exemple) en commençant au numéro 100 PHP 75. Si on réalise le script complet, il affiche plusieurs millions de numéros de 100 PHP 75 à 999 ZZZ 75. L'exécution est donc très longue et risque de bloquer le serveur. Pour effectuer un test, les valeurs des chiffres seront limitées entre 100 et 120.

Effectuer ensuite la même procédure en mettant en réserve les numéros dont le premier groupe de chiffres est un multiple de 100. Stocker ces numéros particuliers dans un tableau. Pour ne conserver que les nombres multiples de 100 nous remplaçons l'instruction `echo` par le stockage des numéros dans un tableau. Il est affiché à la fin de toutes les boucles à l'aide de la fonction `print_r()`. On constate alors qu'il contient déjà 65150 éléments !

Exercice 5. Choisir un nombre de trois chiffres. Effectuer ensuite des tirages aléatoires et compter le nombre de tirages nécessaire pour obtenir le nombre initial. Arrêter les tirages et afficher le nombre de coups réalisés. Réaliser ce script d'abord avec l'instruction `while` puis avec l'instruction `for`. Notez qu'il est rare d'obtenir le nombre cherché en moins de 100 tirages.

Exercice 6. Créer un tableau dont les indices varient de 11 à 36 et dont les valeurs sont des lettres de A à Z. Lire ensuite ce tableau avec une boucle `for` puis une boucle `foreach` et afficher les indices et les valeurs (la fonction `chr(n)` retourne le caractère dont le code ASCII vaut `n`).

Exercice 7. Utiliser une boucle `while` pour déterminer le premier entier obtenu par tirage aléatoire qui soit un multiple d'un nombre donné. Écrire la variante utilisant la boucle `do...while`.

Exercice 8. Rechercher le **plus grand commun diviseur** de deux variables initialisées dans votre script. Gérer au moyen d'une exception le cas où l'une des variables n'est pas entière.

1.3 TD PHP Chaînes de caractères (©J. Engels)

Exercice 1. Transformez une chaîne écrite dans des casses différentes afin que chaque mot ait une initiale en majuscule. Appliquez à la chaîne :

```
"TransFOrmeZ unE ChaîNE éCRITe dans des cASses diFFérenTes  
afin qUe chaQue MOT ait une inITiale en MAJUSCULE"
```

Le résultat est le suivant :

```
Transformez Une Chaîne Écrite Dans Des Casses Différentes  
Afin Que Chaque Mot Ait Une Initiale En Majuscule
```

Exercice 2. En utilisant la fonction `strlen()` écrivez une boucle qui affiche chaque lettre de la chaîne "PHP MySQL" sur une ligne différente.

Exercice 3. Formatez l’affichage d’une suite de chaînes contenant des noms et prénoms en respectant les critères suivants : un prénom et un nom par ligne affichés sur 20 caractères ; toutes les initiales des mots doivent se superposer verticalement. Pour obtenir de bons résultats il est préférable d’utiliser une police à espacement fixe en incluant le texte de chaque ligne dans un élément HTML `<tt>`. Le résultat envisagé est le suivant :

```
Azerky_____ Sophia_____  
Bazertudoh_____ Jean-Michel_____
```

Utilisez ensuite la fonction `vsprintf()` et passer les chaînes à afficher dans un tableau multidimensionnel.

Exercice 4. Utilisez les fonctions adéquates afin que la chaîne `<form action="script.php">` soit affichée telle quelle et non comme du code HTML. Le code HTML de l’affichage est le suivant :

```
&lt;form action=&quot;script.php&quot;&gt;
```

Exercice 5. À partir de deux chaînes quelconques contenues dans des variables, effectuez une comparaison entre elles pour pouvoir les afficher en ordre alphabétique naturel. Utilisez la fonction `strtolower()` avant d’opérer la comparaison, sinon tous les caractères de A à Z sont avant les caractères a à z.

Exercice 6. Effectuez une censure sur des textes (par exemple sur "Zut je me suis trompé") en n’affichant pas ceux qui contiennent le mot `zut`.

Exercice 7. Créez une fonction de validation d’une adresse HTTP utilisant `preg_match()`. Les adresses doivent se conformer au modèle suivant : (1) Commencer par « `www` » ; (2) Suivi par des caractères alphanumériques puis éventuellement un point ou un tiret suivi d’un deuxième groupe de caractères alphanumériques ; (3) Se terminer par un point suivi de l’extension qui peut avoir de 2 à 4 caractères alphanumériques. Le modèle est insensible à la casse. Exemples de résultats à obtenir :

```
www.univ-angers.fr est valide  
www.info.ua.fr est valide  
www.49\_info.ua.fr est invalide  
WWW.info.ua.fr est valide  
www*info.ua.fr est invalide  
www.info;ua.fr est invalide  
www.info.ua.france est invalide  
www.info.ua.ubl.fr est invalide
```

Exercice 8. Créez une expression régulière pour valider un âge inférieur à 100 ans. Le résultat obtenu sur la chaîne "84" est le suivant :

```
84 est un âge valide
```

Exercice 9. Dans la chaîne "PHP 5 \n est meilleur \n que ASP\n et JSP \n réunis", remplacez les caractères \n par
 en utilisant deux méthodes différentes (une fonction ou une expression régulière) :

```
PHP
est meilleur
que ASP
et JSP
réunis
```

1.4 TD PHP Tableaux (©J. Engels)

Exercice 1. Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne. Afficher le tableau avec la fonction `print_r()`.

