

TP2 – INGESUP B1

Exercice 1

Nous souhaitons réaliser une calculatrice *client/serveur*.

Notez que l'exercice n'a pas vraiment d'intérêt en pratique car tous les clients peuvent réaliser leurs calculs sans passer par un serveur. Ici, l'objectif est de vous faire manipuler pour la première fois des données envoyées en utilisant la méthode *POST*.

Réalisez le formulaire suivant :

The diagram shows a web form titled "Calculatrice". It is enclosed in a dashed blue border. Inside, there are three labels with corresponding input fields: "Valeur 1 :" followed by a horizontal line, "Valeur 2 :" followed by a horizontal line, and "Résultat : 0". Below these fields, there are two buttons: "Additionner" and "Multiplier". The entire form is set against a light blue background.

Ensuite, ajoutez le code PHP nécessaire pour rendre cette page fonctionnelle. Faites en sorte que les valeurs 1 et 2 continuent à être affichées lorsque le formulaire qui présente le résultat du calcul est affiché.

Remarque : pour gérer deux boutons de soumission, vous devez donner le même nom aux deux boutons avec, par exemple, `name="op"` dans les deux balises `input`. Ensuite, vous donnez deux *values* différentes. Par exemple, `value="Additionner"` et `value="Multiplier"`. La valeur de `$_POST['op']` sera alors égale à "Additionner" si le premier bouton a été cliqué par l'utilisateur et "Multiplier" sinon. Notez également que l'attribut `value` d'un bouton permet de changer le texte affiché à l'intérieur.

Exercice 2

Nous souhaitons rendre les exercices 3, 7 et 8 du premier TP interactifs. Pour ce faire, je vous propose de réaliser le formulaire suivant et de le connecter au code réalisé lors du premier TP :

Exerice 3 du TP1

Règles :

a->b, c->d, f->g

Clés :

a, f

Résultat : b, g

Calculer

Exerice 7 du TP1

Nombre à convertir :

De la base :

Vers la base :

Résultat :

0

Calculer

Exerice 8 du TP1

1
11
21
1211
111221
312211
13112221

Hauteur :

Afficher

Exercice 3

Réalisez un formulaire permettant de poster un fichier contenant une image au format *jpg*.

Si une image a déjà été postée, elle est affichée en dessous du formulaire. Si une nouvelle image est postée, elle doit remplacer la précédente.

Exercice 4

Dans cet exercice, vous allez mettre en place un outil permettant de naviguer dans les produits d'un site marchand. Toutes les requêtes des visiteurs se font par la page *index.php*.

Par défaut, la page *index.php* affichent la liste des produits disponibles sur le site avec des liens vers les fiches qui décrivent les produits.

Ces fiches sont générées par la page *index.php* lorsque le numéro d'un produit est précisé dans l'url.

Par exemple, l'url *index.php?produit=2* génère la fiche du produit numéro 2. Chaque fiche doit contenir un lien vers la liste des produits.

Un produit est décrit dans un tableau relatif. Ce tableau contient le numéro, le nom, le prix et un bref descriptif du produit.

Commencez par écrire la fonction PHP *genererFicheProduit* qui génère la fiche d'un produit à partir d'un tableau relatif le décrivant.

Par exemple, vous trouvez ci-dessous un appel à la fonction *genererFicheProduit* et la fiche générée par cet appel :

```
$produit = array("numero"=>1,"nom"=>"Montre", "prix"=>"100 euros", "descriptif"=>"Une jolie montre.");  
genererFicheProduit($produit);
```

- **Nom** : Montre
- **Prix** : 100 euros
- **Descriptif** : Une jolie montre.

(Lien vers la liste des produits)

Ensuite, écrivez la fonction PHP *genererListeProduits* qui, étant donné un tableau contenant des produits, affiche une liste de liens vers les fiches des produits.

