Web Programming

#7 PHP - Loops

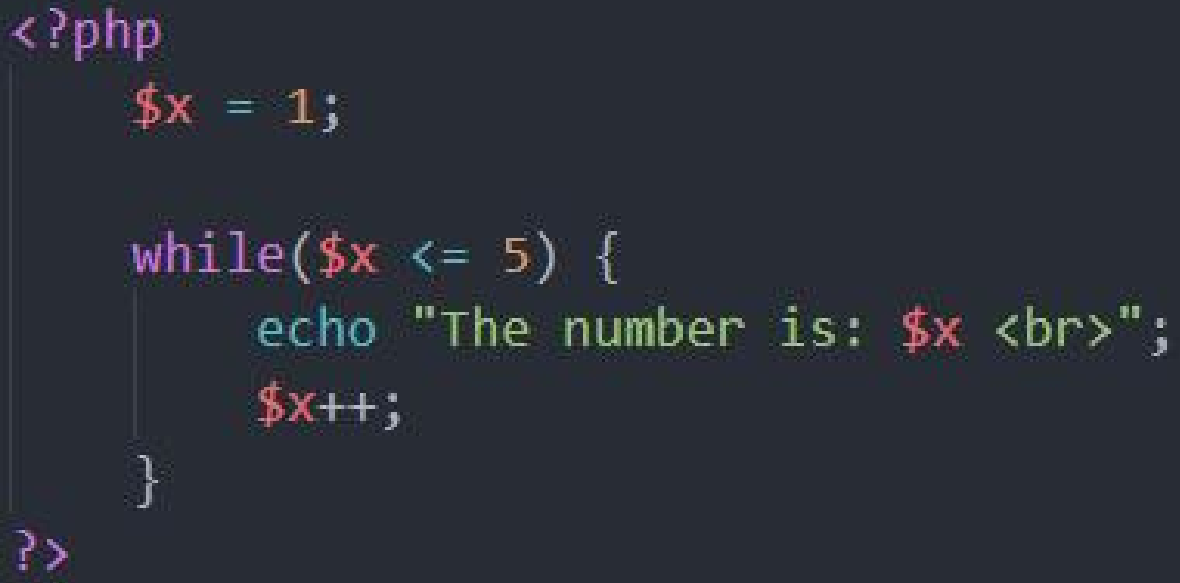
Herman Kabetta, M.T.

hermanka.github.io

Why Use Loops

WHILE ...

```
while (condition is true) {
    // code to be executed;
}
```



```php
<?php
    $x = 1;

    while($x <= 5) {
        echo "The number is: $x <br>";
        $x++;
    }
?>
```

WHILE ... - Exercise

We want to examine every number from 1 to 100 and print the number followed by " is divisible by 33\n" if the number is divisible by 33.

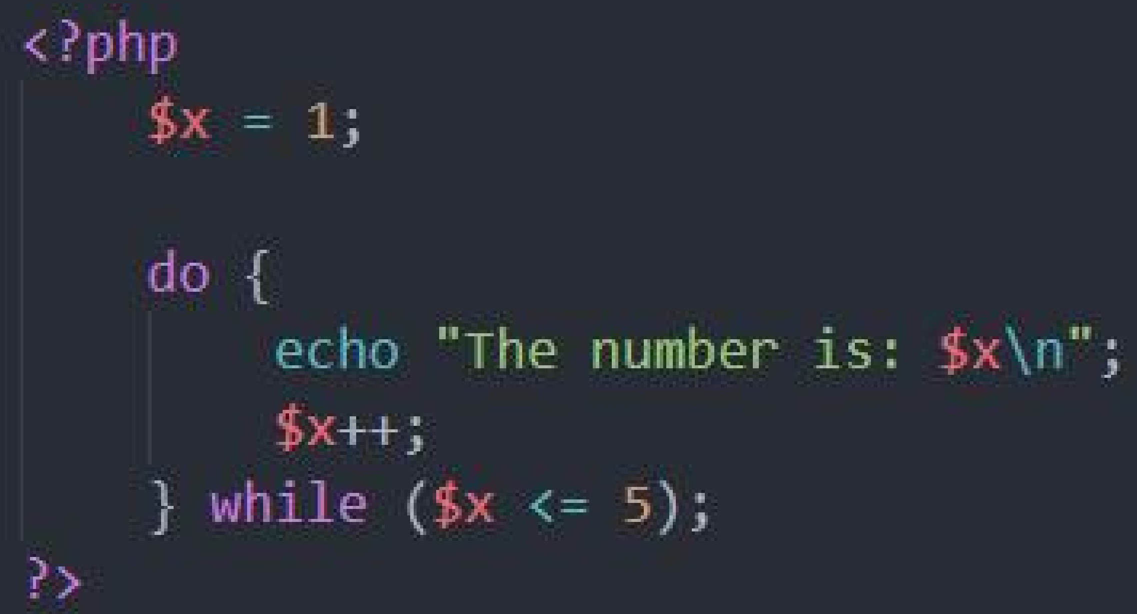
Begin by setting up a loop that counts from 1 to 100.

e.g. Output :

```
33 is divisible by 33
66 is divisible by 33
99 is divisible by 33
```

DO … WHILE …

```
do {
    // code to be executed;
} while (condition is true);
```

