

PHP - EXERCICES

TP 1

13/10/2022

Exercice 1 : Complétez le fichier `exo1.php` de façon qu'il affiche l'heure, les minutes et les secondes courantes. Chaque accès au script donne un affichage différent, sauf si deux accès consécutifs sont effectués en moins d'une seconde. La fonction utile pour cet exercice est la fonction `date` qu'il faut utiliser avec les bons formats (paramètres).

Exercice 2 : Complétez le fichier `exo2.php` de façon que le script affiche la date courante (i.e. la date au moment où on invoque le script) en français. La fonction utile pour cet exercice est la fonction `date()` déjà utilisée dans l'exercice précédent. Vous pouvez aussi expérimenter la forme `switch` à la place du `if/else`.

Exercice 3 : Complétez le fichier `exo3.php` de façon que le script affiche tous les diviseurs d'un entier `N` passé en paramètre. Un nombre `y` est un diviseur d'un nombre `x` (ou encore, `x` est un multiple de `y`) si le reste de la division de `x` par `y` est égal à zéro. Pour tester votre script, n'oubliez pas de passer son paramètre dans l'URL: `http://localhost/web/php-tp1/exo3.php?n=1357`

Exercice 4 : En vous inspirant de l'exercice précédent, complétez le fichier `exo4.php` de façon que le script teste si un nombre entier `N` passé en paramètre est premier ou non. Un nombre entier est premier s'il n'a aucun autre diviseur que 1 et lui-même.

Exercice 5 : Complétez le fichier `exo5.php` de façon que le script affiche les 10 tables de multiplications de 1 à 10 sous la forme d'une table HTML.

Exercice 6 : Complétez le fichier `exo6.php` de façon que le script produise un échiquier `NxN` sous la forme d'une table HTML. Si le paramètre `taille` est passé en paramètre, c'est la taille de l'échiquier. Si le paramètre est absent, la taille par défaut est 8. Pour réaliser simplement cet exercice, vous devez utiliser le style CSS présent dans le fichier `squelette`, et en particulier les classes CSS `blanc` et `noir` qui permettent d'afficher les éléments HTML `<TD>` avec un fond blanc ou noir. Vous devez trouver une règle simple pour déterminer si une case (i.e. un élément HTML `<TD>`) doit être blanche ou noire.

Exercice 7 : Reprenez l'exercice 2 en utilisant les tableaux présents dans le fichier `exo7.php` (fourni). En utilisant ces tableaux, il n'est plus nécessaire d'utiliser de forme conditionnelle (`if` ou `switch`).

Exercice 8 : Complétez le fichier `exo8.php`, de façon que le script génère des numéros de Loto (5 nombres entre 1 et 49, et le numéro chance entre 1 et 10) et les affiche, les 5 nombres apparaissant dans l'ordre. Pour trier un tableau, vous devez utiliser la fonction `sort()`. Une autre fonction utile pour cet exercice est la fonction `in_array()`.

Exercice 9 : Complétez le fichier `exo9.php`, action du formulaire contenu dans `exo9.html` (fourni) de façon que le script affiche un message de bienvenue en anglais qui dépend de la civilité (Monsieur, Madame ou Mademoiselle) et de l'heure courante (i.e. l'heure du moment où on invoque le script). Les différents messages affichés sont :

- Good morning ... welcome to EMSI! à partir de 0h et avant 12h
- Good afternoon ... welcome to EMSI! à partir de 12h et avant 18h
- Good evening ... welcome to EMSI! à partir de 18h et avant 24h

Exercice 10 : Etant donné le fichier HTML `exo10.html` (fourni), écrivez le script PHP `exo10.php` qui est l'action du formulaire contenu dans `exo10.html`. Ce formulaire transmet une date au script `exo10.php` sous la forme de trois entiers, un numéro de jour (compris entre 1 et 31), un

numéro de mois (compris entre 1 et 12) et une année. En retour, le script `exo10.php` affiche le jour de la semaine correspondant à cette date. La fonction utile pour cet exercice est la fonction `mktime()`. Une fois le script `exo10.php` terminé, écrivez le script `exo10-html.php` pour remplacer le fichier statique `exo10.html` : ce script doit générer le code HTML contenu dans `exo10.html` !

Exercice 11 : Le but de cet exercice est de fabriquer un répéteur pour s'entraîner à effectuer des multiplications. L'utilisateur (le client) invoque le script `exo11-formulaire.php` qui affiche en retour un formulaire présentant une multiplication dont les opérandes ont été choisis au hasard dans l'intervalle $[2, 10]$. L'utilisateur soumet, via le formulaire, son résultat pour la multiplication au script de test `exo11-action.php` (l'action du formulaire). Ce script teste si le résultat fourni par l'utilisateur est correct, et affiche en retour le message adéquat, suivi d'un lien pour effectuer éventuellement une nouvelle multiplication. La fonction utile pour cet exercice est la fonction `rand()`. Vous trouverez deux exemples de code HTML produit par ce script.

Exercice 12 : Le but de cet exercice est de fabriquer un jeu qui consiste pour l'utilisateur à deviner un nombre tiré au hasard par le programme. Pour découvrir le nombre secret, l'utilisateur (le client) soumet consécutivement, via un formulaire, des nombres (un par un) à un script (l'action du formulaire), script qui compare le nombre fourni par l'utilisateur au nombre secret et affiche :

- un message de félicitations si les nombres sont identiques, suivi du nombre total de tentatives qui ont été nécessaires pour découvrir le nombre, suivi d'un lien pour éventuellement jouer une autre partie
- le message "trop grand" si le nombre fourni par l'utilisateur est plus grand que le nombre secret
- le message "trop petit" si le nombre fourni par l'utilisateur est plus petit que le nombre secret

NB : Dans les deux derniers cas, le script affiche, à la suite du message, un nouveau formulaire pour que l'utilisateur puisse soumettre un nouveau nombre.

Vous devez réaliser le jeu avec un unique script PHP `exo12.php`, qui doit en même temps produire un formulaire et être l'action de ce formulaire ! En particulier, la première fois où on invoque ce script, il ne doit afficher que le formulaire.

Exercice 13 : Le but de cet exercice est d'enregistrer un client et lui permettre de s'authentifier. Veuillez compléter les tâches suivantes :

- créez une base de données MySQL en utilisant l'utilitaire `phpmyadmin` et y importer le script `exo13.sql`
- complétez la page `exo13.php` qui ne doit être accessible que si l'utilisateur est authentifié et qu'il a le rôle "admin". S'il n'est pas authentifié, il doit être redirigé vers la page "`exo13-login.php`". Vous aurez besoin de la fonction "`header()`" pour effectuer la redirection.
- la page "`exo13-login.php`" contient un formulaire de login. Le traitement doit être fait sur la même page. Si l'authentification est correcte on redirige vers la page `exo13.php`, sinon on lui affiche un message d'erreur dans un div "error" et on affiche encore le même formulaire
- si l'utilisateur n'a pas encore de compte, il peut le créer sur la page `exo13-register.php`. Pensez à gérer les cas où l'adresse e-mail existe déjà dans la base de données.