Exemple :

```
Array (
  [Dupont] => Array ( [0] => Paul [1] => Paris [2] => 27 )
  [Schmoll] => Array ( [0] => Kirk [1] => Berlin [2] => 35 )
  [Smith] => Array ( [0] => Stan [1] => Londres [2] => 45 )
)
```

Exercice 2. Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux associatifs dont les clés sont le prénom, la ville de résidence et l'âge de la personne avec une série de valeurs associées.

Exemple :

```
Array (
  [Dupont] => Array ( [prenom] => Paul [ville] => Paris [age] => 27 )
  [Schmoll] => Array ( [prenom] => Kirk [ville] => Berlin [age] => 35 )
  [Smith] => Array ( [prenom] => Stan [ville] => Londres [age] => 45 )
)
```

Exercice 3. Utilisez une boucle `foreach` pour lire les tableaux des exercices 1 et 2.

Lecture du tableau de l'exercice 1 :

```
Elément Dupont :
élément 0 :Paul
élément 1 :Paris
élément 2 :27
Elément Schmoll :
élément 0 :Kirk
élément 1 :Berlin
élément 2 :35
Elément Smith :
élément 0 :Stan
élément 1 :Londres
élément 2 :45
```

Lecture du tableau de l'exercice 2 :

```
Element Dupont :
prenom :Paul
ville :Paris
age :27
Element Schmoll :
prenom :Kirk
ville :Berlin
age :35
Element Smith :
prenom :Stan
ville :Londres
age :45
```

Exercice 4. Utilisez une boucle `while` pour lire les tableaux des exercices 1 et 2 avec la fonction `each()`.
Lecture du tableau de l'exercice 1 :

```
Personne: Dupont
clé 0 valeur Paul
clé 1 valeur Paris
clé 2 valeur 27
Personne: Schmoll
clé 0 valeur Kirk
clé 1 valeur Berlin
clé 2 valeur 35
Personne: Smith
clé 0 valeur Stan
clé 1 valeur Londres
clé 2 valeur 45
```

Lecture du tableau de l'exercice 2 :

```
Personne: Dupont
prenom:Paul
ville:Paris
age:27
Personne: Schmoll
prenom:Kirk
ville:Berlin
age:35
Personne: Smith
prenom:Stan
ville:Londres
age:45
```

Exercice 5. Créez un tableau contenant une liste d'adresses de sites recommandés :

```
$tab=array(
"PHP"=>"http://www.php.net",
"MySQL"=>"http://www.mysql.org",
"SQLite"=>"http://www.sqlite.org");
```

Puis créez un lien aléatoire vers le premier site de la liste après avoir trié le tableau en ordre aléatoire en utilisant la fonction `array_rand()` qui retourne la clé de l'élément pris au hasard (et non pas `shuffle()`).

Exercice 6. Créez un tableau d'entiers variant de 1 à 63, puis à partir de celui ci un autre tableau de nombres variant de 0 à 6.3. Créez ensuite un tableau associatif dont les clés `X` varient de 0 à 6.3 et dont les valeurs sont `sin(X)`. Affichez le tableau de valeurs dans un tableau HTML.

Le résultat affiché est le tableau HTML suivant :

```
Tableau de valeurs de la fonction sinus
X sin( X )
0 0
0.1 0.0998334166468
0.2 0.198669330795
0.3 0.295520206661
...
5.9 -0.37387666483
6 -0.279415498199
6.1 -0.182162504272
6.2 -0.0830894028175
6.3 0.0168139004843
```


Exercice 7. Créez un tableau contenant une liste d'adresses e-mail. Extrayez le nom de serveur de ces données, puis réalisez des statistiques sur les occurrences de chaque fournisseur d'accès.

Exemple de résultat :

Fournisseur d'accès : free.com = 16.67 %

Fournisseur d'accès : fiscali.fr = 50 %

Fournisseur d'accès : waladoo.fr = 33.33 %

1.5 TD PHP Formulaires (©J. Engels)

Exercice 1. Créer un formulaire comprenant un groupe de champs ayant pour titre "Adresse client". Le groupe doit permettre la saisie du nom, du prénom, de l'adresse, de la ville et du code postal. Les données sont ensuite traitées par un fichier PHP séparé récupérant les données et les affichant dans un tableau HTML. Utiliser la fonction `stripslashes()` pour supprimer le caractère d'échappement `\` qui est automatiquement ajouté dans les chaînes saisies.