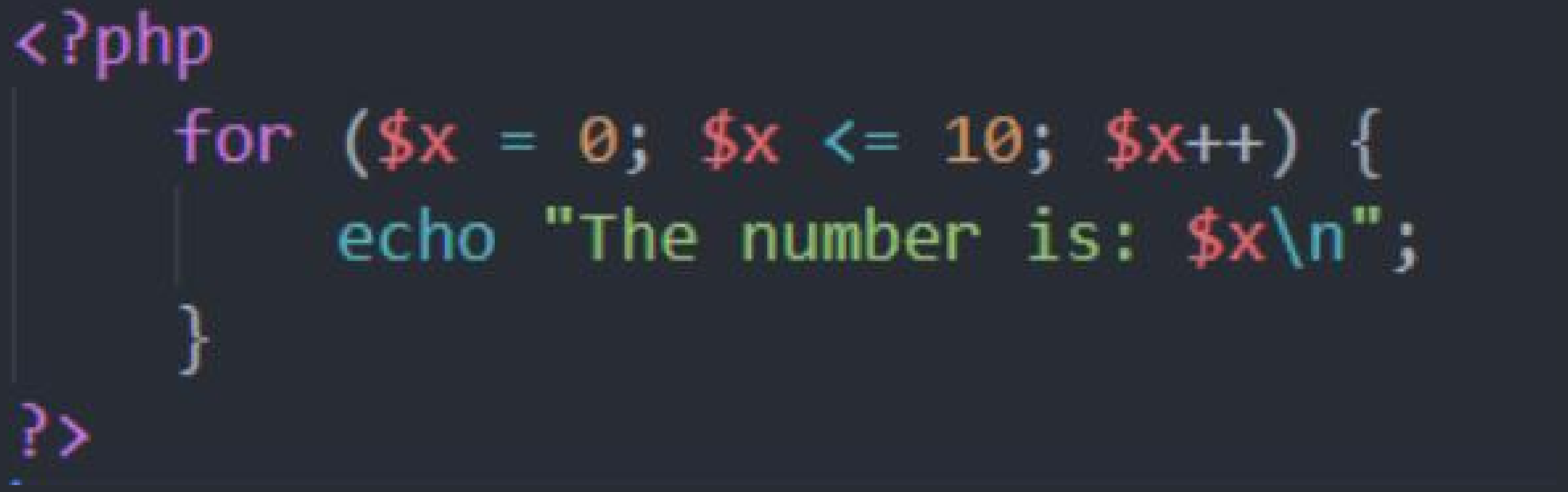


```php
<?php
    $x = 1;

    do {
        echo "The number is: $x\n";
        $x++;
    } while ($x <= 5);
?>
```

FOR

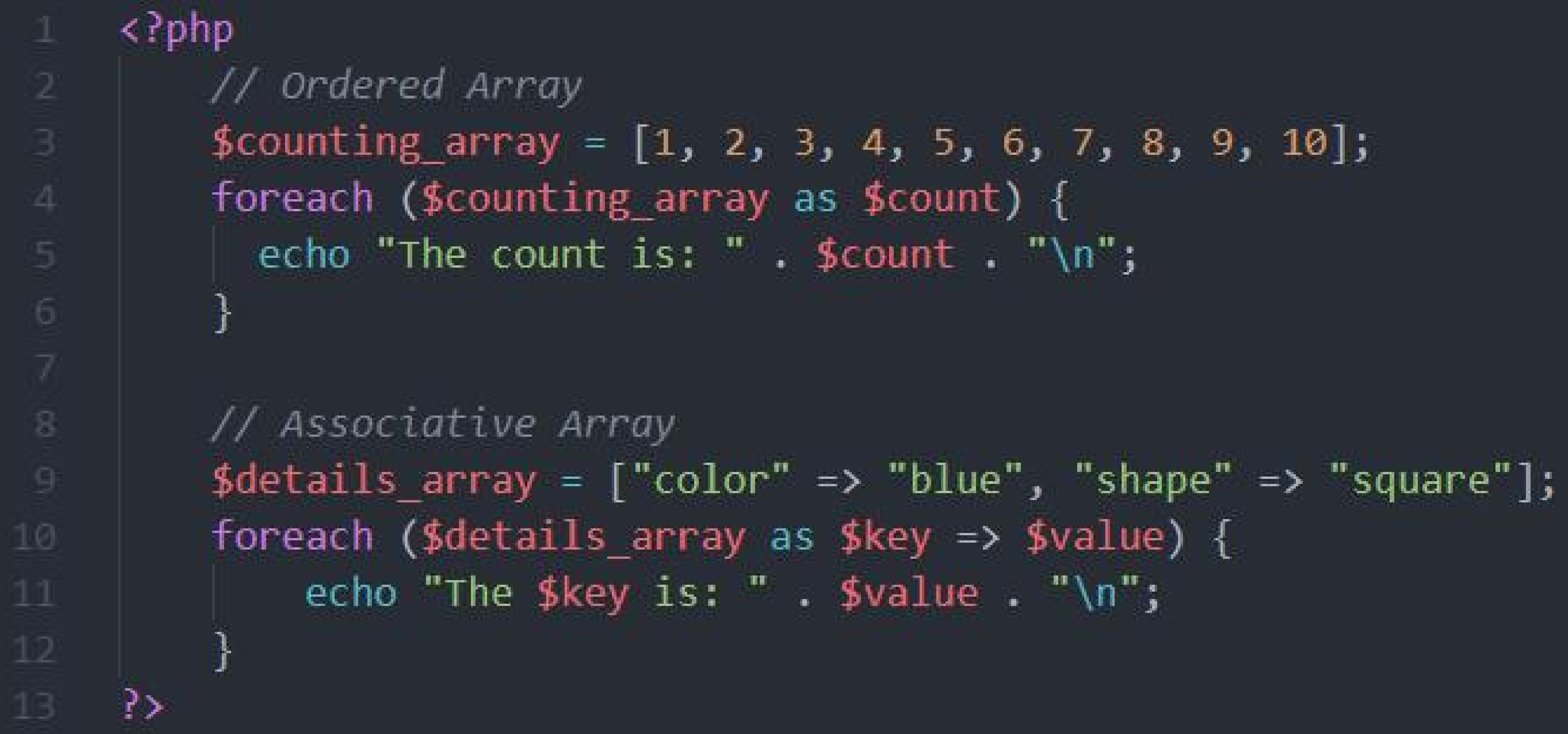
```
for (init; conditional; increment) {
    // code to be executed;
}
```



```php
<?php
    for ($x = 0; $x <= 10; $x++) {
        echo "The number is: $x\n";
    }
?>
```

FOREACH

The foreach loop is used for iterating over an array.

