



Figure I.1 - Représentation de votre Lib_ACS (vision d'artiste)

Lib_ACS

Ta propre bibliothèque à toi tout seul

Résumé : Ce projet a pour but de vous faire coder une bibliothèque de fonctions usuelles.

Table des matières :

[I - Préambule](#)

[II - Introduction](#)

[III - Consignes générales](#)

[IV - Partie obligatoire](#)

IV.1 - Considérations techniques

IV.2 - Part1 - Fonctions de la lib

IV.2 - Part2 - Fonctions supplémentaires

[V - Partie Bonus](#)

[VI - Rendu](#)

I - Préambule



Figure I.2 - La où je serais pendant ce projet

Ce projet a pour but de vous occuper pendant les vacances, parce que soyons honnêtes, je vais vous manquer. Et aussi de vous familiariser avec les fonctions. Peu importe le langage, les fonctions sont indispensables.

Je vous propose un accompagnement musical afin de mener au mieux votre projet. Si vous n'aimez pas, c'est que visiblement vous avez des goûts musicaux pauvres, mais vous avez probablement d'autres qualités comme avoir beaucoup d'amis sur Facebook ou bien toucher votre coude avec votre langue. Bref. Voici une liste non exhaustive :

- [accompagnement musical 1](#)
- [accompagnement musical 2](#)
- [accompagnement musical 3](#)
- [accompagnement musical 4](#)
- [accompagnement musical 5](#)

II - Introduction

Le projet ACS_Lib a pour but de vous faire écrire une bibliothèque de fonctions utiles en PHP que vous pourrez utiliser dans la majorité de vos projets (sauf fonctions déjà existantes bien sûr). Vous aurez pas mal de code à écrire, ce qui en fait un excellent projet pour consolider vos compétences et faire le point sur votre avancement.



Figure II.1 Citation de Gandhi (ou pas).

III - Consignes générales

- Les fonctions sont à réalisées dans l'ordre que vous le souhaitez et vous pouvez tout à fait utiliser les fonctions déjà codées pour réaliser les suivantes. La difficulté n'est pas croissante et l'ordre tout à fait arbitraire. Un peu comme dans un jeu vidéo où vous pouvez réaliser les quêtes dans l'ordre que vous voulez et utiliser le loot des précédentes pour faciliter les suivantes.
- Vous devez respecter une norme de nommage et de codage :
 - Tous les noms des fonctions doivent commencer par `fn_`
 - Tous les noms doivent être en anglais.
 - Utiliser le camelCase pour le noms des variables si besoin.
 - Le nom de vos arguments doit être cohérent, ex: si votre fonction prend en paramètre une chaîne de caractère alors le paramètre sera `$str`.
 - N'utilisez pas de noms "interdits" pour vos variables.
 - Indenter et commenter correctement votre code.
 - Il ne doit pas y avoir d'écho, `var_dump()` ou `print_r()` dans vos fonctions. Utiliser des **return**.
- Vous devez réaliser un README propre à la racine de ce projet.
- Un fichier php par fonction.
- Vous avez le droit d'utiliser uniquement native isset.
- Vos fonctions doivent être protégées, vous devez gérer les erreurs. Leur comportement doit être le même que les fonctions natives de php. Aussi je vous indique à bien lire le manuel.

IV - Partie obligatoire

IV.1 Considérations techniques

- Une chaîne de caractère peut être utilisée comme un tableau.
- Interdiction d'utiliser les variables globales.

IV.2 Part1 - Fonctions de la lib

Vous devez recoder les fonctions suivantes :

- strlen
- count
- empty
- sort
- intval
- strtolower
- strtoupper
- trim
- strstr
- strcmp
- array_diff
- str_replace

IV.3 Part2 - Fonctions supplémentaires

Si vous avez besoin d'autres fonctions pour coder les précédentes, libre à vous de les créer, et de les ajouter à votre lib. Plus vous étoffez votre lib, plus elle sera utile.

V - Partie bonus

Si vous avez parfaitement finie la partie obligatoire, cette section propose quelques pistes pour aller plus loin. Un peu comme quand vous achetez un DLC pour un jeu vidéo.

Voici d'autres fonctions à recoder :

- explode
- implode
- ltrim
- rtrim
- str_rot13
- str_shuffle
- strip
- substr

VI - Rendu

Votre travail doit être régulièrement pusher sur git. Seul le travail présent sur votre dépôt comptera.

Bon courage, et n'oubliez pas de faire des beaux commits et un beau README!