Exemple de résultat obtenu :

```
Confirmation de vos coordonnées  
nom : Lagarde  
prenom : Richard  
adresse : 556 Rue de l'Odéon  
ville : Paris  
code : 75006
```

Exercice 2. Améliorer le script précédent en vérifiant l'existence des données et en affichant une boîte d'alerte JavaScript si l'une des données est manquante.

Exercice 3. Le fichier suivant peut-il être enregistré avec l'extension `.php` ou `.htm` ? Où se fait le traitement des données ?

```
<!DOCTYPE html>  
<html>  
<head>  
<title> Insertion des données </title>  
</head>  
<body>  
<form method="post" action="ajout.php" >  
//Suite du formulaire  
</form>  
</body>  
</html>
```

Exercice 4. Comment faire pour que les données soient traitées par le même fichier que celui qui contient le formulaire ? Proposer deux solutions.

1.6 TD PHP Fonctions (©J. Engels)

Exercice 1. Créez une fonction PHP qui affiche une boîte d'alerte à partir de la fonction JavaScript dont la syntaxe est `alert("chaîne_de_caractères")`. Cette fonction peut être appelée avec comme paramètre le texte du message à afficher. Elle est particulièrement utile pour afficher des messages d'erreur de manière élégante, sans que ces derniers restent écrits dans la page. La fonction retourne la valeur `TRUE`. Ceci n'est pas obligatoire mais peut permettre un contrôle d'exécution.

Exercice 2. Écrivez une fonction de lecture de tableaux multidimensionnels en vous inspirant de l'exemple 7.3. L'affichage se fait sous forme de tableau HTML dont les titres sont les clés des tableaux.

Exercice 3. Écrivez une fonction qui retourne la somme de la série de terme général $un = x^{2n} + 1/n!$. Les paramètres de la fonction sont `n` pour le nombre d'itérations et `d` pour le nombre de décimales affichées pour le résultat.

Exercice 4. Écrivez une fonction dont le paramètre passé par référence est un tableau de chaînes de caractères et qui transforme chacun des éléments du tableau de manière que le premier caractère soit en majuscule et les autres en minuscules, quelle que soit la casse initiale des éléments, même si elle est mixte.

Exercice 5. À partir de la fonction sinus de PHP, écrivez une fonction qui donne le sinus d'un angle donné en radian, en degré ou en grade. Les paramètres sont la mesure de l'angle et l'unité est symbolisée par une lettre. Le deuxième paramètre doit avoir une valeur par défaut correspondant aux radians.

Exercice 6. Réaliser la programmation des coefficients du binôme (ou triangle de Pascal). Pour mémoire, il s'agit de la suite suivante :

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
...
```

En remarquant que la première colonne et la diagonale valent toujours 1 et que chaque autre élément est égal à la somme de celui qui est au dessus et de celui qui se trouve sur la diagonale gauche (exemples $3=2+1$ ou $6=3+3$).

Exercice 7. Créez une fonction anonyme qui retourne la division entière de deux nombres.

2 TP PHP

2.1 TP PHP 0 - Environnement de travail

Vous travaillerez sur les postes AIO avec un serveur web **LAMP** (Linux Apache MySQL PHP) que vous lancerez à chaque séance dans un conteneur **Docker**. Ce serveur hébergera, de manière temporaire, vos ressources web (fichiers HTML/CSS, images, etc), vos scripts PHP et vos données sous MySQL.

Docker

- Assurez-vous qu'aucune instance de Firefox n'est en cours d'exécution avant de lancer le conteneur Docker.
- Pour lancer un conteneur Docker, exécutez la commande `_php7` dans un terminal et attendez que le terminal vous rende la main après avoir affiché les deux comptes d'accès `root` et `phpmyadmin` à l'application web phpMyAdmin (voir plus bas).
- Pour vérifier que le serveur Apache fonctionne, lancez Firefox et chargez l'URL ¹ `http://localhost`. Le serveur doit afficher une page listant le contenu de son répertoire racine comprenant le sous-répertoire `Mes_projets_web` et le script `phpinfo.php`
- Pour vérifier que le module PHP fonctionne sous Apache, chargez `http://localhost/phpinfo.php`. Le script `phpinfo.php` doit générer une page affichant les caractéristiques de l'installation Apache/PHP.
- Pour arrêter et supprimer le conteneur Docker, exécutez la commande `exit`. Si vous supprimez la console de lancement d'un conteneur sans avoir arrêté le conteneur au préalable avec `exit`, il vous faudra exécuter la commande `_clean` avant de pouvoir relancer un nouveau conteneur.

Répertoire de travail & Sauvegarde

Quel que soit l'éditeur utilisé, vous devrez placer tous vos fichiers et répertoires (PHP, JS, HTML, ...) dans le répertoire `Mes_projets_web` situé dans votre répertoire personnel, répertoire auquel accède le serveur LAMP pour servir vos pages web :

- Chemin local : `$HOME/Mes_projets_web`
- Chemin dans le conteneur : `/home/php_dev/Mes_projets_web`
- URL : `http://localhost/Mes_projets_web/`

Attention : vos fichiers seront détruits une fois le conteneur arrêté, et notamment à chaque déconnexion en fin de séance.

Veillez donc à les sauvegarder, en utilisant par exemple

- `scp` ou FileZilla pour les transférer dans votre répertoire personnel sur le serveur `starwars` du département informatique ;
- ou `git` pour cloner un dépôt git distant dans `Mes_projets_web` et y sauvegarder vos modifications en fin de séance.

1. Le port 80 de localhost est redirigé (par configuration) vers le port 80 du conteneur sur lequel écoute le serveur Apache donc inutile de le consigner dans l'URL.

Xdebug

PHP est configuré avec **Xdebug** qui permet de relayer avertissements et erreurs PHP en HTML à l'utilisateur/développeur. Testez Xdebug en chargeant le script suivant :