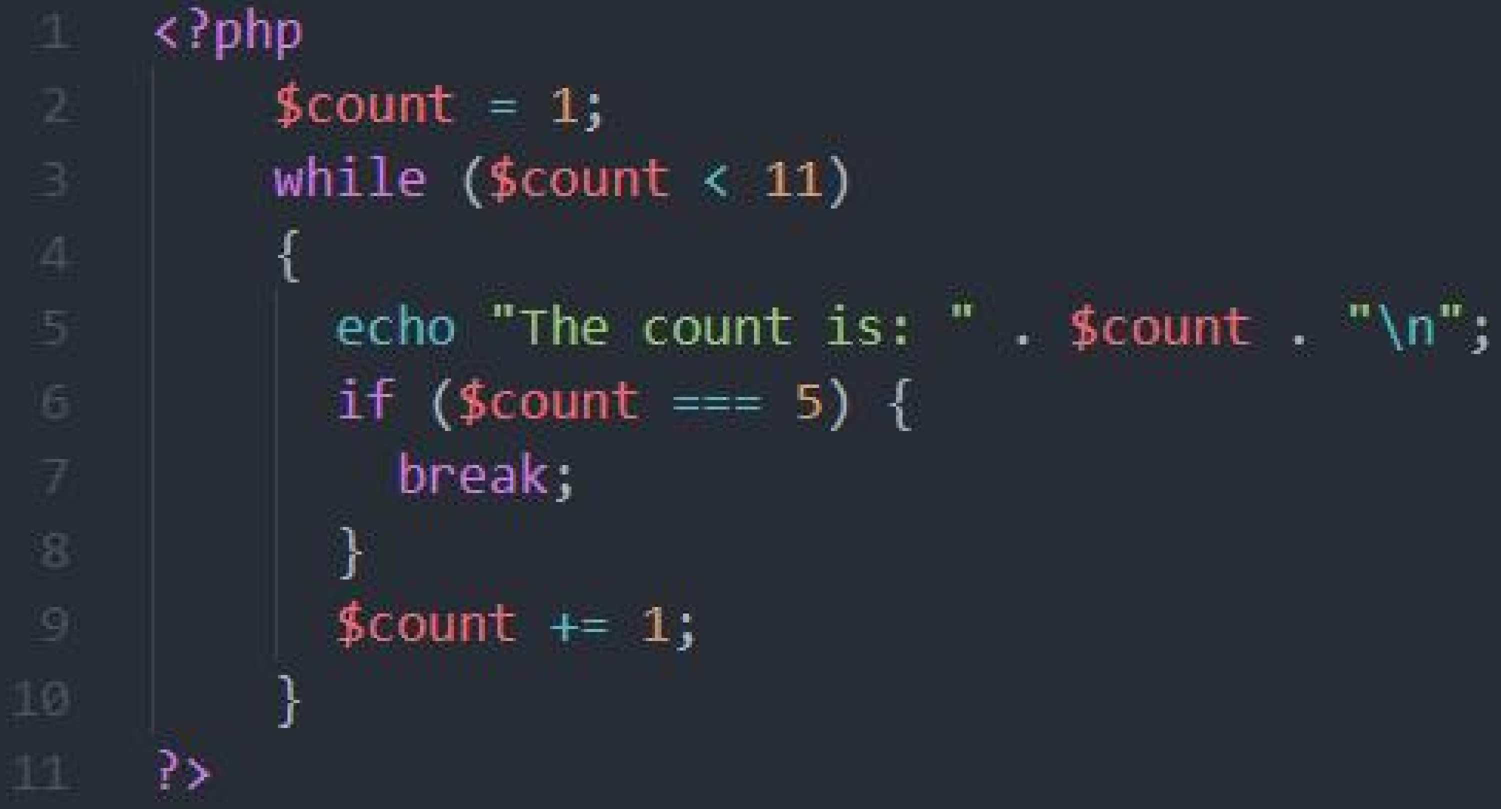


```php
<?php
    // Ordered Array
    $counting_array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
    foreach ($counting_array as $count) {
      echo "The count is: " . $count . "\n";
    }

    // Associative Array
    $details_array = ["color" => "blue", "shape" => "square"];
    foreach ($details_array as $key => $value) {
        echo "The $key is: " . $value . "\n";
    }
?>
```

