UNIVERSITÉ PARIS 1

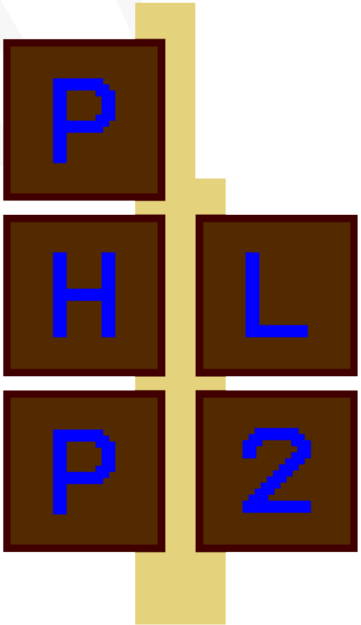
PANTHÉON SORBONNE

Développement Web - PHP

PHP - Bases

07 février 2021

Version 2



Fabrice BOISSIER <fabrice.boissier@univ-paris1.fr>

Copyright

Table des matières

1 Consignes Générales

Consigne Générale 0 : Vous devez lire le sujet.
Consigne Générale 1 : Vous devez respecter les consignes.
Consigne Générale 2 : Vous devez rendre le travail dans les délais prévus.

Consigne Générale 3 : Le non-respect des consignes entraînera des pénalités pouvant aller jusqu'à la note non négociable de 0.

Consigne Générale 4 : Le travail doit être rendu dans le format décrit à la section Format de Rendu.

Consigne Générale 5 : Le travail rendu ne doit pas contenir de fichiers binaires, temporaires, ou d'erreurs (`*~`, `*.o`, `*.a`, `*.so`, `*#*`, `*core`, `*.log`, `*.exe`, binaires, ...).

Consigne Générale 6 : Dans l'ensemble de ce document, la casse (caractères majuscules et minuscules) est très importante. Vous devez strictement respecter les majuscules et minuscules imposées dans les messages et noms de fichiers du sujet.

Consigne Générale 7 : Dans l'ensemble de ce document, `nom1-nom2` correspond à la combinaison des deux noms de votre binôme (par exemple pour Fabrice BOISSIER et Sabrine EDDED, cela donnera `boissier-edded`).

Consigne Générale 8 : Dans l'ensemble de ce document, le caractère ␣ correspond à une espace (s'il vous est demandé d'afficher ␣␣␣, vous devez afficher trois espaces consécutifs).

Consigne Générale 9 : Tout retard, même d'une seconde, entraîne la note non négociable de 0.

Consigne Générale 10 : La triche (échange de code, copie de code ou de texte, ...) entraîne **au mieux** la note non négociable de 0.

Consigne Générale 11 : En cas de problème avec le projet, vous devez contacter le plus tôt possible les responsables du sujet aux adresses mail indiquées.

Conseil : N'attendez pas la dernière minute pour commencer à travailler sur le sujet.

2 Format de Rendu

Responsable(s) du projet :	**Fabrice BOISSIER** <fabrice.boissier@univ-paris1.fr>
Balise(s) du projet :	**[PHP][DM]**
Nombre d'étudiant(s) par rendu :	2
Procédure de rendu :	Envoi par mail
Nom du répertoire :	`nom1-nom2-DM`
Nom de l'archive :	`nom1-nom2-DM.zip`
Date maximale de rendu :	07/02/2021 23h42
Durée du projet :	2 semaines
Architecture/OS :	WAMP ou MAMP
Langage(s) :	PHP
Compilateur/Interpréteur :	**Apache/PHP**
Options du compilateur/interpréteur :	

L'arborescence attendue pour le projet est la suivante :

```
nom1-nom2-DM/
nom1-nom2-DM/AUTHORS.txt
nom1-nom2-DM/
nom1-nom2-DM/src/
nom1-nom2-DM/src/exo1/
nom1-nom2-DM/src/exo1/exo1_fun.php
nom1-nom2-DM/src/exo2/
nom1-nom2-DM/src/exo2/exo2_fun.php
nom1-nom2-DM/src/exo3/
nom1-nom2-DM/src/exo3/exo3_fun.php
nom1-nom2-DM/src/exo4/
nom1-nom2-DM/src/exo4/exo4_fun.php
nom1-nom2-DM/src/exo5/
nom1-nom2-DM/src/exo5/exo5_fun.php
```

Les fichiers suivants sont requis :

`AUTHORS.txt` contient les nom(s) et prénom(s) du (ou des) auteur(s).