```
<?php
echo "rapport d'erreurs préconfiguré : ".error_reporting()."<br/>";
error_reporting(~E_WARNING);
$a = 2 / 0;
echo 'division par 0<br/>';
error_reporting(~E_NOTICE);
echo $b;
echo 'variable indéfinie<br/>';
error_reporting(~E_ALL);
require "nowhere.file";
echo "ne s'affichera pas<br/>";
?>
```

Commentez ensuite les différents appels à la fonction **error_reporting** utilisant les **constantes prédéfinies** **E_WARNING**, **E_NOTICE** et **E_ALL** et analysez le résultat. Effectuez les mêmes opérations en utilisant PHP en ligne de commande.

phpMyAdmin

phpMyAdmin est une interface web aux SGBD MySQL écrite en PHP. phpMyAdmin est préinstallée dans le conteneur Docker et accessible à l'adresse : <http://localhost/phpmyadmin>.² Vous pourrez vous y connecter avec le compte **root** et le mot de passe qui vous est communiqué par le shell de lancement du conteneur.^{3 4} Toute base de données MySQL que vous créerez devra être exportée sous forme de script MySQL afin d'être sauvegardée et réimportée la session suivante. Export et import se font simplement à partir de l'interface phpMyAdmin. Alternativement, vous pouvez utiliser les commandes suivantes :

- Pour exporter une base de données appelée **dbExample** :
`mysqldump -u root -p dbExample > dbExample.sql`
- Pour importer la base dans le SGBD MySQL :
`mysql -u root -p < dbExample.sql`

2. Si vous rencontrez des problèmes d'authentification avec le compte **root** sur **phpMyAdmin**, arrêtez Firefox, relancez le conteneur puis Firefox de nouveau.

3. Attention, les mots de passe sont renouvelés d'une session à l'autre.

4. Ne pas utiliser le compte **phpmyadmin**.

2.2 TP PHP 1 - Premières pages

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Créez une page web de type “Hello world!” à l'aide d'un script PHP. Le fichier HTML généré doit importer un fichier CSS qui appliquera une couleur de fond de votre choix. L'onglet de la page web affiché par le navigateur sera dénommé **TP-PHP-1.1 - Hello**.

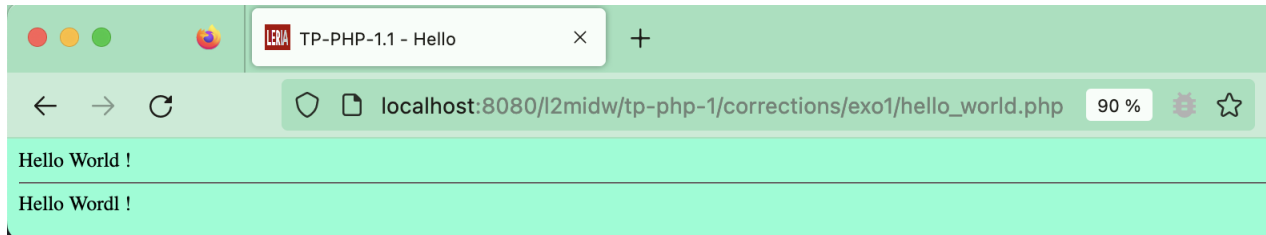


FIGURE 1 – Première page

Exercice 2. Implantez le fichier `main.php` dans un sous-répertoire pour produire ce type de page :

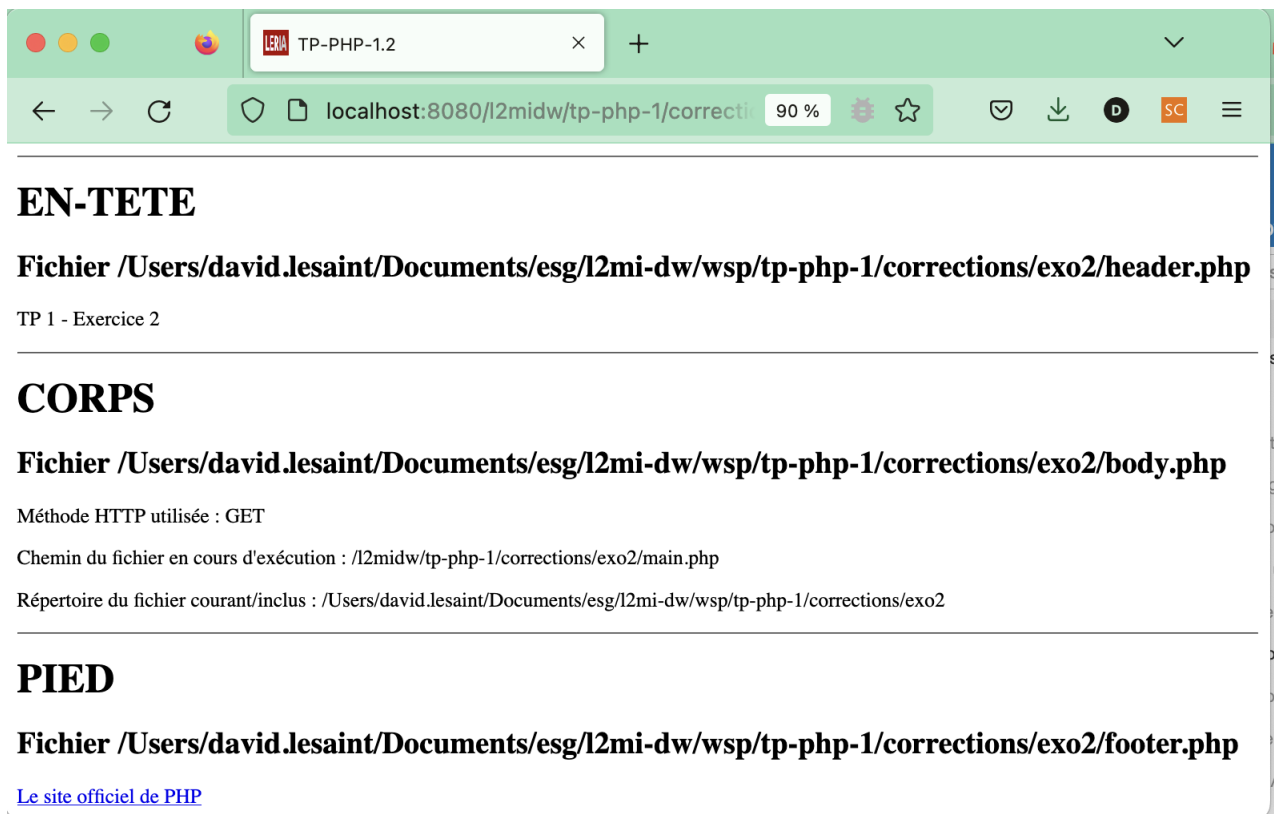


FIGURE 2 – Page web assemblant en-tête, corps et pied de page

Le fichier `main.php` assemble trois fragments du document HTML produits respectivement par les fichiers `header.php`, `body.php` et `footer.php`. Chacun de ces fichiers affiche différentes constantes et variables qui sont initialisées dans `main.inc.php` :

- Une variable dénotant le numéro du TP apparaissant dans le titre et l'en-tête de la page.
- Une variable dénotant le numéro de l'exercice apparaissant dans le titre et l'en-tête de la page.
- Une constante dénotant la méthode HTTP utilisée pour accéder à `main.php` (cf. superglobale `$_SERVER`) et apparaissant dans le corps de la page.
- Une constante dénotant le fichier en cours d'exécution (cf. superglobale `$_SERVER`) et apparaissant dans le corps de la page.
- Une variable dénotant l'URL (`http://www.php.net`) de l'hyperlien du pied de page.
- Une variable dénotant le contenu texte ("`Le site officiel de PHP`") de l'hyperlien du pied de page.

Le nom de chaque fichier inclus doit apparaître dans le fragment HTML qu'il génère (cf. constante magique `FILE`). Les caractères `é` et `°` doivent s'afficher correctement quel que soit l'encodage utilisé dans votre navigateur (à modifier via le menu **Affichage**→**Encodage du texte**).

2.3 TP PHP 2 - Génération HTML

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Le tableau suivant affiche la comparaison par la relation == (opérateur binaire booléen en PHP) des littéraux de l'ensemble {TRUE,FALSE,1,0,-1,"1","0","-1",NULL,[],""}. Par exemple, 1=="1" est vrai (dénnoté par TRUE) alors que 1==[] est faux (dénnoté par la chaîne vide).

==	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	[]	""
TRUE	TRUE		TRUE		TRUE	TRUE		TRUE			
FALSE		TRUE		TRUE			TRUE		TRUE	TRUE	TRUE
1	TRUE		TRUE			TRUE					
0		TRUE		TRUE			TRUE		TRUE		TRUE
-1	TRUE				TRUE			TRUE			
"1"	TRUE		TRUE			TRUE					
"0"		TRUE		TRUE			TRUE				
"-1"	TRUE				TRUE			TRUE			
NULL		TRUE		TRUE					TRUE	TRUE	TRUE
[]		TRUE							TRUE	TRUE	
""		TRUE		TRUE					TRUE		TRUE

FIGURE 3 – Tableau de comparaison avec ==.

Générez ce tableau HTML à l'aide d'un script PHP en créant un tableau associatif des littéraux à tester sur lequel vous itérerez pour générer les en-têtes de lignes et de colonnes et les cellules "résultats".

Exercice 2. Le fichier `tableau.php` est à compléter pour obtenir la page (partiellement) illustrée en Figure 4. Il contient un tableau d'employés `$employees` qu'il s'agit de transformer par script PHP en tableau HTML à 2 colonnes, le bloc apparaissant à droite étant déjà codé en HTML. L'extrait ci-dessous fournit le code HTML attendu pour les 2 premiers employés :