Vous trouvez ci-dessous un appel à la fonction *genererListeProduits* et la page générée par cet appel :

```
$produit1 = array("numero"=>1,"nom"=>"Montre", "prix"=>"100 euros", "descriptif"=>"Une jolie montre.");  
$produit2 = array("numero"=>2,"nom"=>"Souris", "prix"=>"10 euros", "descriptif"=>"Une jolie souris.");  
$produit3 = array("numero"=>3,"nom"=>"Clavier", "prix"=>"20 euros", "descriptif"=>"Un joli clavier.");  
$produits = array($produit1, $produit2, $produit3);  
genererListeProduits($produits);
```

- Montre
- Souris

- Clavier

Terminez votre programme en écrivant le code principal du fichier *index.php* afin qu'il génère la fiche d'un produit si son numéro est précisé dans l'url et la liste des produits dans le cas contraire.



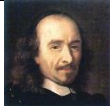

Pour le moment, la liste des produits sera définie au début de votre programme (Pour tester votre programme, vous pouvez utiliser la liste donnée ci-dessus). Notez que si un numéro est donné dans l'url mais qu'il ne correspond à aucun produit du site, vous devez afficher un message d'erreur (ou utiliser les headers pour générer une erreur "Page non trouvée").

Exercice 5

Cet exercice simple a pour objectif de vous faire manipuler les fichiers en PHP.

Vous allez lire des données formatées dans un fichier et les afficher à l'écran sous la forme d'un tableau.

Plus précisément, vous allez lire les informations du fichier joint au TP (*TP2_exo5.txt*) afin de réaliser le tableau suivant :

Prénom	Nom	Photo
Victor	Hugo	
Jean	De La Fontaine	
Pierre	Corneille	
Jean	Racine	

Bien évidemment, si je change les informations présentes dans le fichier, votre tableau doit changer automatiquement.

En d'autres termes, votre programme doit uniquement utiliser le fait que le caractère séparateur est le point-virgule et que le tableau est composé de trois colonnes intitulées respectivement "Prénom", "Nom", "Photo".

Toutes les autres informations doivent être extraites du fichier.

Exercice 6

L'objectif de cet exercice est de vous faire écrire dans un fichier en PHP.

Réalisez la fonction *logVisite()* qui écrit dans et à la suite du fichier *log.txt* la date, l'heure courantes et l'adresse IP du visiteur. Si le fichier *log.txt* n'existe pas, il doit être créé.

Afin de permettre à l'administrateur du site de savoir quelles sont les heures de la journée où le site est le plus utilisé, réalisez la fonction *afficherStatistiques()* qui affiche le nombre de visites effectuées aux différentes heures de la journée le nombre de visites

par exemple, s'il y a eu 2 visites le 30/03/2010 à 14h32 et 3 visites le 02/02/2011 à 14h50, vous devez afficher 5 visites pour le créneau 14h-15h.

Notez que, pour tester votre programme, vous pouvez afficher le nombre de visites pour chaque minute de la journée.

Exercice 7

L'objectif de cet exercice est de réaliser un compteur de visites pour votre site Web.

Le nombre de visites de votre site va être maintenu à jour dans un fichier et sera stocké sous la forme d'une chaîne de caractères. Pour que votre code soit facilement intégrable dans un site Web quelconque, je vous demande d'écrire les quatre fonctions décrites ci-dessous dans le fichier *compteur.inc.php*. Ce fichier est à inclure lorsque l'on souhaite utiliser le compteur :

- La fonction *lireNombre(\$nomFichier)* lit le nombre présent dans le fichier nommé *\$nomFichier*. Si le fichier n'existe pas, la valeur 0 doit être retournée.
- La fonction *ecrireNombre(\$nomFichier,\$nombre)* écrit le nombre *\$nombre* dans le fichier nommé *\$nomFichier*.
- Si le fichier n'existe pas, il est créé. Sinon, le nombre présent dans le fichier est remplacé par le nouveau.
- La fonction *compterVisites()* compte les visites : elle incrémente le nombre présent dans le fichier *compteur.txt* (si ce fichier n'existe pas, il devra être créé et initialisé).
- La fonction *afficherVisites()* lit le nombre présent dans le fichier *compteur.txt* et affiche ce nombre. Vous pouvez vous "amuser" à faire un affichage sympa (par exemple un dégradé de couleurs).

Exercice 8

Ecrire une fonction PHP qui permet d'afficher l'arborescence complète triée d'un dossier passé en paramètre.