3 Aide Mémoire

Le travail doit être rendu au format ***.zip***, c'est-à-dire une archive **zip** compressée avec un outil adapté (les logiciels *7zip* ou *Keka* sont gratuits et adaptés).
Tout autre format d'archive (rar, 7zip, gz, gzip, bzip, ...) ne sera pas pris en compte, et votre travail ne sera pas corrigé (entraînant la note de 0).

Dans ce sujet précis, vous ferez du code en PHP, qui affichera les résultats dans une page HTML. Les valeurs seront affichées dans une *textarea* dont le texte est généré par des outils multiplateformes supportant les retours à la ligne UNIX (**\n**). Il ne faut donc pas inclure de balise **"
"** pour retourner à la ligne, mais un **"\n"**.

Pour réaliser le travail demandé, nous vous fournirons pour chaque exercice au moins 2 fichiers : **exoN_res.php** (le fichier qui sera appelé pour voir le résultat de votre travail), et **exoN_fun.php** (le fichier contenant la fonction que vous devez coder dans chaque exercice). Optionnellement, un fichier **exoN_data.php** (ou d'autres) peu(ven)t vous être fourni(s) pour indiquer le format de données en entrée. Le **N** correspond au numéro de l'exercice.

Dans tous les cas, vous ne devez rendre que le fichier **exoN_fun.php** avec au moins la fonction demandée remplie (qui peut faire appel à d'autres fonctions que vous définirez dans le **même** fichier). Les autres fichiers seront générés par nos soins pour tester vos fonctions.

Vous ne devez **PAS** utiliser la fonction **echo** pour écrire ! Il faut retourner une chaîne de caractères correctement formattée.

4 Exercice 1

Nom du(es) fichier(s) :	**exo1_fun.php** [my_Calculette(int, int, string)]
Répertoire :	**nom1-nom2-DM/src/exo1/**
Droits sur le répertoire :	
Droits sur le(s) fichier(s) :	
Fonctions recommandées :	**(Bases PHP)**, **(Maths PHP)**, **return**

Objectif : Le but de l'exercice est de créer une mini calculatrice en PHP.

Vous devez écrire une fonction nommée **my_Calculette** qui prendra trois paramètres (deux nombres, puis l'opérateur), et renverra le résultat de l'opération désignée ou un message d'erreur.
Vous devez implémenter les 5 opérations suivantes : l'addition (symbole **+**), la soustraction (symbole **-**), la multiplication (lettre *****), la division (symbole **/**), et le reste de la division euclidienne (symbole **%**).
À la fin du calcul, votre fonction doit renvoyer le résultat.

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette(42, 38, "+");
  echo($my_text);
?>
</textarea>
```

Cas général PHP

```
<textarea cols="80" rows="25" readonly="readonly">
80</textarea>
```

Cas général PHP exécuté

Les deux premiers paramètres doivent être des entiers, et le troisième doit être une chaîne de caractères. Si ça n'est pas le cas, vous devez renvoyer le texte suivant.

Incorrect␣parameters␣type

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette("+", 38, 42);
  echo($my_text);
?>
```

```
</textarea>
```

Cas d'erreur 1 : mauvais paramètres

```
<textarea cols="80" rows="25" readonly="readonly">
Incorrect parameters type</textarea>
```

Cas d'erreur 1 exécuté

Si le troisième paramètre donné n'est ni un **+**, ni un **-**, ni un *****, ni un **/**, ni un **%**, alors vous devez renvoyer le message d'erreur suivant.

Unknown operator

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette(42, 38, "A");
  echo($my_text);
?>
</textarea>
```

Cas d'erreur 2 : opérateur inconnu

```
<textarea cols="80" rows="25" readonly="readonly">
Unknown operator</textarea>
```

Cas d'erreur 2 exécuté

Si le deuxième paramètre donné à la division est 0, vous devez renvoyer le message d'erreur suivant.

Division by 0 is forbidden

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette(42, 0, "/");
  echo($my_text);
?>
</textarea>
```

Cas d'erreur 3 : division par 0

```
<textarea cols="80" rows="25" readonly="readonly">
Division by 0 is forbidden</textarea>
```

Cas d'erreur 3 exécuté

Si plusieurs des problèmes précédents sont rencontrés simultanément, vous devez les gérer dans cet ordre de priorité : le problème de type en priorité, l'opérateur inconnu en second, et la division par 0 en dernier.