```
<tr class="bassalaire">
  <td title="7 voyelles , 872 euros">Geraldine Meyer</td>
  <td><input type="number" min="0" max="100" value="95"/></td>
</tr>
<tr class="hautsalaire">
  <td title="4 voyelles , 4230 euros">Idona Glenn</td>
  <td><input type="number" min="0" max="100" value="25" disabled/></td>
</tr>
```

Chaque ligne HTML correspond à un employé et doit satisfaire aux contraintes suivantes :

- La ligne est classée `hautsalaire` si le salaire de l'employé est > 5000 et classée `bassalaire` sinon.
- La première cellule contient le nom de l'employé et une infobulle qui affichera au survol de la souris le nombre de voyelles comptabilisées dans le nom de l'employé ainsi que son salaire.
- La seconde cellule contient un champ numérique affichant l'âge de l'employé : ce champ est désactivé si l'âge de l'employé est > 50.

Geraldine Meyer	95	
Idona Glenn	25	
Martena Hyde	70	
Colette Mcmillan	37	
Raya Cook	45	
Warren Hendrix	62	
Lionel Best	53	
Louis Brown	41	
Ginger Wolf	77	
Sade May	65	

Haut salaire > 5000
Bas salaire <= 5000
Désactivé si âge > 50

FIGURE 4 – Tableau d’employés (fichier **tableau.php**).

Exercice 3. L’objectif est ici de calculer des statistiques sur le tableau de données défini dans le fichier **tabEmployees.php** par la variable **\$employees**. Vous ne modifierez que le fichier **statistiques.php**. Pour répondre, n’hésitez pas à utiliser les fonctions PHP de manipulation de tableaux (filtrage, tri, parcours, ...). Les 4 statistiques à calculer dans les variables définies au début du fichier sont les suivantes :

- **\$ageMoyen** : âge moyen de tous les employés présents dans le tableau **\$employees**.
- **\$salaireMoyen** : salaire moyen de tous les employés de plus de 55 ans présents dans le tableau **\$employees**. Le calcul devra être fait sur un tableau filtré au préalable (contenant uniquement les employés de plus de 55 ans).
- **\$tenRichest** : extraire les 10 employés qui ont le meilleur salaire à partir du tableau **\$employees**. Ces 10 employés seront stockés dans **\$tenRichest** au format "Prénom Nom (salaire)". Par exemple : **\$tenRichest** = ["Oren Dudley (9985 €)", "Jerome Flynn (9881 €)", ...]

2.4 TP PHP 3 - Manipulation de tableaux

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Complétez le fichier **voyelles.php** afin de créer un tableau associatif qui comptabilise les occurrences des voyelles **a**, **e**, **i**, **o**, **u** apparaissant dans le fichier texte **Latin-Lipsum.txt**. Afin de récupérer sous forme de chaîne de caractères le contenu de ce fichier, utilisez la fonction PHP **file_get_contents**. L'affichage du tableau en fin de script doit aboutir au résultat suivant (l'ordre des éléments n'a pas d'importance) :

