

Boucles PHP

I) Boucles while et for

On peut effectuer une suite d'instructions en boucle afin tant qu'une condition est réalisée. Attention: Il faut s'assurer que la condition ne sera plus remplie à un moment. Sinon, la boucle ne s'arrêtera jamais!

Syntaxe while
<pre>while (condition) { instructions; }</pre>

Remarque: On peut incrémenter une variable entière `$var` à l'aide de l'instruction `$var++`.

Exercice1:

Créer un script qui demande un nombre entier `n` à l'utilisateur et affiche la table de multiplication de `n`.

Exercice 2: Que représente la valeur affichée à la fin du script ci-contre?

exercice_2.php
<pre>\$i = 0; \$n = 0; while (\$i < 100){ \$i = \$i + 2; \$n = \$n + \$i; } echo \$n;</pre>

On peut également faire en boucle une suite d'instructions à l'aide d'une boucle **for**.

- **init** sera effectué au départ de la boucle
- **cond** est une condition qui sera évaluée au début de chaque boucle. On effectue la boucle for seulement si cette condition est vraie.
- **iter** est une instruction qui sera effectuée à la fin de chaque boucle.

Syntaxe for
<pre>for (init; cond; iter) { instructions; }</pre>

Exercice 3:

Que va afficher le script ci-contre?

exercice_3.php
<pre>for (\$i=0; \$i < 10; \$i++){ echo \$i + 3; }</pre>

Exercice TP 4:

Créer un script qui produit l'affichage ci-contre.

Remarque: On peut produire une nouvelle ligne dans la console en affichant la chaîne de la constante `PHP_EOL`.

Affichage
1
22
333
4444
55555

Exercice TP 5: Créer un script affichant les 100 premiers nombres impairs à l'aide d'une boucle for.

Remarque: L'instruction **break** permet de sortir complètement d'une boucle **while**, **for** ou **foreach**.

Exercice TP 6:

Créer un script qui tire 1000 nombres entiers entre 1 et 100 au hasard à l'aide de la fonction `rand()`. Si un, par hasard, on obtient le nombre 42, le script doit s'arrêter et afficher alors le nombre d'entiers ayant été tirés au sort.

III) Boucles foreach

On peut parcourir les éléments d'un tableau `$tableau` à l'aide de l'instruction `foreach`.

Dans le premier exemple, `$val` prendra la valeur de chaque élément du tableau.

Avec clef, `$clef` prendra les valeurs des clefs et `$val` prendra la valeur des éléments correspondants.

Syntaxe <code>foreach</code>
<pre>foreach (\$tableau as \$val){ instruction; }</pre>
Syntaxe <code>foreach</code> avec clef
<pre>foreach (\$tableau as \$clef => \$val){ instruction; }</pre>

exemple_1.php	exemple_2.php
<pre>\$pairs = array(0, 2, 4, 6, 8); foreach(\$pairs as \$val){ echo 3 * \$val . PHP_EOL; if (\$val == 4){ break;} }</pre>	<pre>\$champions = array(2018 => "France", 2014 => "Allemagne", 2010 => "Espagne"); foreach(\$champions as \$clef => \$val){ echo "\$clef : \$val est championne du monde!\n" }</pre>
Console	Console
<pre>PS C:\> php exemple_1.php 0 6 12</pre>	<pre>PS C:\> php exemple_2.php 2018: France est championne du monde! 2014: Allemagne est championne du monde! 2010: Espagne est championne du monde!</pre>

Exercice TP 7:

1. Remplir un tableau de 100 nombres entier entre 0 et 50 au hasard à l'aide de la fonction `rand()`.
2. A l'aide d'une boucle, déterminer le maximum, la moyenne et le minimum de ce tableau.
3. Préciser dans l'affichage un indice du tableau contenant le minimum et le maximum.
4. Préciser dans l'affichage tous les indices du tableau contenant le minimum et le maximum.

Exercice TP 8:

1. Ouvrir le fichier `ex8.php`
2. Pour chaque année, afficher: **2022: France a gagné contre Croatie**
3. Afficher le(s) pays ayant gagné le plus de coupes du monde.
4. Afficher le(s) pays ayant été le plus en finale.

Exercice TP 9:

1. A l'aide de la fonction `scandir()`, afficher le contenu du dossier fourni.
2. Renommer chaque fichier afin d'enlever les caractères spéciaux et de remplacer les espaces par des `_`.

On pourra utiliser les fonctions `rename()` et `str_replace()`.