

Résumé des fonctionnalités vues dans l'exercice 4

F1 : Valider si chaîne est une phrase ou pas.

Une phrase commence par une lettre majuscule et se termine par un point.

La travail à faire était de tester si une phrase est valide ou pas tout en sachant qu'une phrase valide commence par une lettre majuscule et se termine par un point (. ou ? ou !) donc :

1. La chaîne doit commencer par une lettre alphabétique comprise entre A et Z ce qui se traduit par `^[A-Z]`
2. Ce qui suit doit être n'importe quel caractère minuscule, majuscule, virgule, chiffre, un espace... mais pas de point (. ou ? ou !) une ou plusieurs(+) fois se qui se traduit par `[^ . !?]+`
3. Et enfin la phrase doit se terminer par un point simple, un point d'exclamation ou d'interrogation ce qui se traduit par `[. !?]`

Donne l'expression régulière finale est `#^[A-Z][^ . !?]+[. !?]`

Les `#` sont les délimiteurs de l'expression.

4. La fonction nous permettant de voir si une phrase est valide ou pas est donc :

```
function test_Phrase($phrase){
    $regex = '#^[A-Z][^ . !?]+[. !?]' ;
    if(preg_match($regex,$phrase)){
        return true;
    }
    return false;
}
```

Elle retourne true(1) si la phrase est valide sinon false (rien).

F2 : Découper un texte en phrase.

Pour découper un texte en phrases j'ai utilisé une fonction prédéfinie de php qui le `preg_match_all`. Cette dernière dont la syntaxe est : `preg_match_all($regex, $chaîne, $tableau)`. Elle analyse **chaîne** pour trouver l'expression **regex** et met le **resultat** dans tableau. Si **regex** ne se trouve nul part dans **chaîne** la fonction renvoie false.

J'ai fait donc :

```
preg_match_all('#[A-Za-z][^ . !?]+[. !?]', $texte, $texteCoup);
```

Dans ce cas `$regex= '[A-Za-z][^ . !?]+[. !?]`

Ce qui veut dire que dans le texte `$texte` si je vois une chaîne qui commence par une lettre majuscule entre A et Z suivie de tout caractère autre que . ! et ? ou d'un point(.) suivi d'un chiffre et se

terminant par . ! ou ? je considère cette chaîne comme une phrase et je le mets dans le tableau **texteCoup** qui va regrouper l'ensemble des phrases contenues dans le texte.

F3 : Enlever les espaces inutiles d'une Phrase.

Pour supprimer les espaces inutiles dans une phrase j'ai procédé comme suit :

1. Supprimer les espaces en début et en fin d'une phrase
Pour cela j'ai utilisé la fonction prédéfinie de php **trim** en lui donnant en argument une phrase **\$phrase : trim(\$phrase)**
2. Supprimer ou mettre des espaces après (ou/et avant) certains caractères dans le respect des règles d'orthographe
Dans ce cas j'ai utilisé une autre fonction prédéfinie de php qui le **preg_replace**.
Celle dont la syntaxe est la suivante : **preg_replace(\$regex, \$remplacant, \$chaîne)**
analyse la chaîne **chaîne** pour trouver l'expression **regex** et le remplace par **remplacant**.
Pour faire le travail visé j'ai utilisé ces expressions qui vont jouer chacune le rôle de **regex**

```
a) #[ ]+# => Cette expression représente un ou plusieurs
    espaces sera remplacée par une seule espace
b) [ ]?[\'] [ ]?# => un apostrophe entouré de 0 ou plusieurs
    espaces sera remplacé par un apostrophe
c) #[ ]?[ , ]# => une , précédée de 0 ou plusieurs espaces sera
    remplacée par une , suivi d'un espace
d) #[ ]?[:;][ ]?# => un ; précédé de 0 ou plusieurs espaces
    sera remplacé par un ; suivi d'un espace
e) #[ ]+[. ]# => un . précédé par un ou plusieurs espaces sera
    remplacé par un .
f) #[ ]+[? ]# => un ? précédé par un ou plusieurs espaces sera
    remplacé par un ?
g) #[ ]+[! ]# => un ! précédé par un ou plusieurs espaces sera
    remplacé par un !
h) #[ ]+[) ]# => une ) précédée par un ou plusieurs espaces
    sera remplacée par )
i) #[ ( [ ]+# => un ( suivie d'un ou de plusieurs espaces sera
    remplacée par un (
```

J'ai de ce fait créé deux tableaux **\$regex** qui contient les expressions et **\$remplacant** qui contient le remplaçant de chaque expressions. A chaque indice **i** de **\$regex** il y'a son remplaçant dans **\$remplacant** situé à cet même indice.

En regroupant tous les points évoqués ci-dessous l'exercice 4 donne :

```

//fonction pour decouper un texte en phrases
function splitPhrase($texte){
    //le regex signifie tt ce qui commence par une lettre minuscule ou majuscule suivi
    //de tt autre caractere different de . ? et ! ou de . suivi de chiffre 0 ou plusieurs fois
    //termine par . ! ou ?
    $tableau = [];
    preg_match_all('#[A-Za-z][^.!?][.][0-9])*[.!?]#',$texte,$texteCoup);
    $lesPhrases = $texteCoup[0];
    foreach($lesPhrases as $phrase){
        $phrase=trim($phrase);
        //les expressions qu'on doit remplacer
        $regex=['#[ ]+#', '#[ ]?[\']?[ ]?#', '#[ ]?[, ]#', '#[ ]?[:;][ ]?#', '#[<][ ]?#', '#[ ]?>[ ]?#',
            '#[ ]?[. ]#', '#[ ]?[?]#', '#[ ]?[! ]#'];
        //leurs remplacements
        $remplace=[' ', "'", '"', ";", "(", ")", "|", ".", "?", "!"];
        for($j=0;$j<laTaille($regex);$j++){
            $phrase = preg_replace($regex[$j],$remplace[$j],$phrase);
        }
        $phrase=ucfirst($phrase);
        if(laTaille($phrase)<=200){
            $tableau[]=$phrase;
        }
    }
    return $tableau;
}

```