```
Array
(
    [o] => 147
    [e] => 286
    [i] => 274
    [u] => 191
    [a] => 207
)
```

Appuyez-vous sur les fonctions PHP de manipulation de tableaux et de conversion de chaînes, p.ex., **str_split**, **array_filter**, **in_array**, **array_walk**, **array_key_exists**.

Exercice 2. Le fichier **comparaison.php** contient un tableau **\$langues** comptabilisant des voyelles pour 5 textes écrits dans des langues différentes.

On peut comparer et ordonner ces textes en nombre d'occurrences croissant pour une voyelle donnée. Par exemple, **gallois** précède **français** pour la voyelle **o**. En cas d'égalité, on peut utiliser une nouvelle voyelle pour tenter de discriminer 2 textes. Par exemple, **samoan** précède **anglais** pour l'ordre (**o,e**) (248 **o** pour chacun mais 308 **e** pour le premier contre 335 **e** pour le second). Plus généralement, on peut trier ces textes sur la base de n'importe quel ordre total sur les voyelles répertoriées.

Complétez **comparaison.php** pour réaliser ce tri sur la base de l'ordre des voyelles donné par la variable **\$voyelles**. L'affichage attendu est illustrée sur la page suivante. Utilisez la fonction **print_r** pour l'affichage des tableaux et appuyez-vous sur les fonctions PHP de tri et de transformation de tableaux, p.ex., **sort**, **usort**, **array_map**.