BREAK & CONTINUE

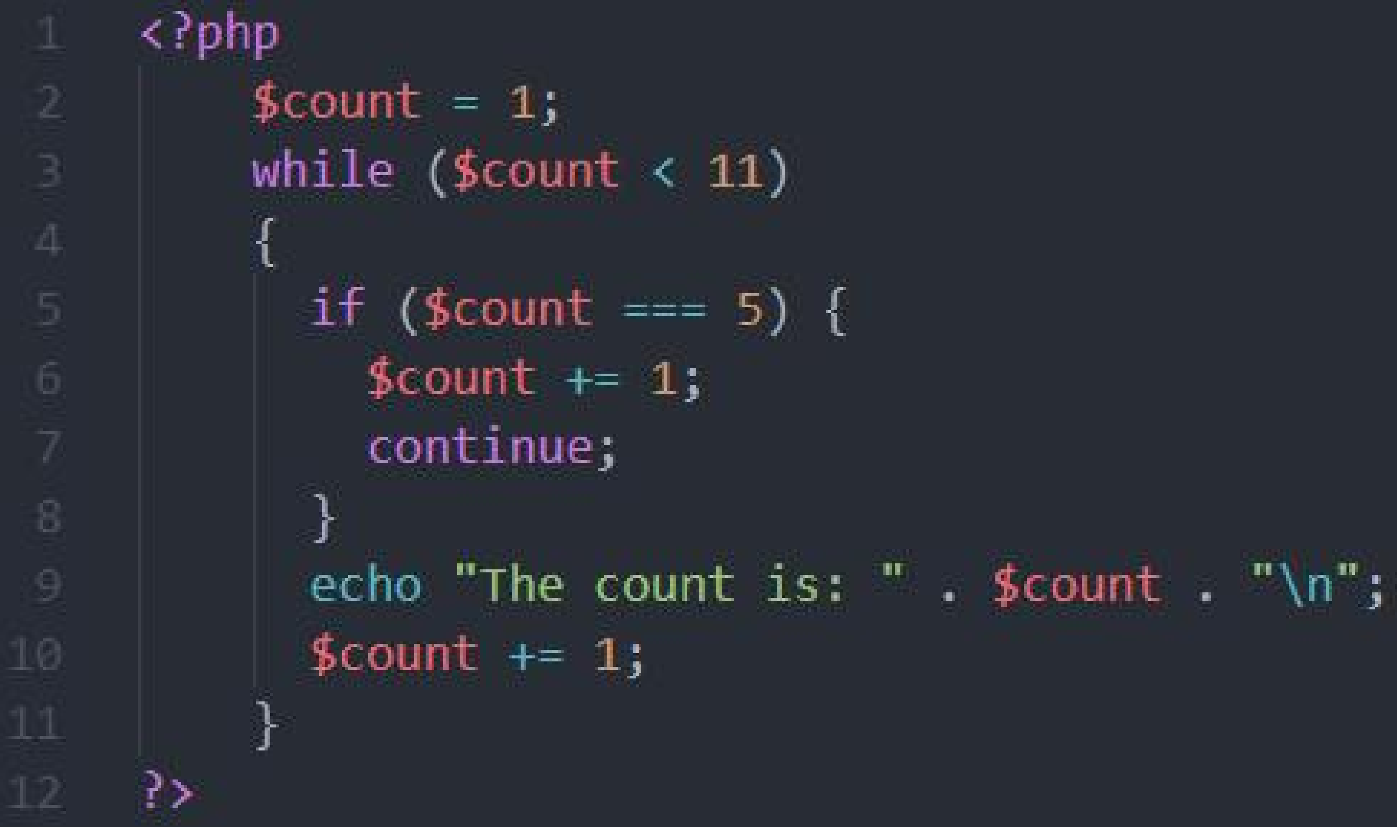
The **break** keyword can be used to terminate any of the loop types early



```php
<?php
    $count = 1;
    while ($count < 11)
    {
      echo "The count is: " . $count . "\n";
      if ($count === 5) {
        break;
      }
      $count += 1;
    }
?>
```

BREAK & CONTINUE

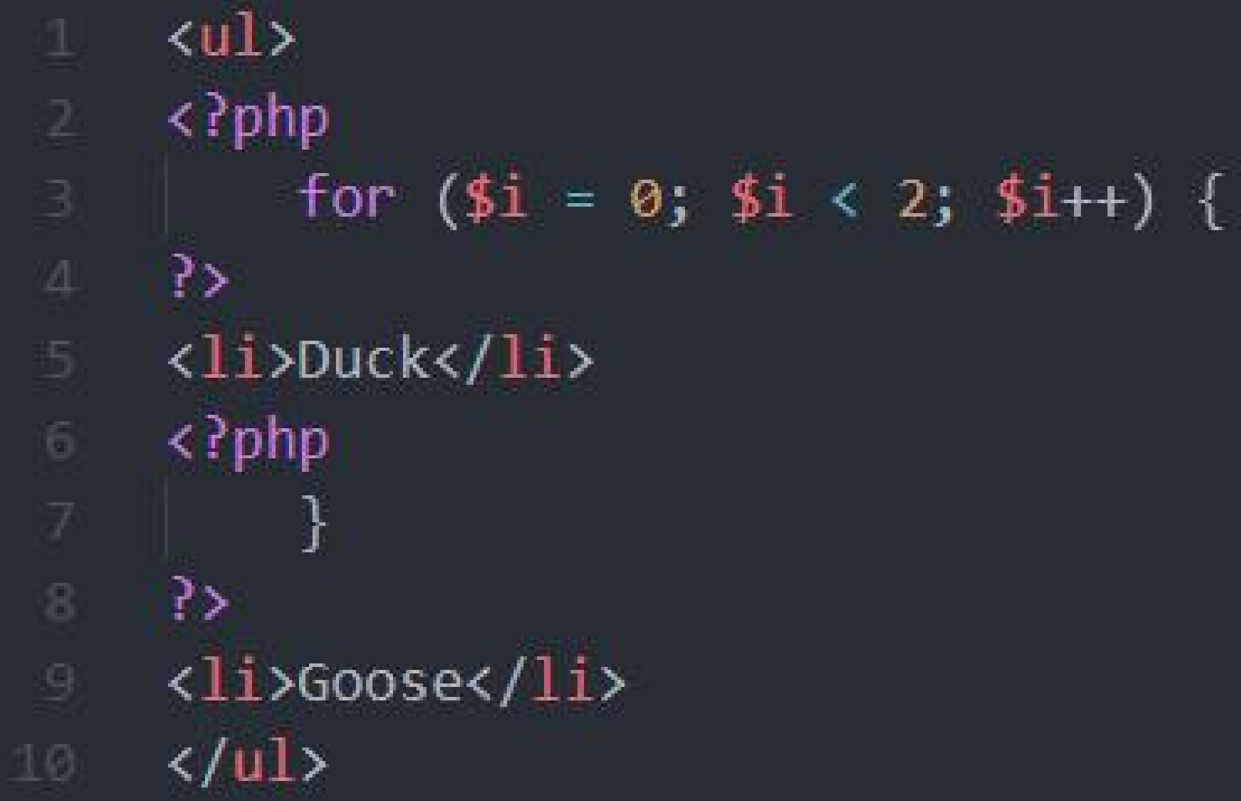
The **continue** keyword is similar to break except it only ends the current iteration early, not the entire loop.



```php
<?php
    $count = 1;
    while ($count < 11)
    {
      if ($count === 5) {
        $count += 1;
        continue;
      }
      echo "The count is: " . $count . "\n";
      $count += 1;
    }
?>
```

