

Fonctions utiles

TP 03 du Module 02 – Les bases du langage PHP

Proposition de solution

- Exercice 1

```
<?php
$chaine = 'bOnJoUr, jE sUiS UnE pHrAsE pOuR tEsTeR.';
echo ucwords(strtolower($chaine)) . '<br>;
```

La fonction `strtolower()` passe toute la chaîne en minuscules. La fonction `ucwords()` passe la première lettre en majuscule. Attention, il ne faut pas se contenter seulement d'appeler `ucwords()` car elle ne modifie pas la casse des autres lettres.

Exécution

Bonjour, Je Suis Une Phrase Pour Tester.

- Exercice 2

```
<?php
echo '<pre style="font-family: courier">';
$personnes = array(array('Norbert', 'DURAND'), array('Melanie', 'MALALANICHE'),
array('Juste', 'LE BLANC'));
foreach ($personnes as $p) {
    printf("%-20s %-20s<br>", $p[0], $p[1]);
}
echo '</pre>;
```

La fonction `printf()` permet de réaliser un affichage formaté à l'instar d'autres langages de programmation. Le premier paramètre de cette fonction est une chaîne de caractères contenant des « trous » qui seront remplacés par les valeurs des arguments suivants. Ces trous sont matérialisés par le caractère `%`.

La lettre `s` indique que la valeur qui sera insérée dans ce trou sera considérée en tant que chaîne de caractères.

Le nombre 20 indique le nombre de caractères à utiliser pour afficher cette valeur.

Le signe moins indique que l'alignement des caractères se fait à gauche (le reste de la place disponible est complété avec des espaces).

Un problème est que le langage HTML ne prend pas en compte le fait qu'il y ait plusieurs espaces. La page s'affichera avec un seul espace à moins que ce texte soit dans une balise `<pre>`. Pour que chaque caractère utilise la même largeur, il faut utiliser une police à taille fixe telle que la police Courier.

Exécution

Norbert	DURAND
Melanie	MALALANICHE
Juste	LE BLANC

- Exercice 3

```
<?php
$listeCours = [
    '10-Développement Web côté serveur avec PHP et Symfony',
    '15-Conduite et gestion de projet',
    '8-Analyse et Conception',
    '5-PL_SQL avec Oracle',
    '6-Développement en couches avec Java SE',
    '13-Développement Web côté serveur avec JEE',
    '19-Développement d\'une application mobile avec Android',
    '20-Développement Android - Projet',
    '14-Projet 2 - Application Web',
    '2-Initiation à la programmation procédurale avec Java',
    '18-Développement Web côté serveur en ASPX',
    '9-Développement Web côté client',
    '21-Développement cross plateforme avec Xamarin',
    '7-Projet1-ClientServeur-JavaSE',
    '1-Algorithmique',
    '11-Integration Graphique',
    '16-Utiliser les frameworks pour le développement JEE',
    '12 - CMS - WordPress',
    '4-SQL avec SQL Server',
    '3-POO avec Java',
    '17-Linux et Admin Apache'
];

natsort($listeCours);
echo '<ul>';
foreach ($listeCours as $cours) {
    echo '<li>'.$cours.'</li>';
}
echo '</ul>';
```

La fonction `natsort()` effectue ce tri naturel.

Exécution

- 1-Algorithmique
- 2-Initiation à la programmation procédurale avec Java
- 3-POO avec Java
- 4-SQL avec SQL Server
- 5-PL_SQL avec Oracle
- 6-Développement en couches avec Java SE
- 7-Projet1-ClientServeur-JavaSE
- 8-Analyse et Conception
- 9-Développement Web côté client
- 10-Développement Web côté serveur avec PHP et Symfony
- 11-Integration Graphique
- 12 - CMS - WordPress
- 13-Développement Web côté serveur avec JEE
- 14-Projet 2 - Application Web
- 15-Conduite et gestion de projet
- 16-Utiliser les frameworks pour le développement JEE
- 17-Linux et Admin Apache
- 18-Développement Web côté serveur en ASPX
- 19-Développement d'une application mobile avec Android
- 20-Développement Android - Projet
- 21-Développement cross plateforme avec Xamarin

- Exercice 4

```
<?php
$maintenant = time();
$naissance = mktime(23, 15, 00, 6, 2, 1982);
echo number_format($maintenant - $naissance, 0, '.', ' ') . ' secondes vécues';
```

La fonction `time()` crée un timestamp UNIX initialisé à la date actuelle.

La fonction `mktime()` crée un timestamp UNIX initialisé à la date passée en paramètre. Attention, l'ordre des paramètres de cette fonction est un peu déroutant pour un non-anglosaxon : c'est d'abord l'heure puis les minutes puis les secondes puis le **mois** puis le **jour** et enfin l'année.

Un timestamp UNIX est le nombre de secondes écoulées entre le 1^{er} Janvier 1970 et la date. En faisant la différence entre les deux timestamps, le nombre de secondes écoulées entre les dates est obtenu.

La fonction `number_format()` permet la mise en forme de la valeur obtenue en utilisant l'espace comme séparateur de millier pour une meilleure lisibilité.

Exécution

1 126 861 978 secondes vécues

- Exercice 5

```
<?php
switch (date('N', mktime(0, 0, 0, 12, 25, 2017))) {
    case 1 : echo 'Lundi';
        break;
    case 2 : echo 'Mardi';
        break;
    case 3 : echo 'Mercredi';
        break;
    case 4 : echo 'Jeudi';
        break;
    case 5 : echo 'Vendredi';
        break;
    case 6 : echo 'Samedi';
        break;
    case 7 : echo 'Dimanche';
        break;
}
```

La fonction `date()` transforme un timestamp UNIX en une chaîne de caractères. Il faut indiquer le format souhaité pour la date via le premier paramètre. Le format N permet d'obtenir le numéro du jour de la semaine (1 pour Lundi jusqu'à 7 pour Dimanche). Le second paramètre est le timestamp UNIX à mettre en forme. Il est créé à l'aide de la fonction `mktime()` (cf. exercice 4).