```
Tri selon l'ordre o,e,i,u,a:
Array
(
    [0] => gallois
    [1] => français
    [2] => zoulou
    [3] => samoan
    [4] => anglais
)
Tri selon l'ordre a,e,i,o,u:
Array
(
    [0] => anglais
    [1] => français
    [2] => gallois
    [3] => zoulou
    [4] => samoan
)
```

2.5 TP PHP 4 - Formulaire

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Complétez le fichier **calculatrice.php** pour afficher une mini-calculatrice proposant les opérations d'addition, soustraction, division, et exponentiation sur deux entiers. Les arguments saisis doivent rester affichées après calcul et toute soumission de formulaire incomplet doit être géré.

Exemple d'affichage après calcul :

Calculatrice en ligne

Nombre X	10
Nombre Y	3
Résultat	1000
Choisissez !	<input type="button" value="Addition x+y"/> <input type="button" value="Soustraction x-y"/> <input type="button" value="Division x/y"/> <input type="button" value="Puissance x^y"/>

Exercice 2. Créez un formulaire de saisie d'adresse e-mail contenant un champ caché destiné à récupérer le type du navigateur de l'utilisateur. Le code PHP affiche l'adresse mail et le nom du navigateur dans la même page après vérification de l'existence des données.

Exemple de résultat :

Saisissez votre e-mail

mail :

Votre mail et votre navigateur

mail :	a@b.c
navigateur :	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:57.0) Gecko/20100101 Firefox/57.0

2.6 TP PHP 5 - Fonctions, formulaires

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Complétez le corps des fonctions du fichier **form_generation.inc.php** permettant de créer le code HTML de deux types de champs de formulaire :

- les champs texte monoligne,
- les boutons de soumission.

Ajoutez une fonction permettant de générer un bouton radio à partir d'un libellé, du nom du champ et de sa valeur. Développez ensuite deux fonctions supplémentaires permettant de générer les balises d'ouverture et de clôture d'un formulaire HTML avec les attributs requis. Complétez enfin le fichier **form_generation.php** qui utilisera ces fonctions pour créer le formulaire illustré ci-dessous. Ce formulaire enverra les données par méthode HTTP POST au fichier **traiter.php**.

Complétez le formulaire

Votre nom

Votre prénom

Paris ☐

Lyon ☐

Exercice 2. Le fichier **villes.csv** modélise différentes villes au format CSV (le point-virgule fait office de séparateur) en donnant pour chacune son nom, sa région, sa population et son statut de préfecture : "1" s'il s'agit d'une préfecture, "0" sinon. Complétez le fichier **population.php** de sorte :

- qu'il lise ce fichier avec une fonction **importerCSV** que vous développerez,
- puis affiche la page illustrée ci-dessous.

Nom	Région	Population	Préfecture
Nantes	Pays de la Loire	282047	oui
Avignon	Provence-Alpes-Côte d'Azur	89592	oui
Marseille	Provence-Alpes-Côte d'Azur	850602	oui
Toulon	Provence-Alpes-Côte d'Azur	165514	oui
Saumur	Pays de la Loire	27486	non

Utilisez la fonction **fgetcsv** pour lire et décomposer chaque ligne du fichier.

Adaptez ensuite votre fichier **population.php** de sorte que les villes apparaissent selon l'ordre alphabétique inverse de leurs régions, et, pour les villes de la même région, par population décroissante.

Nom	Région	Population	Préfecture
Marseille	Provence-Alpes-Côte d'Azur	850602	oui
Toulon	Provence-Alpes-Côte d'Azur	165514	oui
Avignon	Provence-Alpes-Côte d'Azur	89592	oui
Nantes	Pays de la Loire	282047	oui
Saumur	Pays de la Loire	27486	non

Adaptez enfin votre fichier **population.php** afin d'insérer un hyperlien sur le texte **Nom**. Au chargement de la page web, les villes apparaissent triées selon l'ordre défini précédemment et le tableau apparaît sur fond rouge. Un clic sur l'hyperlien doit inverser l'ordre des villes et appliquer une couleur de fond orange au tableau. Quel mécanisme pourrait être utilisé pour que chaque clic permette d'alterner entre les deux couleurs de fond et les tris ?

Nom	Region	Population	Préfecture
Marseille	Provence-Alpes-Côte d'Azur	850602	oui
Toulon	Provence-Alpes-Côte d'Azur	165514	oui
Avignon	Provence-Alpes-Côte d'Azur	89592	oui
Nantes	Pays de la Loire	282047	oui
Saumur	Pays de la Loire	27486	non

FIGURE 5 – Au chargement

Nom	Region	Population	Préfecture
Saumur	Pays de la Loire	27486	non
Nantes	Pays de la Loire	282047	oui
Avignon	Provence-Alpes-Côte d'Azur	89592	oui
Toulon	Provence-Alpes-Côte d'Azur	165514	oui
Marseille	Provence-Alpes-Côte d'Azur	850602	oui

FIGURE 6 – Après clic sur hyperlien

2.7 TP PHP 6 - Création de fichiers

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Créez un fichier qui enregistre la date de chaque connexion et affiche le log des connexions en retour. Exemple d'affichage après plusieurs connexions :

