



IFOSUP (Webdev, 5XSE2) { 2023-2024 }

congé de Détente... ou pas ! ha ha !

Pour rester en forme durant le congé, il est conseillé de
coder un minimum de 30 minutes par jour !

C'est pourquoi je vous ai concocté **quelques exercices !**

Faites ça avec amour, car à la rentrée, chaque exercice sera
corrigé par l'un d'entre vous suite à une sélection aléatoire !

En fonction de la qualité de mon congé,
il se pourrait que cette **prestation** soit **côtée !**

Plat Principal

Terminez les deux premières parties du projet à remettre en fin de parcours :

- Modèles de pages dynamiques
- Gestion des formulaires

Dessert

Réalisez les fonctions en suivant les informations qui leur sont associées. Ensuite testez les dans le terminal :

1

Objectif : Vérifier si le fichier correspondant au chemin passé en argument existe. Si c'est le cas, demander la permission d'écraser.

Si le fichier n'existe pas encore ou si l'utilisateur a accepté d'écraser le fichier, la fonction retourne "true", sinon elle retourne "false".

```
function ecraser_fichier(string $cheminFichier): bool
```

2

Objectif : Créer un nouveau dossier correspondant au chemin passé en argument si celui-ci n'existe pas encore. Accessoirement, offrir la possibilité de créer les dossiers intermédiaires si nécessaire. Si un dossier a été créé, la fonction retourne "true", sinon elle retourne "false".

```
function creer_dossier(string $cheminDossier): bool
```

3

Objectif : Convertir le contenu d'un tableau au format JSON et enregistrer le résultat dans un fichier. Utilisez la fonction "ecrire_fichier()" pour gérer l'éventuelle présence d'un fichier portant déjà ce nom. Si l'enregistrement a été réalisé, la fonction retourne "true", sinon elle retourne "false".

```
function enregistrer_tableauEnFichierJson(string $cheminFichier, array $contenu): bool
```

4

Objectif : Créer une liste, en invitant l'utilisateur à entrer, un par un, les éléments qu'il souhaite inclure dans celle-ci. Chaque élément sera donc représenté par une chaîne de caractères saisie par l'utilisateur. L'utilisateur devra entrer un mot-clé pour préciser au programme que la construction de la liste est terminée. La fonction retournera le résultat la liste sous forme d'un tableau classique (clés numériques).

```
function creer_liste(): array
```

5

Objectif : Importer le contenu du fichier JSON dont le chemin est passé en premier argument de la fonction et retourner la conversion sous forme de tableau.

```
function importer_fichierJsonContenu(string $cheminFichier): array
```

6

Objectif : Afficher le contenu d'une liste à l'aide d'une boucle (on oublie la fonction "print_r()". La liste devra être importée à l'aide de la fonction "importer_fichierJsonContenu()".

```
function afficher_listeContenu(string $cheminFichier): void
```

7

Objectif : Duplique un fichier existant et demande le nom désiré pour le fichier dupliqué. Utilisez la fonction "ecrire_fichier()" pour gérer l'éventuelle présence d'un fichier portant déjà ce nom. Si le second argument, qui est facultatif, est passé à la fonction, et que le fichier de destination existe déjà, le fichier sera automatiquement écraser sans demander la permission à l'utilisateur. Si la duplication a été réalisée, la fonction retourne "true", sinon elle retourne "false".

```
function dupliquer_fichier(string $cheminFichier, ?string $cheminFichierDuplique = null): bool
```

8

Objectif : Récupérer un élément de façon aléatoire d'une liste situé dans un fichier JSON correspondant au chemin précisé en premier argument. Le deuxième argument offre la possibilité à l'utilisateur de ne pas accepter les doublons. C'est à dire que tous les éléments de la liste doivent avoir été sélectionnés avant qu'un même élément ne puisse être sélectionné une deuxième fois. L'utilisation de la fonction "dupliquer_fichier()" pourrait être une solution pour mettre en place cette fonctionnalité.

```
function recuperer_elemAleatoire(string $cheminFichier, ?bool $estPersistant = false): string
```

9

Objectif : Récupérer un groupe généré de façon aléatoire à partir d'une liste situé dans un fichier JSON correspondant au chemin précisé en premier argument. Le deuxième argument précise le nombre d'élément désiré par groupe. La liste devra être importée à l'aide de la fonction "importer_fichierJsonContenu()". La fonction retourne la liste des groupes sous forme de tableau.

```
function recuperer_groupeAleatoire(string $cheminFichier, int $nbrMembreParGroupe): array
```