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette("A", 42, 0);
  echo($my_text);
?>
</textarea>
```

Cas d'erreurs

```
<textarea cols="80" rows="25" readonly="readonly">
Incorrect parameters type</textarea>
```

Cas d'erreurs exécuté

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo1_fun.php");
  $my_text = my_Calculette(42, 0, "A");
  echo($my_text);
?>
</textarea>
```

Cas d'erreurs

```
<textarea cols="80" rows="25" readonly="readonly">
Unknown operator</textarea>
```

Cas d'erreurs exécuté

5 Exercice 2

Nom du(es) fichier(s) :	**exo2_fun.php** [my_Sapin(int)]
Répertoire :	**nom1-nom2-DM/src/exo2/**
Droits sur le répertoire :	
Droits sur le(s) fichier(s) :	
Fonctions recommandées :	**(Bases PHP)**, **return**

Objectif : Le but de l'exercice est d'afficher un sapin en ASCII art, dont la taille varie selon le paramètre donné.

Vous devez écrire la fonction **my_Sapin** qui prendra un entier en paramètre, et affichera selon ce paramètre un sapin.

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo2_fun.php");
  $my_text = my_Sapin(2);
  echo($my_text);
?>
</textarea>
```

Cas général PHP

```
<textarea cols="80" rows="25" readonly="readonly">
  /\
 /  \
/    \
______
  ||
</textarea>
```

Cas général PHP exécuté

De façon précise, voici les spécifications pour les cas 1, 4, et 5 :

```
 /\      1 espace / 0 espace  \
/  \     0 espace / 2 espaces \
____     4 _
 ||      1 espace 2 |
```

Cas général 1

```
    /\          4 espaces / 0 espace  \
   /  \         3 espaces / 2 espaces \
  /    \        2 espaces / 4 espaces \
 /      \       1 espace  / 6 espaces \
/        \      0 espace  / 8 espaces \
__________      10 _
    ||          4 espaces 2 |
```

Cas général 4

```
     /\         5 espaces / 0 espace   \
    /  \        4 espaces / 2 espaces  \
   /    \       3 espaces / 4 espaces  \
  /      \      2 espaces / 6 espaces  \
 /        \     1 espace  / 8 espaces  \
/          \    0 espace  / 10 espaces \
____________    12 _
     ||         5 espaces 2 |
```

Cas général 5

Une exception doit être gérée, dans le cas où le paramètre donné est inférieur ou égal à 0, vous afficherez :

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo2_fun.php");
  $my_text = my_Sapin(0);
  echo($my_text);
?>
</textarea>
```

Cas 0

```
<textarea cols="80" rows="25" readonly="readonly">
/\
||
</textarea>
```

Cas 0 exécuté

ATTENTION

Les retours à la ligne ne doivent pas être faits avec la balise **"
"**, mais avec **"\n"**. (se référer à la section Aide Mémoire)

6 Exercice 3

Nom du(es) fichier(s) :	**exo3_fun.php** [my_Reduction(array, int)]
Répertoire :	**nom1-nom2-DM/src/exo3/**
Droits sur le répertoire :	
Droits sur le(s) fichier(s) :	
Fonctions recommandées :	**(Bases PHP)**, **date_format**, **(int)**

Objectif : Le but de l'exercice est de calculer des réductions sur des listes de factures.

A partir d'un fichier de factures émises stockées dans des tableaux, vous allez afficher ces factures en ajoutant 3 colonnes supplémentaires (le montant avec réduction appliquée, le taux de réduction, et le montant de la réduction). Tous les montants sont stockés sous forme de centimes (300 signifie 300 centimes, soit 3 euros).

Pour rappel, voici comment se calculent les réductions :

Prix final après réduction :	$PrixFinal =$	$PrixInitial - Reduction$
Réduction (20%) depuis un prix :	$Reduction =$	$PrixInitial \times 20 \div 100$
Prix initial si réduction (20%) :	$PrixInitial =$	$PrixFinal \div 0,80(Coef)$
Coefficient de réduction de 20% :	$Coefficient(0,80) =$	$1 - (20 \div 100)$

Le tableau en entrée est de cette forme :

```
[[ID_client (int), Date (DateTime), ID_Facture (int), Montant (
   int)],

 [ID_client (int), Date (DateTime), ID_Facture (int), Montant (
   int)],

  ...]
```

Tableau en entrée

Le fichier de sortie sera de cette forme :

```
ID client ; Date ; Num Facture ; Montant Final ; Montant Initial ; Taux
   Reduction ; Reduction
```

Sortie attendue

Exemple de structure d'entrée :

```
42421337;2006/06/06;456789;200
36153615;2018/05/30;123456;123
```