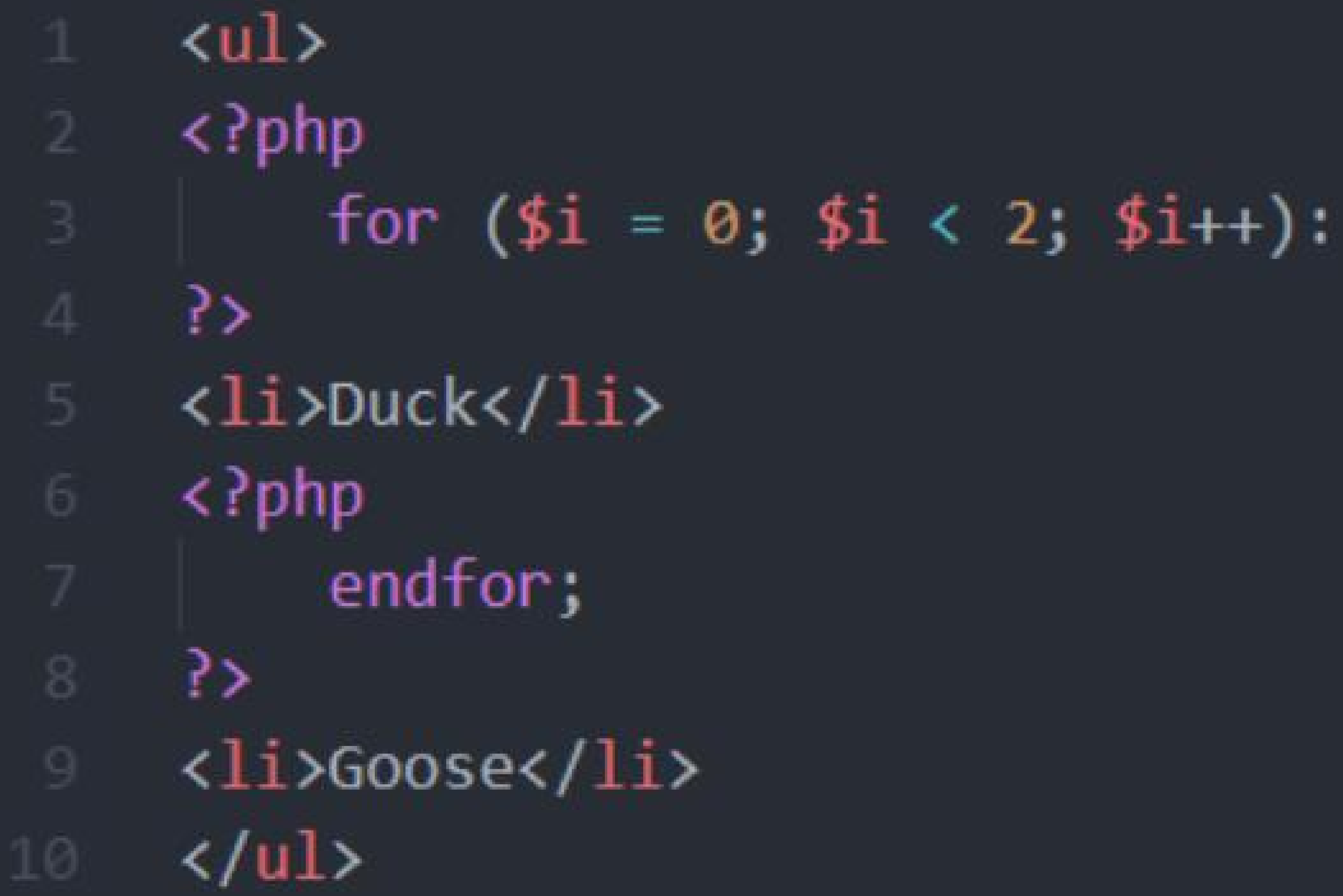
LOOPS IN HTML

Embedding PHP code in HTML



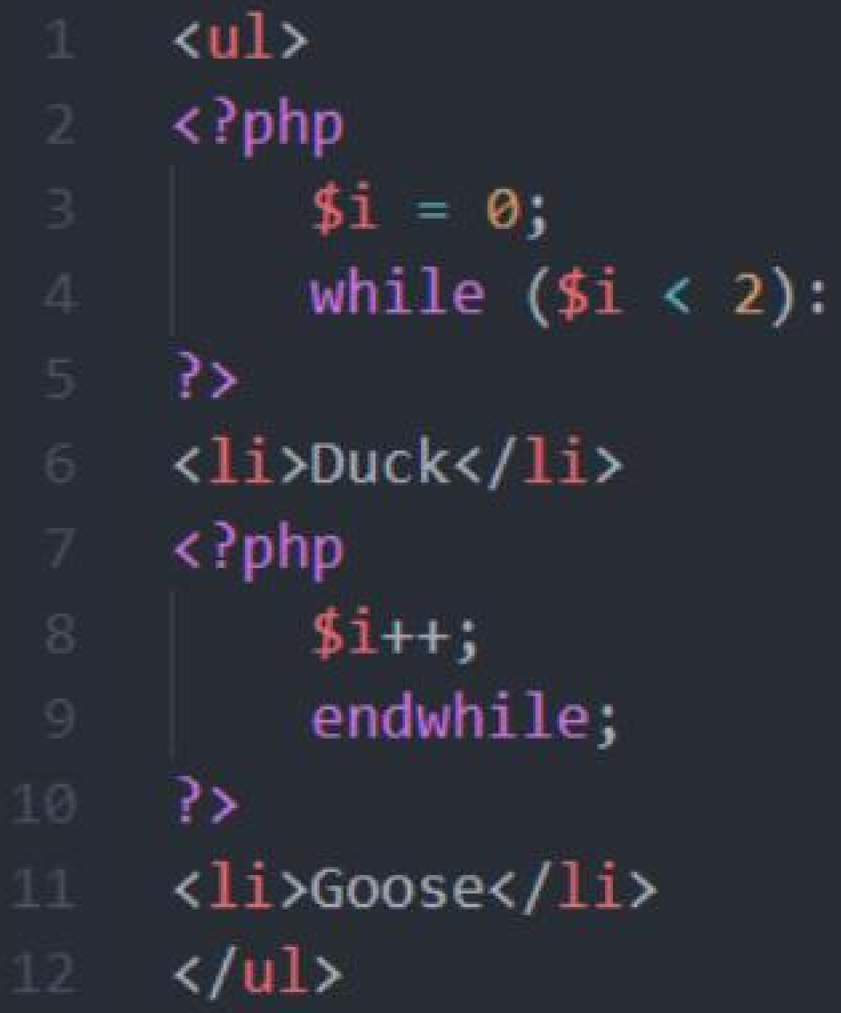
```
<ul>
<?php
    for ($i = 0; $i < 2; $i++) {
?>
<li>Duck</li>
<?php
    }
?>