Exécution

Lundi

- Exercice 6

```
<?php
echo '<ul>';
for ($i = 2018; $i <= 2037; $i++) {
    echo '<li> ' . $i . ' : ';
    switch (date('N', mktime(0, 0, 0, 5, 1, $i))) {
        case 1 :
        case 5 :
            echo 'Week-end prolongé !';
            break;
        case 6 :
        case 7 :
            echo 'Désolé !';
            break;
        default :
            echo 'En semaine';
            break;
    }
    echo '</li>';
}
echo '</ul>';
```

La première différence par rapport à l'exercice précédent est qu'une boucle est utilisée pour parcourir toutes les années comprises entre 2018 et 2037. Le second changement est pour le switch : les cas 1 et 5 (Lundi et Vendredi) sont groupés pour indiquer que ce jour férié crée un week-end prolongé et les cas 6 et 7 (Samedi et Dimanche) sont également regroupés pour indiquer que ce jour férié est pendant un week-end. Dans le reste des cas (Mardi, Mercredi et Jeudi), un message indiquera que ce jour férié est en milieu de semaine.

Exécution

- 2018 : En semaine
- 2019 : En semaine
- 2020 : Week-end prolongé !
- 2021 : Désolé !
- 2022 : Désolé !
- 2023 : Week-end prolongé !
- 2024 : En semaine
- 2025 : En semaine
- 2026 : Week-end prolongé !
- 2027 : Désolé !
- 2028 : Week-end prolongé !
- 2029 : En semaine
- 2030 : En semaine
- 2031 : En semaine
- 2032 : Désolé !
- 2033 : Désolé !
- 2034 : Week-end prolongé !
- 2035 : En semaine
- 2036 : En semaine
- 2037 : Week-end prolongé !

- Exercice 7

```
<?php
ini_set('date.timezone', 'Europe/Paris');
echo '<ul>';
for ($i = 2018; $i <= 2037; $i++) {
    $paques = easter_date($i);
    $ascension = $paques + 39 * 24 * 60 * 60;
    echo '<li>' . date('d/m/Y', $ascension) . '</li>';
}
echo '</ul>';
```

La première difficulté consiste à trouver la date de Pâques pour une année donnée. Le calcul est un peu complexe... Heureusement, il est déjà implémenté dans PHP avec la fonction `easter_date()`. Cette fonction prend en paramètre l'année pour la date de Pâques recherchée et retourne un timestamp UNIX. Pour trouver la date de l'Ascension, il faut ajouter 39 jours. Un timestamp UNIX étant un nombre de secondes, il faut ajouter 39 jours convertis en secondes.

Exécution

- 10/05/2018
- 30/05/2019
- 21/05/2020
- 13/05/2021
- 26/05/2022
- 18/05/2023
- 09/05/2024
- 29/05/2025
- 14/05/2026
- 06/05/2027
- 25/05/2028
- 10/05/2029
- 30/05/2030
- 22/05/2031
- 06/05/2032
- 26/05/2033
- 18/05/2034
- 03/05/2035
- 22/05/2036
- 14/05/2037

- Exercice 8

```
<?php
echo '<table><tr><th colspan="7">';
switch (date('n')) {
    case 1: echo 'Janvier';
        break;
    case 2: echo 'Février';
        break;
    case 3: echo 'Mars';
        break;
    case 4: echo 'Avril';
        break;
    case 5: echo 'Mai';
        break;
    case 6: echo 'Juin';
        break;
```

```

case 7: echo 'Juillet';
        break;
case 8: echo 'Août';
        break;
case 9: echo 'Septembre';
        break;
case 10: echo 'Octobre';
        break;
case 11: echo 'Novembre';
        break;
case 12: echo 'Décembre';
        break;
}
echo ' ' . date('Y') . '</th></tr><tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th><th>S</th><th>D</th></tr><tr>';
$donnees = mktime(0, 0, 0, date('n'), 1);
for ($i = 1; $i < date('N', $donnees); $i++) {
    echo '<td></td>';
}
for ($i = 1; $i <= date('t'); $i++) {
    echo '<td>' . $i . '</td>';
    if (date('N', $donnees) == 7) {
        echo '</tr><tr>';
    }
    $donnees += 60 * 60 * 24;
}
for ($i = date('N', $donnees); $i <= 7; $i++) {
    echo '<td></td>';
}
echo '</tr></table>';

```

La fonction `date()` utilisée avec le format `n` permet d'obtenir le numéro du mois (1 pour Janvier jusqu'à 12 pour Décembre).

Si le second paramètre n'est pas renseigné, la date actuelle est utilisée par défaut. Le second paramètre est le timestamp UNIX à mettre en forme. Il est créé à l'aide de la fonction `mktime()` (cf. exercice 4). Selon le numéro de ce mois, le nom du mois correspondant est affiché.

La fonction `date()` avec le format `Y` permet d'obtenir l'année sur 4 chiffres.

La variable `$donnees` est initialisée au premier jour du mois courant. Selon le jour de la semaine de cette date, il faut laisser un certain nombre de cases vides dans le tableau, ceci est réalisé par la première boucle.

Tous les jours du mois sont itérés dans la seconde boucle. Pour connaître le nombre de jours dans un mois, il faut encore une fois faire appel à la fonction `date()` mais cette fois avec le paramètre `t`. Pour chaque jour, son numéro est affiché dans une case du tableau. Si le jour est un dimanche, la ligne est terminée, il faut donc commencer une nouvelle ligne.