```
24/Jan 09:01:33
24/Jan 09:01:16
24/Jan 09:01:18
24/Jan 09:01:19
24/Jan 09:01:20
24/Jan 09:01:21
24/Jan 09:01:22
```

Utilisez les fonctions :

- **fopen** (en mode lecture ou ajout), **fclose** et **flock** pour verrouiller/déverrouiller l'accès au fichier.
- **time** pour obtenir le **timestamp UNIX**.
- **date** pour formater l'affichage d'un timestamp, eg. `date("d/M H:m:s",$t)` produit l'affichage ci-dessus pour chaque timestamp `$t`.

Exercice 2. Créez un fichier permettant d'afficher une liste d'étudiants et de s'y inscrire en fournissant nom, prénom et groupe (I1...I6) : Le script créera et mettra à jour un fichier au format CSV conte-

Inscrivez-vous ou affichez la liste des étudiants

Nom :

Prénom :

Groupe :

Otto Didact : vous êtes inscrits dans le groupe I1

Inscrivez-vous ou affichez la liste des étudiants

Nom :

Prénom :

Groupe :

ID	NOM	PRENOM	GROUPE
1	De Ronsard	Pierre	I1
2	Du Bellay	Joachim	I1
3	Hugo	Victor	I2
4	Baudelaire	Charles	I6
5	Rimbaud	Arthur	I5
6	Verlaine	Paul	I4
7	Appolinaire	Guillaume	I2
8	Eluard	Paul	I5
9	Aragon	Louis	I2
10	Prévert	Jacques	I6

nant les étudiants. Utilisez la fonction **fgetcsv** pour lire et décomposer chaque ligne du fichier. Pour chaque étudiant du listing, insérez un hyperlien dans le champ nom permettant de lui envoyer un mail (...href="mailto:X.Y@etud.univ-angers.fr"...). Veillez à ce que l'inscription d'un étudiant de nom **<h1 style="color:red">Trompe</h1>** et prénom Ronald se passe correctement.

2.8 TP PHP 7 - Conversion JSON

Décompressez l'archive déposée sur Moodle pour ce TP. Le dossier résultant contient différents fichiers à réutiliser ou à compléter.

Exercice 1. Le fichier **langages.php** contient 5 tableaux, chacun comptabilisant des voyelles d'un texte écrit dans une langue spécifique. Ces tableaux sont regroupés dans un tableau associatif dont les clés identifient chaque langue. Complétez la fonction **creerFichier(\$f,\$v,\$l)** qui crée un fichier au format CSV, de nom **\$f**, dont la première ligne affiche les voyelles du tableau **\$v**, et qui transcrit les différents tableaux de **\$l**, comme illustré ci-dessous :

```
LANGUE,o,e,i,u,a
anglais,248,355,222,107,198
français,178,435,240,212,201
gallois,152,264,221,107,266
samoan,248,308,334,208,656
zoulou,203,298,311,274,411
```

Cette fonction doit pouvoir être réutilisable avec d'autres jeux de données (autres langues, autres voyelles, ...).

Exercice 2. Transformez le contenu du fichier JSON **country-by-capital-city.json** en un tableau associatif dont la clé de chaque élément correspond au nom d'un pays et sa valeur à la capitale du pays. Le tableau sera trié en ordre alphabétique décroissant des continents. Utilisez la fonction PHP adéquate pour lire des fichiers JSON.

Exercice 3. Le fichier **countries.csv** est au format CSV et utilise le point-virgule comme séparateur de champs. Chaque ligne correspond à un pays et indique son continent suivi de son type de gouvernement. Complétez le fichier **csv.php** pour répondre aux questions qui suivent.

1. Extraire de **countries.csv** les types de gouvernements sous forme de tableau PHP sans se soucier des doublons.
2. Eliminer les doublons du tableau, le trier alphabétiquement, le consigner dans le fichier CSV **governments.csv** (fichier à une seule colonne) et l'afficher avec la fonction PHP **var_export**.