

Boucles en PHP

Mot-clé de rupture PHP

En PHP, `break` peut être utilisé pour terminer l'exécution d'une boucle, `for` ou `foreach` `while` `do...while`.
L'un des inconvénients d'une utilisation intensive des instructions `break` est que le code peut devenir moins lisible.

```
// We can use the break statement to end
the loop once the count reaches 4
$count = 1;
while ($count < 10)
{
    echo "The count is: " . $count . "\n";
    if ($count === 4) {
        break;
    }
    $count += 1;
}
```

PHP fait des boucles while

En PHP, `do...while` les boucles sont très similaires à `while` loops. La principale différence est que les `do...while` boucles exécutent toujours leur bloc de code au moins une fois et continuent leur exécution tant que leur instruction conditionnelle est vraie. Même si la condition est fausse, le bloc de code s'exécutera une fois.

La syntaxe d'une `do...while` boucle est :

```
do {
    #code block
} while (/*conditional*/);
```

```
// This loop counts from 0 to 100
$count = 0;
do {
    echo "The count is: " . $count . "\n";
    $count += 1;
} while ($count <= 100);
```

PHP pour la boucle

En PHP, une `for` boucle est couramment utilisée pour exécuter un bloc de code un certain nombre de fois. La syntaxe utilise trois expressions :

```
for (#expression 1; #expression  
2; #expression 3)  
{  
    # code block  
}
```

Le premier est évalué une seule fois avant la première itération

La seconde est évaluée avant chaque itération. S'il est TRUE, le bloc de code est exécuté. Sinon, la boucle se termine.

Le troisième est évalué après chaque itération.

```
// This for loop counts from 1 to 50  
for ($count = 1; $count < 51; $count++)  
{  
    echo "The count is: " . $count . "\n";  
}
```

PHP boucles while

En PHP, `while` les boucles répètent l'exécution de leur bloc de code tant que leur instruction conditionnelle est vraie.

La syntaxe d'une `while` boucle est :

```
while (/*conditional*/) {  
    #code block  
}
```

```
// This while loop counts from 0 to 100  
$count = 0;  
while ($count <= 100)  
{  
    echo "The count is: " . $count . "\n";  
    $count += 1;  
}
```

Boucle foreach PHP

En PHP, la `foreach` boucle est utilisée pour itérer sur un tableau. Le bloc de code est exécuté pour chaque élément du tableau et la valeur de cet élément est disponible pour être utilisée dans le bloc de code.

La syntaxe est :

```
foreach ($array as $value) {  
    #code block  
}
```

À chaque itération, une `$value` de `$array` est disponible pour être utilisée dans le bloc de code.

Pour accéder aux clés ainsi qu'aux valeurs d'un tableau, la syntaxe peut être modifiée en :

```
foreach ($array as $key =>  
    $value) {  
    #code block  
}
```

PHP continuer mot-clé

En PHP, `continue` peut être utilisé pour terminer l'exécution d'une itération de boucle pendant une boucle `for`, `foreach` ou `while`. L'exécution du code se poursuit avec la prochaine itération de la boucle. `while do...while`

Le `continue` mot-clé est similaire à `break` sauf qu'il ne termine que l'itération en cours plus tôt, pas la boucle entière.

```
// This foreach loop counts from 1 to 5  
$nums = [1, 2, 3, 4, 5];  
foreach ($nums as $num) {  
    echo "The num is: " . $num . "\n";  
}
```

```
// This code counts from 1 to 10 but  
skips over 5  
$count = 1;  
while ($count < 11)  
{  
    if ($count === 5) {  
        $count += 1;  
        continue;  
    }  
    echo "The count is: " . $count . "\n";  
    $count += 1;  
}
```

PHP raccourci de boucle while

En PHP, le raccourci pour une `while` boucle est :

```
while(/*condition*/):  
# code block  
endwhile;
```

Lors de l'intégration dans HTML, cela est préférable à la syntaxe des crochets, car il est beaucoup plus clair quel bloc de code se termine par le `endwhile` .

```
<ul>  
<?php  
$i = 0;  
while ($i < 2):  
?>  
<li>Duck</li>  
<?php  
$i++;  
endwhile;  
?>  
<li>Goose</li>  
</ul>
```

PHP pour raccourcir la boucle

En PHP, le raccourci pour une `for` boucle est :

```
for (/*condition*/):  
# code block  
endfor;
```

Lors de l'intégration dans HTML, cela est préférable à la syntaxe des crochets, car il est beaucoup plus clair quel bloc de code se termine par le `endfor` .

```
<ul>  
<?php  
for ($i = 0; $i < 2; $i++):  
?>  
<li>Duck</li>  
<?php  
endfor;  
?>  
<li>Goose</li>  
</ul>
```

Raccourci de la boucle PHP foreach

En PHP, le raccourci pour une `foreach` boucle est :

```
foreach ($array as $value):  
# code block  
endforeach;
```

Lors de l'intégration dans HTML, cela est préférable à la syntaxe des crochets, car il est beaucoup plus clair quel bloc de code se termine par le `endforeach` .

```
<ul>  
<?php  
$array = [0, 1];  
foreach ($array as $i):  
?>  
<li>Duck</li>  
<?php  
endforeach;  
?>  
<li>Goose</li>  
</ul>
```