

Durée : 1h30min.

Nom complet :

Exercice 1 (question de cours/lecture)

1) Que fait la fonction « `isset()` » ?

.....

.....

.....

2) Donner une expression régulière permettant de décrire une la forme d'une date saisie sous forme d'une chaîne de caractères : possibilité de séparer entre les éléments de la date par le tiré, la barre oblique ou l'espace si ces éléments sont composés par chiffres. Autrement, le mois est exprimé en lettres, et dans ce cas la séparation se fait par un espace. Le mot représentant le mois, peut ne pas avoir de sens (f).

- a. "1992-6-11"
- b. "1992/6/11"
- c. "1992 6 11"
- d. "1992 juin 11"
- e. "1992 JUIN 11"
- f. "1992 IBraHIMOVic 11"

.....

.....

.....

3) Lors de l'envoi d'un fichier (ex. image) dans un formulaire, c'est la superglobale `$_SERVER` qui est utilisée. Vrai ou Faux

.....

4) Dans ce groupe de fonctions {`addslashes` – `header` – `htmlentities` – `htmlspecialchars`} il y en a une qui n'appartient pas à la famille composée par les autres. Laquelle ?

.....

.....

5) « `array_push` » est utilisée pour insérer un nouvel élément à la fin d'un tableau. Vrai ou faux ?

.....

Exercice 2 (programmation web) [5pts]

Décrire de A à Z, la mise en place d'un système d'authentification basé sur les sessions, pour donner un accès protégé par mot de passe à une page « `cockpit.php` ». Ne pas dépasser une page (schémas et rédaction).

Critères d'évaluation :

- Organisation et enchaînement logique des étapes [1pt]
- Schémas et figures explicatifs [1pt]
- Code compact et correct [1pt]
- Code commenté et décomposé en fonctions [1pt]
- Utilisation de fonctions php natives [1pt]

Durée : 1h30min.

Nom complet :

Durée : 1h30min.

Nom complet :

Exercice 3 (application web minimale) [10pts]

« food nutrition dataset » est une base de données alimentaire qui existe sur *kaggle* sous le lien suivant : <https://www.kaggle.com/shrutisaxena/food-nutrition-dataset>

Le fichier « food.csv » est le seul utilisé dans cet exercice. Il regroupe une collection de données nutritionnelles organisées sous forme d'une matrice (des lignes et des colonnes) de 4713(aliments)x47(propriétés). Nous allons proposer une application web (site web minimal) pour afficher les données de « food.scv ».

Le site se compose de 3 fichiers PHP, et un fichier CSV : `categorie.php` , `food.scv` , `index.php` et `utils.php`.

Fichier `index.php`

```
<?php
    require_once 'utils.php';
    $categories = /* appel d'une fonction d'utils.php */
    sort($categories); // trier dans l'ordre lexical
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Nutrition</h1>
    <form action="categorie.php" method="GET">
        <select name="cat" id="56" onchange="this.form.submit()">
            <?php foreach($categories as $cat){ ?>
                <option value="<?php echo $cat ?>"><?php echo $cat ?></option>
            <?php } ?>
        </select>
    </form>
</body>
</html>
```

Le code `onchange="this.form.submit()"` permet de soumettre le formulaire à chaque sélection d'une nouvelle option du bouton select.

Question : préparer la variable `$categories` en faisant adéquatement appel à une fonction du fichier « `utils.php` » et en lui passant l'argument qu'il faut.

Réponse :

Durée : 1h30min.

Nom complet :

Fichier **categorie.php**

```
<?php
    /* ajouter une ligne pour le bon fonctionnement de ce script */
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>détail categorie</h1>
    <?php
        if(isset($_GET)){ // starts first if
            $cat = $_GET['cat'];
            echo "<h2>".$cat."</h2>";
            $sa = select_all("food.csv");
            $detail = detaille_categorie($sa, $cat);
        }
        <table border="1">
            <tr>
                <?php $cols = meta_data($sa); ?>
                <?php foreach($cols as $col){ ?>
                    <td><?php echo $col ?></td>
                <?php } ?>
            </tr>
            <?php foreach($detail as $desc){ ?>
                <tr>
                    <?php foreach($desc as $col){ ?>
                        <td><?php echo $col ?></td>
                    <?php } ?>
                </tr>
            <?php } ?>
        </table>
        <?php } // ends the first if ?>
    </body>
</html>
```

Question : que faut il ajouter en début de ce script pour son bon fonctionnement ?

