Exercice 1 :
Consultez la documentation PHP pour comprendre les fonctions

- `trim()` qui permet de "supprimer" les espaces en début et fin de chaîne.
- `str_replace()` qui permet de remplacer une sous-chaîne par une autre
- `explode()` qui permet de découper une chaîne de caractères selon un séparateur.
- `implode()` la réciproque de la précédente

Q 1 . Définissez une fonction PHP à un argument (chaîne de caractère). Cette chaîne peut contenir des '+' qui joueront le rôle de séparateurs de la chaîne en plusieurs parties. Le résultat de la fonction est le code HTML représentant une suite de paragraphes, chaque paragraphe contient l'une des parties de la chaîne, selon la position des '+'. Les espaces éventuels en début ou fin de chaque partie sont à éliminer.
Par exemple si l'argument est `"Et qu'on sorte+  Vistement : +Car Clément + Le vous mande."` le résultat sera :
`<p>Et qu'on sorte</p><p>Vistement :</p><p>Car Clément</p><p>Le vous mande.</p>`
Testez cette fonction sur plusieurs exemples.
NB : il faut utiliser les 4 fonctions présentées plus haut !
Q 2 . Construire une fonction PHP `enSpan($s)` analogue à la précedente. Cette fois le séparateur est le signe '-' **précédé et suivi d'un espace**. Le résultat est cette fois une liste d'éléments `<span>`
Par exemple si `$s` vaut `Dupont - Durand`, le résultat vaudra `<span>Dupont</span><span>Durand</span>`

Lecture et écriture de fichiers en PHP

Ouverture d'un fichier

Avant de réaliser des opérations sur un fichier, il est nécessaire de l'ouvrir avec la fonction `fopen` dont la syntaxe (simplifiée) est :

```
resource fopen(string $nom, string mode)
```

Le résultat de `fopen` est une « ressource » (un genre d'identifiant du fichier ouvert) qu'il faut impérativement mémoriser pour pouvoir la passer ensuite aux fonctions de lecture ou d'écriture. Le résultat vaut `FALSE` en cas d'échec.
Le premier paramètre de `fopen` est le nom du fichier que l'on souhaite utiliser (éventuellement avec son chemin). Le second paramètre détermine le mode d'accès au fichier. Voici les principaux :

- mode `"r"` : le fichier est ouvert en lecture seule. La lecture commence au début du fichier.
- mode `"w"` : le fichier est ouvert en écriture seule. Si le fichier existait, il est préalablement remis à 0. S'il n'existait pas, il est créé.
- mode `"a"` : le fichier est ouvert en écriture seule, il **n'est pas** remis à 0, et les données écrites sont ajoutées en fin de fichier. Si le fichier n'existait pas, il est créé.

(NB : d'autres modes existent, à lire dans la documentation PHP)

Lecture à partir d'un fichier, ligne par ligne

```
string fgets(resource $id_file)
```

Cette fonction lit une ligne du fichier. Si la lecture s'est bien déroulée, le résultat de la fonction est une chaîne contenant la ligne lue. Si une erreur survient (par exemple la fin du fichier), `fgets` renvoie le booléen `FALSE`.

Écriture dans un fichier

```
integer fputs(resource $id_file, string chaine)
```

Écrit la chaîne passée en second paramètre dans le fichier d'identifiant `$id_file`. L'entier renvoyé par cette fonction est le nombre d'octets écrits, ou `FALSE` en cas d'erreur.

Fermeture d'un fichier

Quand on a fini d'utiliser un fichier, il faut le fermer à l'aide de la fonction `fclose` :

```
fclose(resource $id_file)
```

Remarque : Pour les opérations de lecture et d'écriture, il faut que le serveur ait les droits nécessaires à la réalisation de ces opérations au niveau du système de fichiers.

Exemple

Le petit programme ci-dessous lit un fichier de texte dans son ensemble et affiche le nombre de lignes lues et leur longueur moyenne

```
<?php
$file = fopen('mon_fichier.txt','r'); //ouverture du fichier
$ligne = fgets($file); //tentative de lecture
$cpt = 0; $sommeLongueurs = 0;
while ($ligne !== FALSE) { // une ligne a vraiment été lue
    $cpt++;
    $sommeLongueurs += strlen($ligne);
    $ligne = fgets($file); // tentative de lecture de la ligne suivante
}
fclose($file);
if ($cpt==0)
  echo "<p>Fichier vide</p>";
else {
  echo "<p>$cpt lignes lues. Longueur moyenne : " . $sommeLongueurs/$cpt ."</p>";
}
?>
```

Exercice 2 :
(Exercice d'application immédiate)
Créez tout d'abord sur le serveur webtp un dossier nommé `seance2`. Par la suite vous réaliserez tous les exercices de la séance d'aujourd'hui dans ce dossier.
Transférez dans ce dossier le fichier `liste_noms.txt` qui vous est fourni. Ce fichiers contient des noms, à raison d'un par ligne.
Concevez une programme PHP qui lit la totalité de ce fichier et construit une page web contenant la liste de tous les noms figurant dans le fichier (chaque nom sera un item de la liste).

Exercice 3 :
Un terrain de jeu de forme carrée est composé de cases qui peuvent recevoir chacune un pion noir ou un pion blanc (on peut penser au jeu de Dames, par exemple, mais nos terrains auront des côtés de longueur quelconque).
Un terrain sera présenté en HTML par une `table` avec la convention suivante :

- Une cellule qui contient un pion blanc contiendra la lettre B. La cellule possèdera la classe « `blanc` »
- Une cellule qui contient un pion noir contiendra la lettre N. La cellule possèdera la classe « `noir` »
- Une cellule vide ne contient rien et n'appartient ni à la classe « `blanc` » ni à la classe « `noir` »

NB : dans les cellules, chaque lettre B et ou N sera placée dans un élément `span`.

On suppose disposer d'un fichier contenant la description d'un terrain sous forme de texte. Chaque ligne du terrain ets représentée par une ligne de texte. Chaque case est codée par un caractère : « `-` » (pour une case vide), « `B` » ou « `N` ». Voici un exemple de terrain 5×5 :

```
-B--N
B-NN-
-----
NN-BB
NBNBN
```

Les données sont donc correctes si toutes les lignes ont la même longueur et si le nombre de lignes est égal à cette longueur (on pourra néanmoins trouver une ou plusieurs lignes vides après la description du terrain).

Q 1 . Écrire un programme PHP qui produit une page HTML affichant le terrain défini par le fichier :

1. Dans un premier temps, supposez les données correctes, sans les vérifier.

2. Puis vous testerez les cas d'erreurs : lignes de longueurs différentes, terrain non carré.
   En cas d'erreur des données vous ferez en sorte de ne pas afficher la table mais un message d'erreur. Pour ce faire, vous ne pourrez pas "afficher" les lignes au fur et à mesure de la lecture des données. Vous pourrez, par exemple, construire le code de la table HTML dans une chaîne et n'afficher cette chaîne qu'à la fin de la lecture, s'il n'y a pas eu d'erreurs.

Q 2 . CSS

Associer un style pour faire en sorte que les cases du terrain soient vraiment carrées et qu'elles aient une bordure. Les lettres ne seront pas affichées, mais les cases apparaîtront blanches, noires ou grisées (pour les cases vides)

Q 3 . JS (à faire en dehors de la séance)

Associer un fichier javascript pour rendre le terrain « interactif ».

1. Un clic sur une case vide y mettra un pion blanc

2. Un clic sur une case blanche y mettra un pion noir

3. Un clic sur une case noire la videra