<li>Goose</li>
</ul>
```

Shorthand LOOPS IN HTML



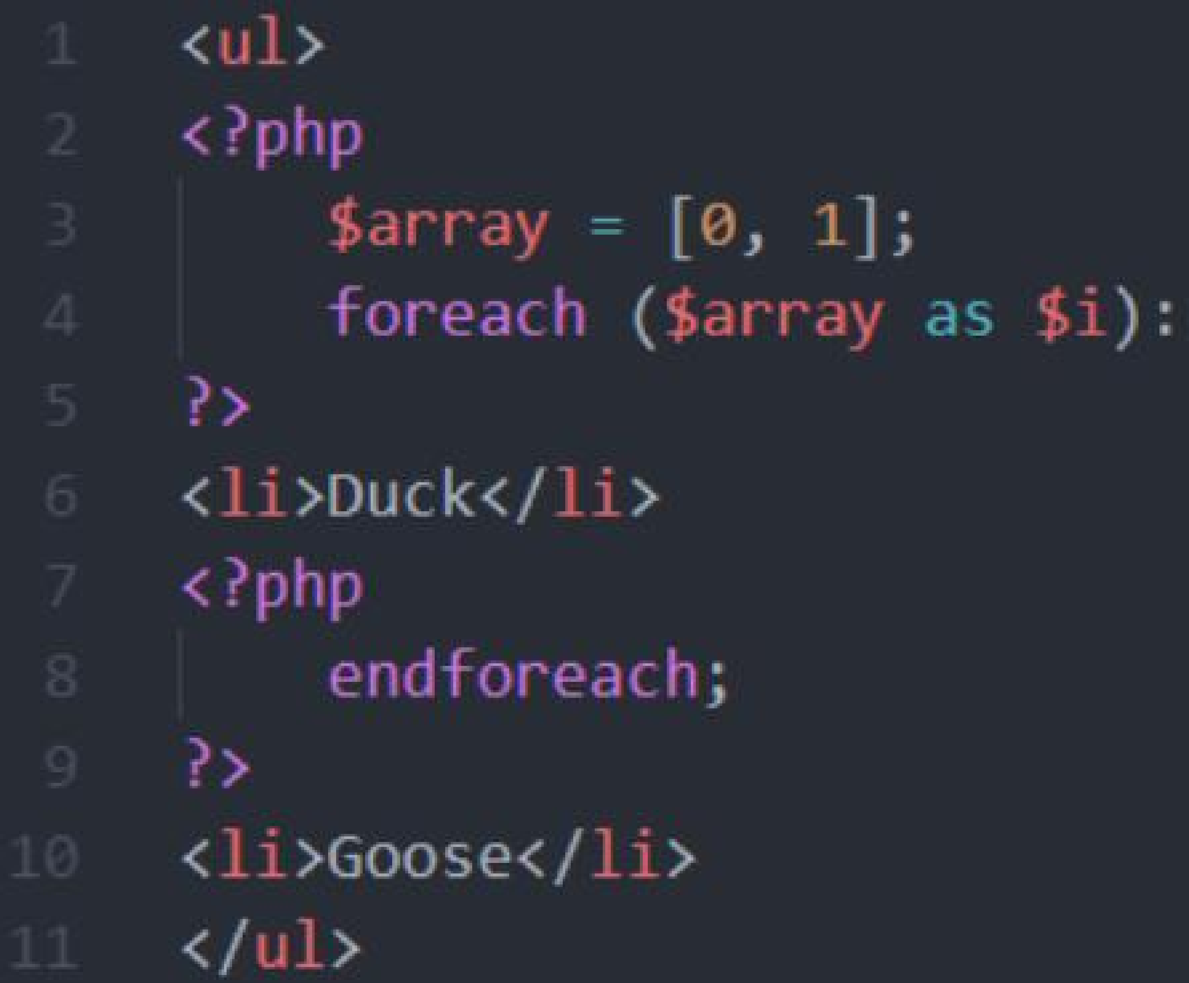
```
<ul>
<?php
    for ($i = 0; $i < 2; $i++):
?>
<li>Duck</li>
<?php
    endfor;
?>
<li>Goose</li>
</ul>
```