Exemple de données en entrée

Sortie attendue pour cette entrée si on effectue une réduction de 33% :

```
42421337;06/06/06;456789;134;200;33;66
36153615;18/05/30;123456;83;123;33;40
```

Sortie attendue pour l'entrée précédente avec 33% de réduction

(Les fichiers que nous testerons ne seront pas de mauvaise qualité : les données seront toujours correctement formatées)

Attention : Certains nombres risquent de subir des virgules. Pour éviter cela, utilisez **`((int) (x / y))`** pour ne conserver que la partie entière (`123` en valeur initiale donnera donc par division `40` en réduction et `83` en valeur finale). Attention, cette fonction tronque selon la virgule : si vous tronquez avant ou après certaines opérations, les résultats ne seront pas les mêmes ! Lors du calcul des valeurs, choisissez avec attention la ligne exacte où vous tronquerez les nombres, afin de ne pas ajouter ou retirer 1 centime.

Vous devez rendre un fichier nommé **`exo3_fun.php`** qui contiendra au moins la fonction **`my_Reduction`** qui effectuera le calcul et renverra une chaîne contenant le résultat.
La fonction prendra en paramètre un tableau qui contient lui-même des tableaux associatifs (chaque case du tableau général correspond à une facture).

```php
$tab1["ID_Client"] = 42421337;
$tab1["Date"] = new DateTime('2006-06-06');
$tab1["ID_Facture"] = 456789;
$tab1["Montant"] = 200;
$data[] = $tab1;

$tab1["ID_Client"] = 36153615;
$tab1["Date"] = new DateTime('2018-05-30');
$tab1["ID_Facture"] = 123456;
$tab1["Montant"] = 123;
$data[] = $tab1;
```

Tableau en entrée (exo3_data.php)

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
```

```
  require_once("exo3_data.php");
  require_once("exo3_fun.php");
  $my_text = my_Reduction($data, 33);
  echo($my_text);
?>
</textarea>
```

Appel de la fonction

```
<textarea cols="80" rows="25" readonly="readonly">
42421337;06/06/06;456789;134;200;33;66
36153615;18/05/30;123456;83;123;33;40
</textarea>
```

Sortie HTML attendue

ATTENTION

Les retours à la ligne ne doivent pas être faits avec la balise **"
"**, mais avec **"\n"**. (se référer à la section Aide Mémoire)

7 Exercice 4

Nom du(es) fichier(s) :	**exo4_fun.php** [my_TicTacToe(array)]
Répertoire :	**nom1-nom2-DM/src/exo4/**
Droits sur le répertoire :	
Droits sur le(s) fichier(s) :	
Fonctions recommandées :	**(Bases PHP)**, **return**

Objectif : Le but de l'exercice est de déterminer quel est le gagnant d'une partie de morpion (ou *Tic Tac Toe* en angais).

Les règles sont relativement simples : chaque joueur place un **X** ou un **O** à chaque tour dans un tableau de 3 cases sur 3 cases, le premier joueur alignant trois fois son symbole a gagné (une ligne verticale, horizontale, ou en diagonale). Il est relativement simple de faire un match nul (aucun joueur n'arrive à aligner trois fois son symbole).

```
X O X
O X O
X O O
```

Exemple de jeu où le joueur X a gagné par la diagonale

Vous devez écrire une fonction nommée **my_TicTacToe** qui prendra en paramètre un tableau de tableaux. La fonction renverra tout d'abord le tableau de jeu, puis elle renverra le gagnant.

Exemple d'entrée :

```php
$line1 = array("X", "O", "X");
$line2 = array("O", "X", "O");
$line3 = array("X", "O", "O");

$game[] = $line1;
$game[] = $line2;
$game[] = $line3;
```

Exemple de tableau de jeu (exo4_data.php)

```php
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo4_data.php");
  require_once("exo4_fun.php");
  $my_text = my_TicTacToe($game);
  echo($my_text);
?>
</textarea>
```

Appel de la fonction

Si un joueur est gagnant, vous devez l'indiquer avec les chaînes de caractères :

Player␣X␣won
Player␣O␣won

Si aucun joueur n'est gagnant, vous devez l'indiquer ainsi :

It's␣a␣draw

```
<textarea cols="80" rows="25" readonly="readonly">
XOX
OXO
XOO
Player X won</textarea>
```

Cas général gagnant

```
<textarea cols="80" rows="25" readonly="readonly">
OXX
XXO
OOX
It's a draw</textarea>
```

Cas général en match nul

Un cas d'erreur doit être géré avant tout affichage. Si le tableau contient des caractères autres que **X** ou **O**, il faut indiquer le message suivant :

Incorrect␣game

