

**Séance 1a:****accès au webtp:**<http://webtp.fil.univ-lille1.fr/~benzina/dossier/fichier>**date\_default\_timezone\_set('Europe/Paris');****date('d/m/Y');** affiche la date (ici Y pour année sur 4 chiffres).**PHP\_VERSION:** cte qui range la version de php**PHP\_OS:** la version du système d'exploitation du serveur**strlen(\$string): int :** renvoie la longueur de la chaîne**substr(\$string, \$start, [\$length]): string:**

Retourne le segment de string défini par start et length (combien de caractère)

```

$rest = substr("abcdef", -1);
// retourne "f"
$rest = substr("abcdef", -2);
// retourne "ef"
$rest = substr("abcdef", -3, 1);
// retourne "d"
$rest = substr("abcdef", 0, -1);
// retourne "abcde"
$rest = substr("abcdef", 2, -1);
// retourne "cde"
$rest = substr("abcdef", 4, -4);
// retourne false
$rest = substr("abcdef", -3, -1);
// retourne "de"
echo substr('abcdef', 0, 8); // abcdef
echo substr('abcdef', -1, 1); // f

```

**substr(\$str,0,\$i):** retourne les \$i premier caractère de \$str**<table>:** pour commencer une table!!**<td>:** pour une cellule**<tr>:** pour une nouvelle ligne**<th>:** pour une cellule d'en-tête (en gras)**Séance 1b:****trim(\$str):** supprime les espaces en début et fin de chaîne. (\n, \t, " "...)**str\_replace(\$remplacer, le remplacement, dans):**

```

// Génère : <body text='black'>
$bodytag = str_replace("%body%",
"black", "<body text='%body%'>");

```

```

// Génère : Hll Wrld f PHP
$vowels = array("a", "e", "i", "o",
"u", "A", "E", "I", "O", "U");
$onlyconsonants =
str_replace($vowels, "", "Hello World
of PHP");

```

```

// Génère : You should eat pizza, beer,
and ice cream every day
$phrase = "You should eat fruits,
vegetables, and fiber every day.";
$healthy = array("fruits", "vegetables",
"fiber");
$yummy = array("pizza", "beer", "ice
cream");
$newphrase = str_replace($healthy,
$yummy, $phrase);
// Affiche F car A est remplacé par B,
puis B est remplacé par C, et ainsi de
suite...
// Finalement, E est remplacé par F
$search = array('A', 'B', 'C', 'D',
'E');
$replace = array('B', 'C', 'D', 'E',
'F');
$subject = 'A';
echo str_replace($search, $replace,
$subject);

```

**explode(séparateur, chaîne):**  
**table****implode(\$glue, \$pieces(array)) : str**

## NOTES TPS TW2

### Ouverture d'un fichier:

**\$file = fopen(\$fichier, \$mode)**  
retourne FALSE en cas d'échec. (si le fichier n'est pas trouvé)

#### les modes:

- 'r': lecture seule (dès le début)  
- 'w': écriture seule. Si le fichier existe il est préalablement remis à 0 sinon il est créé  
- 'a': écriture seule. Il n'est pas remis à 0 et les données sont ajoutées en fin de fichier (si il n'existe pas il est créé)

### lecture d'un fichier:

**\$ligne = fgets(\$file)**

⇒ renvoie FALSE si c la fin du fichier

### Ecriture dans un fichier:

**\$c = fputs(\$file, \$chaine):**

écrit la chaîne passée en second paramètre dans \$file.  
ça renvoie le nombre d'octets écrits ou FALSE en cas d'erreur

### Fermeture d'un fichier:

**fclose (\$file);**

#### EX:

```
$file = fopen('fichier.txt','r');
$ligne = fgets($file);
echo "<ul>";
while ($ligne!==FALSE){
    echo "<li>".$ligne."</li>\n";
    $ligne = fgets($file);
}
fclose($file);
echo "</ul>";
```

### Exception:

**throw new Exception("msg");**

#### RQ:

-Pour lire une ligne il faut la trimmer.

⇒

-Pour éviter les lignes vides à la fin du fichier on fait:

```
while ($ligne!==FALSE &&
$ligne!="")
```

⇒ Si je lis déjà une ligne le \$cptLignes=1 et la fin de la boucle while, je lis une ligne et j'incrémente \$cptLignes.

### Feuille2:

#### les fichiers:

-**couverture**: contient les images  
-**data**: contient les données (les fichiers) sur lesquels on va travailler

-**lib**: contient les fichiers des classes ou fonctions php.

-**views**: contient le fichiers HTML et un seul élt php dans lequel il y a un echo de la variable finale à afficher

-**dehors**: le(s) fichier(s) qui va générer la page

1. inclusion de la bibliothèque de fonction
2. lecture des fichier
3. mémorisation du traitement du fichier dans des variable
4. inclusion de la page template qui est dans **views**

**strpos(\$string, \$chaineAChercher):**

```
$mystring = 'abc';
$findme = 'a';
$pos = strpos($mystring,
$findme);
```

retourne l'entier de la position FALSE si pas trouvé

#### EX:

```
$pos = strpos($line, ":");
$name =
trim(substr($ligne,0,$pos);
$value =
trim(substr($line,$pos+1);
```

## NOTES TPS TW2

**print\_r(..):** pour le débogage

**le fichier .htaccess:**

Order deny, allow  
Deny from all

⇒ dans le fichier **data,lib,views**

⇒ **couvertures** doit rester accessible: il contient des images faisant partie des page web du site.

```
function readBook($file){
    $str = fgets($file);
    if ($str==FALSE)
        return FALSE;
    else {
        $i=0;
        while ($str!=FALSE) {
            if (trim($str)==""){
                $str=fgets($file);
                $i+=1;
                $found = FALSE;
            }
            else {
                $pos = strpos($str, ":");
                if ($pos==FALSE){
                    throw new Exception("La
description de ligne est
incorrecte");
                }
                else{
                    $tabLigne =
explode(":", $str);
                    $propriete[$i][trim($tabLigne[0])
]=trim($tabLigne[1]);
                    $str = fgets($file);
                    $found = TRUE;
                }
            }
        }
        if (!$found)
            return $found;
        else
            return $propriete;
    }
}
```

### Feuille 3:

**Une fonction de comparaison**

indique quelle est la petite valeur:

fcomp(\$x,\$y)<0, >0, ==0

**usort(\$tableau,**  
**"nom\_de\_fct\_de\_comp");**

function compareAbs(\$x, \$y){

\$cmp = abs(\$x)-abs(\$y);

if (\$cmp != 0)

return \$cmp

else

return \$x - \$y;

}

⇒ retourne la difference de valeur absolue si ils sont pas égaux et la valeur la plus petite (en nég/pos) si ils sont égaux.

### RQ:

Pour trier les tables associatives il existe les fonctions **uksort** et **uasort**.

**strcmp():** est une fonction de comparaison de chaînes de caractères, selon l'ordre lexicographique(même resultat qu'avec la fonction sort. (sort et usort sont deux fonctions différentes).