Shorthand LOOPS IN HTML



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

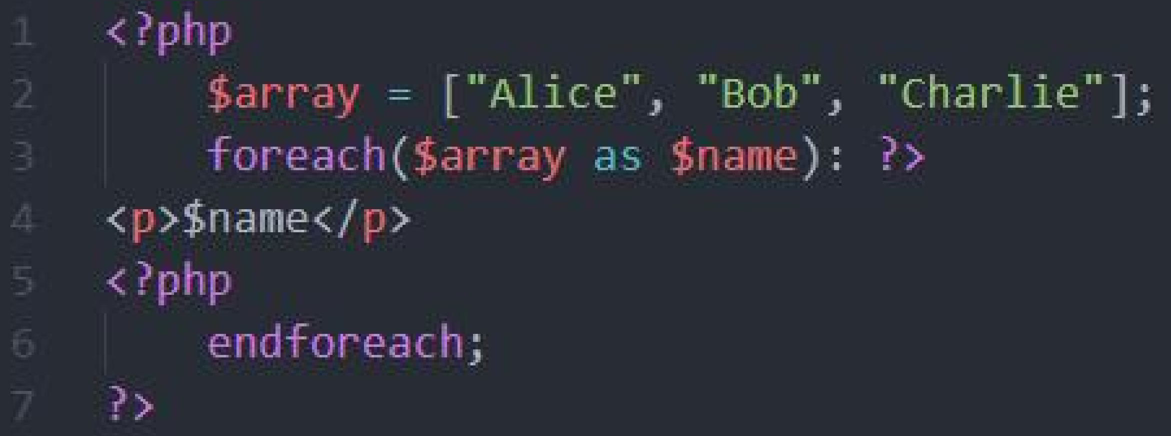
Shorthand LOOPS IN HTML



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

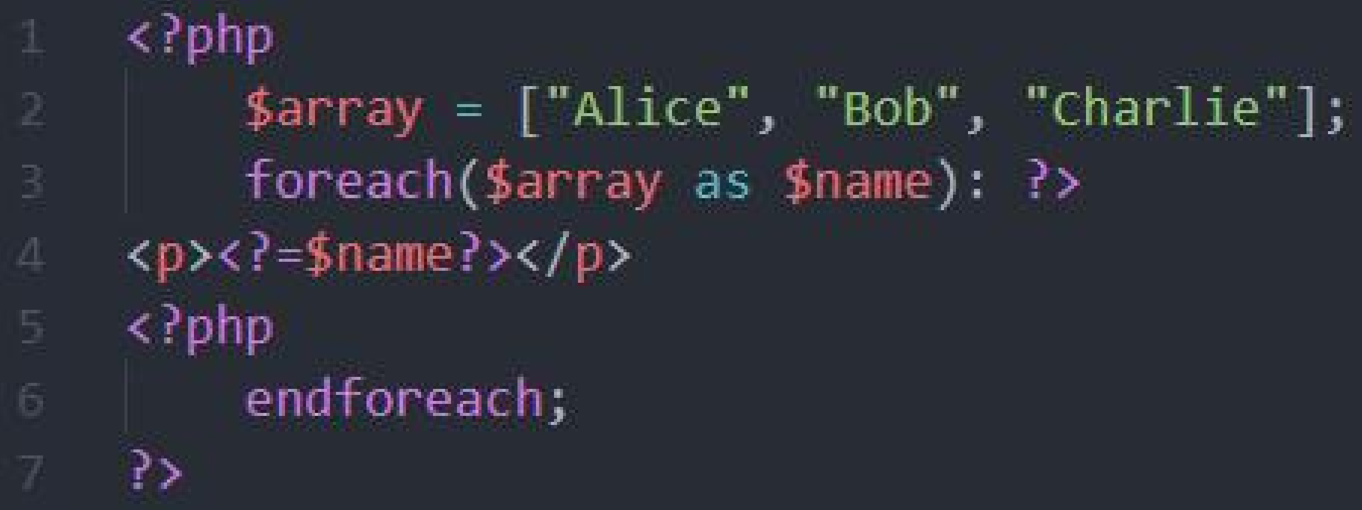
Code Block Considerations

✗



```
<?php
    $array = ["Alice", "Bob", "Charlie"];
    foreach($array as $name): ?>
<p>$name</p>
<?php
    endforeach;
?>
```