```
$line1 = array("X", "O", "A");
$line2 = array("O", "X", "O");
$line3 = array("O", "O", "O");

$game[] = $line1;
$game[] = $line2;
$game[] = $line3;
```

Exemple de tableau de jeu incorrect (exo4_data.php)

```
<textarea cols="80" rows="25" readonly="readonly">
Incorrect game</textarea>
```

Cas d'erreur

ATTENTION

Les retours à la ligne ne doivent pas être faits avec la balise **"
"**, mais avec **"\n"**. (se référer à la section Aide Mémoire)

8 Exercice 5

Nom du(es) fichier(s) :	**exo5_fun.php** [my_Loginisation(array, array)]
Répertoire :	**nom1-nom2-DM/src/exo5/**
Droits sur le répertoire :	
Droits sur le(s) fichier(s) :	
Fonctions recommandées :	**gettype**, **in_array**, **strlen**, **substr**

Objectif : Le but de l'exercice est de créer des logins à partir d'une liste de noms et prénoms.

Il est courant en informatique de devoir créer des logins. Dans cet exercice, il faudra créer des logins pour un groupe précis de personnes.

En entrée, deux paramètres sont donnés : un tableau avec une liste de référence, et un autre tableau avec une liste de personnes dont il faut créer les logins. Afin de ne pas créer de login à n'importe qui, seules les personnes apparaissant dans la liste de référence disposeront d'un login.
La construction du login se fait comme suit : l'initiale du prénom en minuscule, le nom de famille en minuscules, les deux derniers chiffres de l'année de naissance. Par exemple, pour ***Eva Bien*** née en *1998*, le login attendu est **ebien98**.
Dans certains cas, certaines personnes ne disposent que de l'année de naissance. Dans ce cas, on utilise toujours les deux dernières décimales pour le login, mais il faut s'assurer que la base de référence contienne la même information. Si deux utilisateurs ont les mêmes nom et prénom, mais pas la même date de naissance, il s'agira de deux personnes distinctes.

Exemple d'entrées :

```php
$ReferenceList[] = array("Marc", "Acin", "05-05-1998");
$ReferenceList[] = array("Charles", "Attan", "20-06-1997");
$ReferenceList[] = array("Eva", "Bien", "23-02-1998");
$ReferenceList[] = array("Jean", "Bombeur", "17-01-1999");
$ReferenceList[] = array("Jean", "Bonnot", "1997");
$ReferenceList[] = array("Claire", "Hon", "14-07-1997");
$ReferenceList[] = array("Lydie", "Le", "02-03-1997");
$ReferenceList[] = array("Julie", "Tali", "18-12-1998");
$ReferenceList[] = array("Oreste", "Torrent", "22-04-1998");
$ReferenceList[] = array("Elena", "Turelle", "08-10-1998");
```

Exemple de liste de référence

```php
$RequestList[] = array("Eva", "Bien", "23-02-1998");
$RequestList[] = array("Jean", "Bombeur", "24-12-2000");
$RequestList[] = array("Jean", "Bonnot", "1997");
$RequestList[] = array("Julie", "Tali", "18-12-1998");
$RequestList[] = array("Clement", "Thine", "13-04-1998");
```

Exemple de liste de demandes

Les tableaux seront fournis à la fonction ainsi :

```
<textarea cols="80" rows="25" readonly="readonly">
<?php
  require_once("exo5_listeref.php");
  require_once("exo5_data.php");
  require_once("exo5_fun.php");
  $my_text = my_Loginisation($ReferenceList, $RequestList);
  echo($my_text);
?>
</textarea>
```

Appel de la fonction

La sortie sera de cette forme :

```
<textarea cols="80" rows="25" readonly="readonly">
ebien98
jbonnot97
jtali98
</textarea>
```

Sortie HTML attendue

Vous devez rendre un fichier nommé **exo5_fun.php** qui contiendra au moins la fonction **my_Loginisation** qui effectuera le traitement et renverra une chaîne de caractères. La fonction prendra en paramètre deux tableaux : le premier contiendra la liste de référence des personnes autorisées, et le deuxième contiendra la liste des personnes demandeuses.

On considèrera que les listes en entrée sont correctement formattées, et jamais vides.

ATTENTION

Les retours à la ligne ne doivent pas être faits avec la balise **"
"**, mais avec **"\n"**. (se référer à la section Aide Mémoire)