

CONSIGNES ET PREREQUIS

L'ensemble de cette corbeille d'exercices doit être réalisée en PHP. Pour ce faire vous pouvez utiliser la plateforme WAMP ou XAMPP.

Il existe des équivalents sous Linux (Lamp) et macOS (Mamp).

Pour bien appréhender PHP, comme pour n'importe quel langage, il vous faudra beaucoup manipuler.

N'hésitez donc pas à le faire en dehors de cette corbeille d'exercices.

Pour ceux qui ont déjà de bonnes bases en PHP et qui auraient terminés cette corbeille rapidement, vous pouvez améliorer les scripts existants ou en créer de nouveaux.

RESSOURCES

- [How to Use XAMPP to Run Your Own Web Server](#)
- [How To A Configure Virtual Host On Windows 10](#)

EXERCICES

Vous devrez créer un fichier php par exercice que vous nommerez `exercicen.php` (où *n* représente le numéro de l'exercice). Ces fichiers devront être positionnés dans un répertoire différent de celui de la racine du serveur Web qui pourra être redirigé via des Virtual Hosts (voir ressources).

EXERCICE 1 – AFFICHAGE D'INFORMATIONS BASIQUES

Ecrire un script permettant de 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 2 – MANIPULATION D'UNE CHAÎNE DE CARACTÈRES

Déclarer une variable contenant une chaîne de caractères puis l'afficher de différentes manières :

1. Mettre la première lettre de la phrase en majuscule
2. Mettre chaque mot de la phrase avec son initiale en majuscule
3. Remplacer chaque espace par des _
4. Compter le nombre de mots.

EXERCICE 3 – TABLEAU MULTIDIMENSIONNEL

Soit le tableau suivant :

Marque	Modèle	Type
Audi	A1	Citadine
Volkswagen	Passat	Berline
Volkswagen	Golf	Compact
Mazda	CX-5	SUV

Reproduire dans un script ce tableau en utilisant un tableau associatif multidimensionnel (clés et valeurs). Trier ce tableau par marque puis par modèle, et afficher le résultat dans une boucle.

EXERCICE 4 - POO

En utilisant le tableau précédent :

1. Déclarer une classe voiture contenant comme attributs les éléments de l'entête ;
2. Créer un constructeur afin de recevoir des paramètres par défaut pour chaque nouvelle instance ;
3. Créer une méthode permettant d'afficher toutes les propriétés de l'objet instancié ;
4. Instancier un objet par ligne du tableau et lancer la fonction pour chacun d'entre eux ;
5. Changer un ou plusieurs attributs de l'une des instances puis ré-afficher les valeurs (une méthode peut éventuellement être créée pour setter la valeur).

EXERCICE 5 - FORMULAIRE

1. Créer une page intégrant un formulaire de type POST contenant les 3 champs du tableau précédent.
 2. Faire un script qui récupère les données envoyées en POST puis qui les affiche de 3 manières différentes en fonction du paramètre passé en GET (html, csv ou json). Si le paramètre est invalide un message doit l'indiquer, et s'il est absent prendre html par défaut.
- ➔ Comment appelle-t-on les variables `$_POST` et `$_GET` ?

EXERCICE 6 - INCLUSIONS

1. Créer trois pages php : `header.php`, `nav.php` et `footer.php` :
 - Le fichier `header.php` contiendra le début du code html (balise header).
 - Le fichier `nav.php` devra lister tous les fichiers exercice présent dans le répertoire sous forme de liens. Le fichier en cours ne sera pas un lien.
 - Le fichier `footer.php` contiendra le pied de page et la fin du code html.
2. Modifier chaque fichier exercice afin d'inclure au bon endroit les 3 fichiers créés précédemment. Exemple pour le fichier `exercice1.php` :

<code>header.php</code>
<code>nav.php</code>
<i>Contenu de l'exercice 1</i>
<code>footer.php</code>

- ➔ La structure adoptée dans cette corbeille est-elle la meilleure ? Si non pourquoi ?
- ➔ Quelle(s) instruction(s) permette(nt) d'inclure des fichiers dans PHP ? Quelles sont les différences ?

EXERCICE 7 – SESSIONS ET COOKIES

Créer une page `session.php` qui permettra de forcer l'authentification en utilisant le principe des sessions. Cette page devra :

1. Tester si l'utilisateur est connecté ;
2. Afficher un formulaire contenant les champs email et mot de passe (mettre les bons attributs pour les balises `input`) si l'utilisateur n'est pas connecté ;
3. Vérifier que les informations saisies via le formulaire sont valides. Pour ce faire il faudra intégrer un tableau multidimensionnel contenant les clés `email`, `password` et `name`. Mettre en dur 1 ou plusieurs couples nom/mot de passe qui devront matcher avec ceux saisis dans la page et des noms fictifs. Par exemple :

email	password	name
igallet@viacesi.fr	sn	Jérémy
rcoma@viacesi.fr	rouan	Roland

4. Renvoyer un message si le login n'est pas valide ou si le mot de passe est incorrect. Vous devrez intégrer des filtres de nettoyage et de validation dans votre script afin de filtrer les valeurs saisies et de vérifier que les champs saisis sont dans le bon format ;
5. Créer un cookie contenant l'email de l'utilisateur et qui sera la valeur par défaut du champs login du formulaire (s'il est défini). Le cookie aura une durée d'expiration d'un an.

Créer une page `logout.php` permettant de déconnecter l'utilisateur.

Modifier enfin la page `header.php` afin d'inclure la page de session et d'afficher le nom de la personne connectée ainsi qu'un lien vers la page de déconnexion.

- ➔ Quelle(s) amélioration(s) pourraient être apportée(s) au niveau de la gestion des utilisateurs ?
- ➔ Que faut-il obligatoirement faire sur un site Web lorsque l'on utilise des cookies ?
- ➔ Quelle est la durée de vie maximale d'un cookie ?

ET APRES...

Si vous avez terminé tous les exercices et que votre CER est aux petits oignons, vous pouvez améliorer votre mini-site par exemple en ajoutant une page `index.php` et en revoyant la structure des pages ainsi que le design.