



W1 - Piscine PHP

W-WEB-024

Rush 2

my_tar

Rush 2

repository name: rush2
language: PHP



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

COMPÉTENCES à ACQUÉRIR

- PHP
- Bash
- Systèmes de compression

Bienvenue dans ce second Rush de piscine ! :)

Ce rush va vous apprendre comment sont générées les archives “zip”, “tar”, “tgz”, “7z”, etc.

CONSIDÉRATIONS

Pour ce rush, vous êtes libre de travailler comme vous l'entendez, à distance, en home office, ou en groupe au même endroit ; en fonction des consignes données par votre équipe pédagogique. Dans tous les cas, vous **DEVEZ** mettre en place une organisation de groupe et une gestion de projet.

DÉROULÉ DE LA SOUTENANCE

- Vous devez être acteur de votre soutenance !
- Vous devez commencer la soutenance en expliquant ce que vous avez mis en place pour réaliser le projet (rétro-planning, découpage, difficultés rencontrées, tâches non terminées, etc) avec un support à l'appui (git, trello, gantt, uml, doc, pdf, etc.) [5 min]
- Vous devez ensuite présenter toutes les fonctionnalités que vous aurez implémenté en suivant le fil conducteur du sujet. [15 min]



Tout ce qui n'est pas montré en soutenance n'est pas noté.

- Enfin il pourra vous être demandé de préciser certains aspects du projet ou d'effectuer un recode de tout ou partie de celui-ci. [5 min]
- Il vous sera enfin fait un retour sur la qualité globale de votre rendu et sur le déroulé de la soutenance. [5 min]



Afin de devenir de bons développeurs il vous faut être rigoureux et méthodique. Cela s'applique aussi à la qualité du code que vous allez produire. Pour cela, vous devez respecter des normes de code ! Pour le PHP, par exemple, vous devez respecter la norme **PSR-12**.



Vous pouvez vérifier la norme de vos projets PHP avec PHP_CodeSniffer : https://github.com/squizlabs/PHP_CodeSniffer



Vous pouvez vérifier la norme de vos pages HTML sur le site du W3C : <https://validator.w3.org/>



Toute erreur de norme entraînera un malus conséquent sur votre note finale.



Toute erreur d'un script PHP, ou JS, sera sanctionné par des points malus.

ETAPE 1

Rendu : rush2_step1/

Écrire un programme exécutable `my_tar.php` qui prend en ligne de commande, un ou plusieurs arguments, qui sont des noms de fichiers ou dossiers à archiver. Le programme génère en sortie une tarball nommée `output.mytar` (et l'écrase si elle existe déjà).

```
Terminal
~/W-WEB-024> ls
file1 file2 file3 folder1 my_tar.php
~/W-WEB-024> php my_tar.php file1 file2 folder1
~/W-WEB-024> ls
file1 file2 folder1 my_tar.php output.mytar
~/W-WEB-024>
```

Il faudra bien entendu gérer le maximum de formats de fichiers (txt, avi, png, mp3, etc.).



Vous devez gérer la récursivité. Il n'est pas impossible qu'un dossier contienne lui-même d'autres dossiers et ainsi de suite. La commande `tree` peut rapidement vous apporter un aperçu.



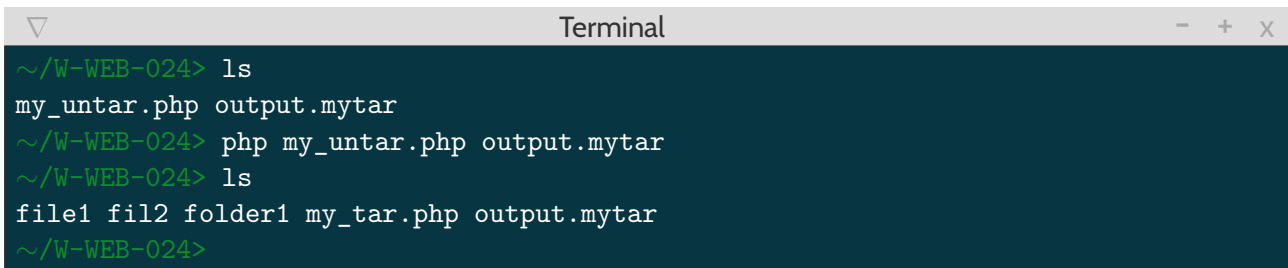
Il est strictement interdit de reprendre une fonction qui génère une archive. Tout l'intérêt de ce sujet étant de recréer soit même les fonctionnalités. Si vous avez un doute sur l'autorisation ou l'interdiction des fonctions que vous utilisez, demandez aux assistants.

ETAPE 2

Rendu : rush2_step2/

Écrire un programme exécutable `my_untar.php` qui prend en ligne de commande, un ou plusieurs arguments, qui sont des tarball dont il faut extraire les données. Le programme génère en sortie un ou plusieurs fichiers ou dossiers. En cas de conflit, lorsqu'un fichier ou dossier existe déjà, un prompt doit proposer plusieurs choix :

1. Écraser
2. Ne pas écraser
3. Écraser pour tous (ne plus redemander)
4. Ne pas écraser pour tous (ne plus redemander)
5. Arrêter et quitter

A terminal window titled "Terminal" with a dark green background. It shows the following commands and output:

```
~/W-WEB-024> ls
my_untar.php  output.mytar
~/W-WEB-024> php my_untar.php output.mytar
~/W-WEB-024> ls
file1 file2 folder1 my_tar.php output.mytar
~/W-WEB-024>
```

ETAPE 3: INTERFACE WEB

Rendu : rush2_step3/

Dans cette partie, vous réaliserez une interface web qui aura pour but de créer une archive depuis votre navigateur.

Cette interface web doit contenir :

- Un champ pour définir le nom de l'archive.
- Une fonction pour ajouter des fichiers (drag and drop, formulaire, etc.).
- Une liste de tous les fichiers ajoutés.
- Un bouton pour générer l'archive.
- Un bouton pour télécharger l'archive.



Inspirez-vous de l'interface ci-dessous. Vous êtes libre d'ajouter des éléments de style ou des animations.



Nous vous conseillons fortement de passer par un serveur apache avec PHP pour utiliser votre algorithme de compression.

Nom de l'archive:

Ajouter un fichier

Fichiers ajoutés

| |
|-----------|
| tar.php |
| test.php |
| image.png |
| audio.mp3 |
| ... |

Générer un fichier tar

Télécharger l'archive



BONUS

Rendu : rush2_bonus/

Meilleur sera le taux de compression de votre archive, plus votre Rush sera réussi ! Vous devez à présent commencer à réfléchir aux différentes façons d'optimiser la taille de votre archive et la vitesse de compression.

Pour chaque amélioration apportée à ce processus et présentée en soutenance, des points bonus vous seront accordés.



Définition de la "compression" : https://fr.wikipedia.org/wiki/Compression_de_donn%C3%A9es
Comparaison des compressions : <http://rlwp.free.fr/WPFF/comploc.htm>