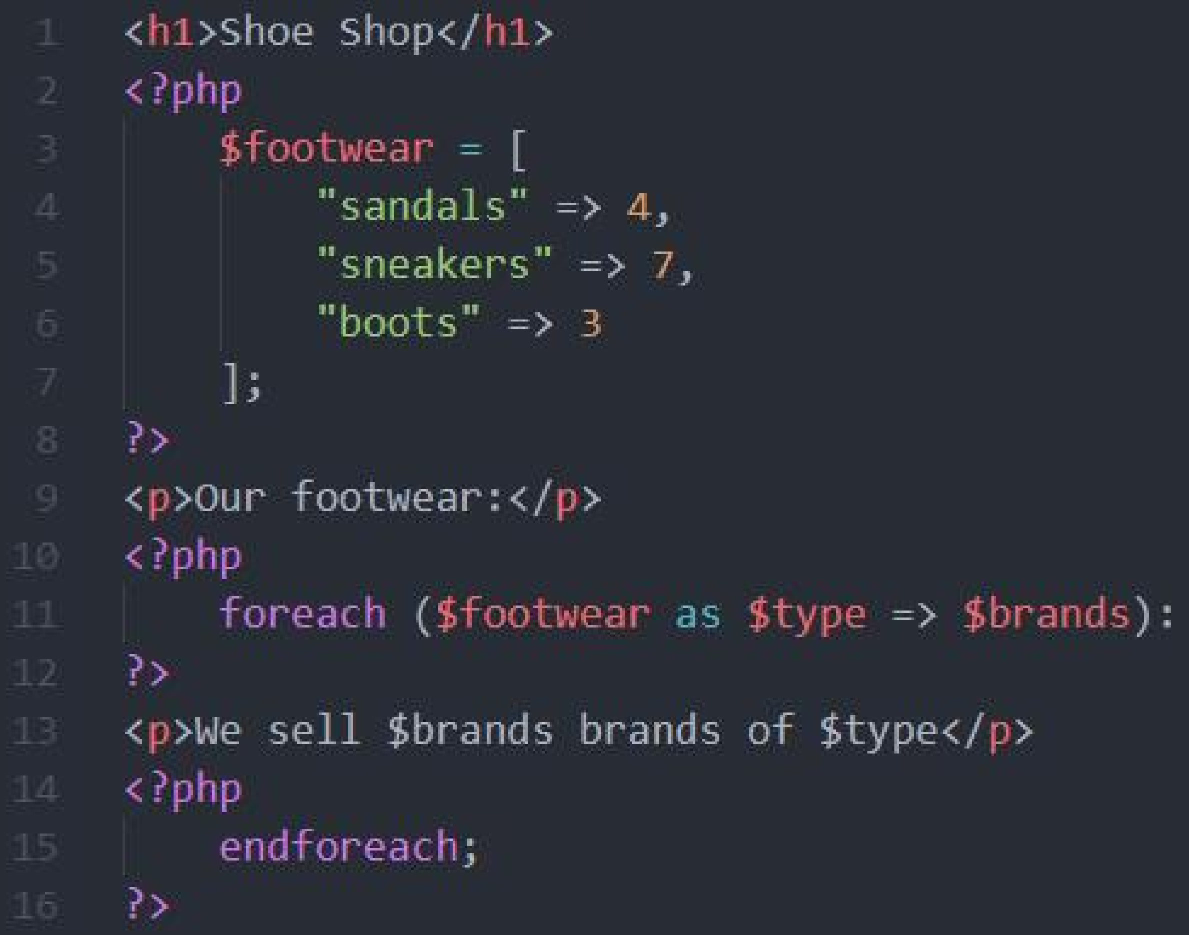
✓



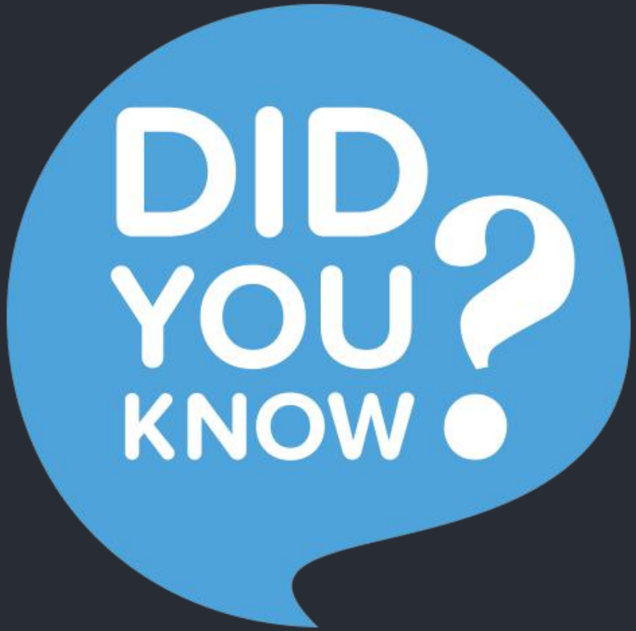
```
<?php
    $array = ["Alice", "Bob", "Charlie"];
    foreach($array as $name): ?>
<p><?=$name?></p>
<?php
    endforeach;
?>
```

Exercise

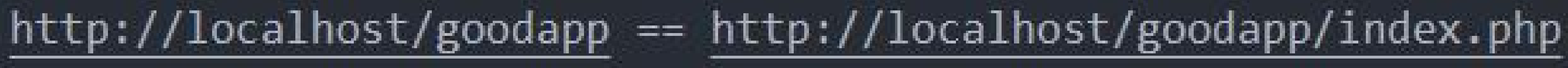
Fixes code below using echo shorthand



```
<h1>Shoe Shop</h1>
<?php
    $footwear = [
        "sandals" => 4,
        "sneakers" => 7,
        "boots" => 3
    ];
?>
<p>Our footwear:</p>
<?php
    foreach ($footwear as $type => $brands):
?>
<p>We sell $brands brands of $type</p>
<?php
    endforeach;
?>
```



The **index.php** is a PHP file that is the entry point of any website or application.



```
http://localhost/goodapp == http://localhost/goodapp/index.php
```