Réponse :

Durée : 1h30min.

Nom complet :

Fichier **utils.php**

```
<?php
function afficher_die($t, $msg){
    /* affiche verticalement le tableau $t, puis affiche le message $msg et
       stoppe l'exécution */
}

function afficher($ligne){
    /* affiche verticalement le tableau $t */
}

function select_all($f){
    /* retourne un tableau dont les valeurs sont données par les lignes du
       fichier csv $f */
}

function meta_data($t){
    /* retourne le premier élément du tableau $t */
}

function compter($t){
    /* retourne le nombre de lignes dans le tableau $t sauf la première */
}

function detaille_categorie($content, $categorie){
    /* retourne un tableau de toutes les lignes du tableau $content qui ont
       la valeur $categorie dans la première colonne */
}

function select_all_categories($f){
    /* retourne sans répétitions toutes les catégories, première colonne, du
       fichier csv $f */
}
```

Question : Donner une implémentation de toutes les fonctions du fichier « utils.php » en faisant très attention au type des paramètres (quelquefois c'est un tableau mais d'autres c'est un fichier ! entre autres)

Réponses :

Durée : 1h30min.

Nom complet :

Durée : 1h30min.

Nom complet :

Annexe

- Pour avoir une idée globale sur le fichier « food.csv », voir le tableau séparé.
- La première ligne de « food.csv », les noms des colonnes sont donnés comme suit :

```
<?php afficher_die(meta_data(select_all("food.csv")), "Toutes les colonnes."); ?>
```

```
Array
(
    [0] => Category
    [1] => Description
    [2] => Nutrient Data Bank Number
    [3] => Data.Alpha Carotene
    [4] => Data.Ash
    [5] => Data.Beta Carotene
    [6] => Data.Beta Cryptoxanthin
    [7] => Data.Carbohydrate
    [8] => Data.Cholesterol
    [9] => Data.Choline
    [10] => Data.Fiber
    [11] => Data.Kilocalories
    [12] => Data.Lutein and Zeaxanthin
    [13] => Data.Lycopene
    [14] => Data.Manganese
    [15] => Data.Niacin
    [16] => Data.Pantothenic Acid
    [17] => Data.Protein
    [18] => Data.Refuse Percentage
    [19] => Data.Retinol
    [20] => Data.Riboflavin
    [21] => Data.Selenium
    [22] => Data.Sugar Total
    [23] => Data.Thiamin
    [24] => Data.Water
    [25] => Data.Fat.Monosaturated Fat
    [26] => Data.Fat.Polysaturated Fat
    [27] => Data.Fat.Saturated Fat
    [28] => Data.Fat.Total Lipid
    [29] => Data.Household Weights.1st Household Weight
    [30] => Data.Household Weights.1st Household Weight Description
    [31] => Data.Household Weights.2nd Household Weight
    [32] => Data.Household Weights.2nd Household Weight Description
    [33] => Data.Major Minerals.Calcium
    [34] => Data.Major Minerals.Copper
    [35] => Data.Major Minerals.Iron
    [36] => Data.Major Minerals.Magnesium
    [37] => Data.Major Minerals.Phosphorus
    [38] => Data.Major Minerals.Potassium
    [39] => Data.Major Minerals.Sodium
    [40] => Data.Major Minerals.Zinc
    [41] => Data.Vitamins.Vitamin A - IU
    [42] => Data.Vitamins.Vitamin A - RAE
    [43] => Data.Vitamins.Vitamin B12
    [44] => Data.Vitamins.Vitamin B6
    [45] => Data.Vitamins.Vitamin C
    [46] => Data.Vitamins.Vitamin E
    [47] => Data.Vitamins.Vitamin K
)
```

Toutes les colonnes.