

Développement Internet**PHP - Exercices****1 Notions de base**

1. Ecrivez une page `calcul.php` qui initialise une variable `i` à 25, l'affiche et calcule `i*2+1`
2. Une page `loop1.php` qui affiche les valeurs paires de 0 à 50 sur des lignes séparées
3. Une page `loop2.php` qui affiche 20 lignes indiquant des nombres ordinaux. Par exemple :
 - 1er passage
 - 2ème passage
 - 3ème passage
 - ...
 - 20ème passage
4. Une page `facture.php` qui crée 3 tableaux `nom_prod`, `quantite`, et `prix_unitaire` représentant respectivement une liste de produits sélectionnés dans une commande sur un magasin en ligne, la quantité de chaque produit et leur prix unitaire. Affichez ensuite un récapitulatif de la commande dans une table HTML, qui affiche chaque élément, leur prix unitaire et la quantité, ainsi que le prix total de la commande.
5. Modifiez `facture.php` afin d'utiliser une matrice `facture` mixte (tableau scalaire puis associatif), par exemple `facture[clé][indice]` où `clé` est une chaîne de caractère et `indice` est un nombre



2 Utilisation de fonctions

2.1 Traitement des chaînes de caractères

Considérons un tableau reprenant des adresses mail.

Dans une fonction prenant ce tableau en paramètre, déterminez tous les noms de domaines qui apparaissent (par exemple pour l'adresse `tnicodeme@he2b.be`, le nom de domaine est `he2b.be`), et le nombre de fois que chaque nom de domaine apparaît.

```
<?php
$emails = array("gerard.manfroid@chaudiere.be",
    "g.dubol@lachance.fr",
    "mimi@lachance.fr",
    "paul.ice@police.be");

function frequence_domaines($emails) {
    //TODO...
}

//frequence_domaines($emails) doit renvoyer :
array("chaudiere.be" => 1, "lachance.fr" => 2, "police.be" => 1);
?>
```

2.2 Traitement des tableaux

Créez une fonction `stripslashes()` utilisant des fonctions incorporées de PHP qui permettra de supprimer les caractères `"\"` au sein d'un vecteur `$titres` passé en paramètre.

```
<!DOCTYPE html><html>
<head>
<title>Labo DIN : instruction : supprimer les \ en bloc </title>
</head>
<body><?php
function stripslashes_array ( $arg = array() ) {
    # à compléter ...
}
$titres= array ("C\'est pas la mer à boire",
    "L\'univers sans \"faim\"",
    "Le \$, \"une monnaie de singe\"");
$newtitres = stripslashes_array($titres);
echo "$newtitres[1] <br>"; // L'univers sans "faim"
echo "$newtitres[2] <br>"; // Le $, "une monnaie de singe"?
?></body></html>
```

3 Gestion de formulaires

3.1 Saisie de données

Ecrivez une page `saisie.html` dans laquelle se trouve un formulaire permettant de rentrer les informations suivantes : `titre`, `texte` et `auteur`. Ce formulaire appelle le programme de `saisie_cmd.php`, qui affiche les données encodées.

- Créez ces pages, en appelant `saisie_cmd.php` avec la méthode GET.
- Idem, avec la méthode POST.
- Ajoutez un bouton "Annuler". Lorsque l'utilisateur clique sur ce bouton, la page `saisie_cmd.php` affiche uniquement "Opération annulée".
- Ajoutez un bouton "Convertir". Lorsque l'utilisateur clique sur ce bouton, la page `saisie_cmd.php` affiche les informations entrées dans le formulaire, converties en minuscules.

3.2 Quiz

Ecrivez un quiz d'arithmétique. Ce quiz affiche une question à la fois, et propose un calcul arithmétique basique formé de deux nombres entiers choisis aléatoirement, et d'une opération (addition, multiplication ou soustraction) choisie également au hasard.

- Version de base : le quiz indique à l'utilisateur si la réponse entrée est correcte ou non, puis permet de tenter une nouvelle question.
- Première variante : si l'utilisateur entre une réponse incorrecte, le quiz offre propose à nouveau la même question, avec jusqu'à trois tentatives par question.
- Deuxième variante : le joueur tente 20 questions, et le quiz tient compte du nombre de questions (**différentes**) déjà tentées, et du nombre de réponses correctes entrées. À la fin des questions, le quiz affiche son score au joueur.