HERAT UNIVERSITY
COMPUTER SCEINE FACULTY

Web Engineering II (Semester 6)

(PHP Loops & Functions)

LECTURER: HAMED AMIRY

PHP Loops

> Often when you write code, you want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.

> In PHP, we have the following looping statements:

PHP Loops

> **while** - loops through a block of code while a specified condition is true

> **do...while** - loops through a block of code once, and then repeats the loop as long as a specified condition is true

> **for** - loops through a block of code a specified number of times

> **foreach** - loops through a block of code for each element in an array

PHP Loops - While

The while loop executes a block of code while a condition is true. The example below defines a loop that starts with

i=1. The loop will

continue to run as

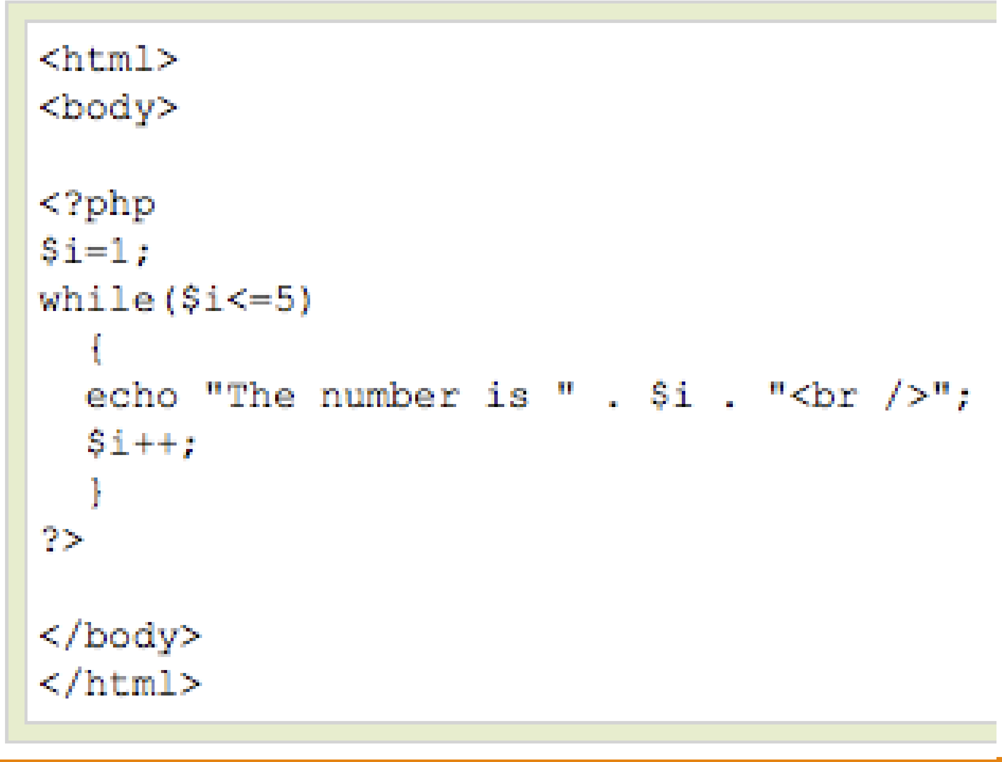
long as i is less

than, or equal to 5.

i will increase by 1

each time the loop

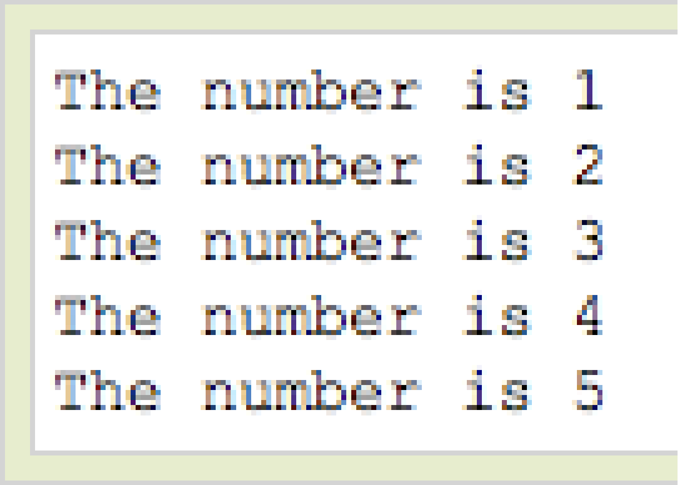
runs:



```
<html>
<body>

<?php
$i=1;
while($i<=5)
  {
  echo "The number is " . $i . "<br />";
  $i++;
  }
?>

</body>
</html>
```

PHP Loops - While

Output:



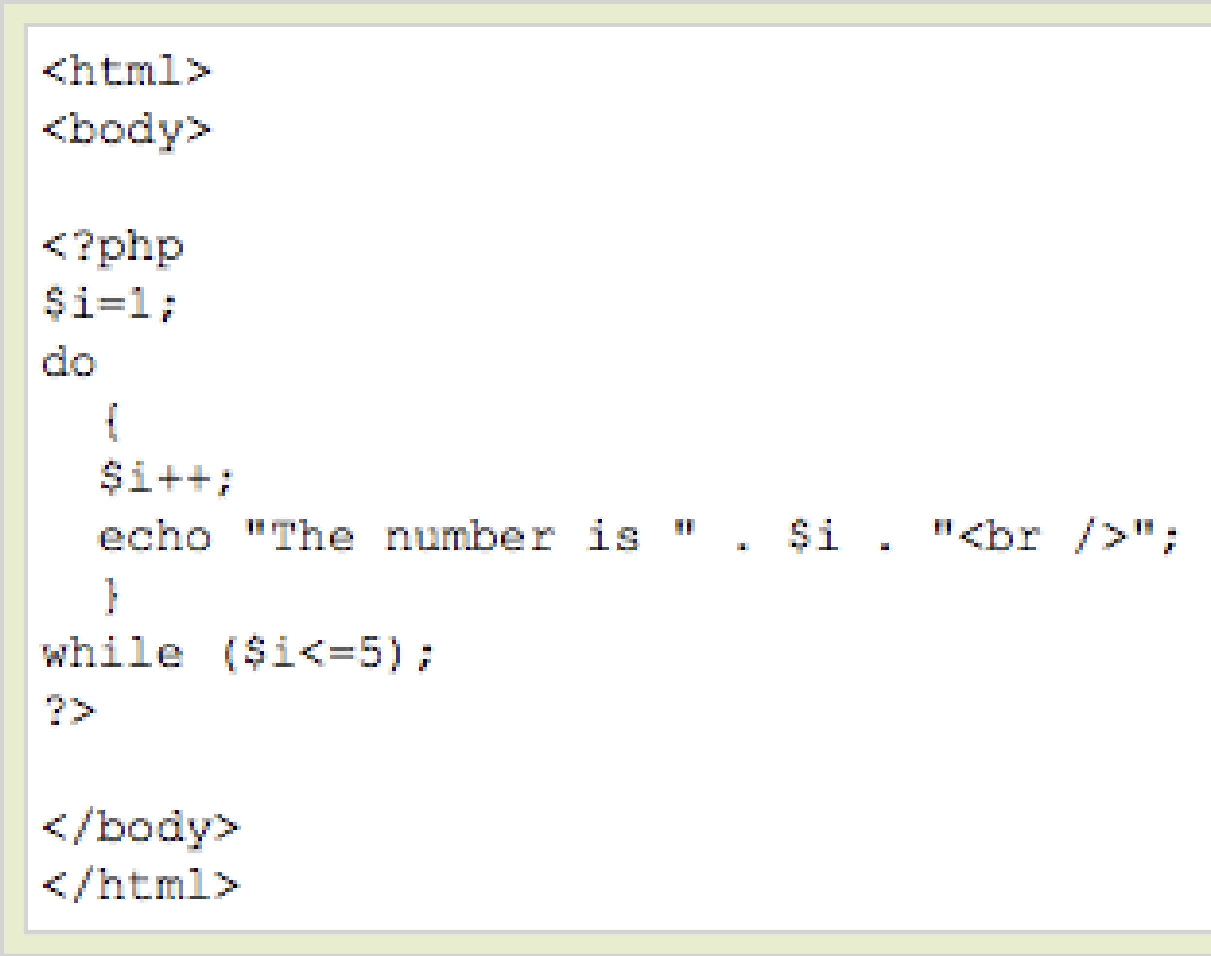
```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

PHP Loops – Do … While

The do...while statement will always execute the block of code once, it will then check the condition, and repeat the loop while the condition is true.

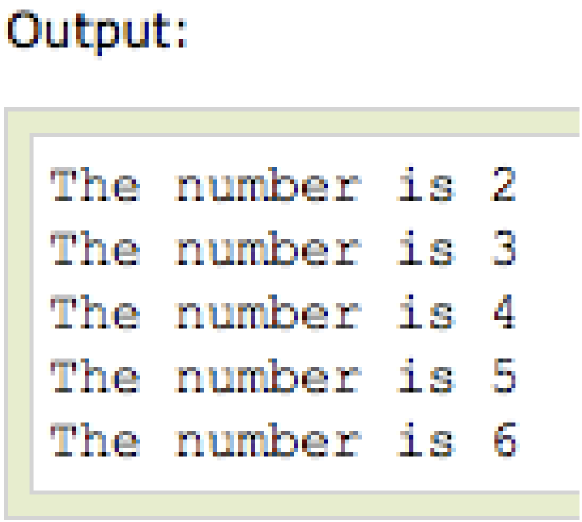
The next example defines a loop that starts with i=1. It will then increment i with 1, and write some output. Then the condition is checked, and the loop will continue to run as long as i is less than, or equal to 5:

PHP Loops – Do … While



```
<html>
<body>

<?php
$i=1;
do
  {
  $i++;
  echo "The number is " . $i . "<br />";
  }
while ($i<=5);
?>

</body>
</html>
```

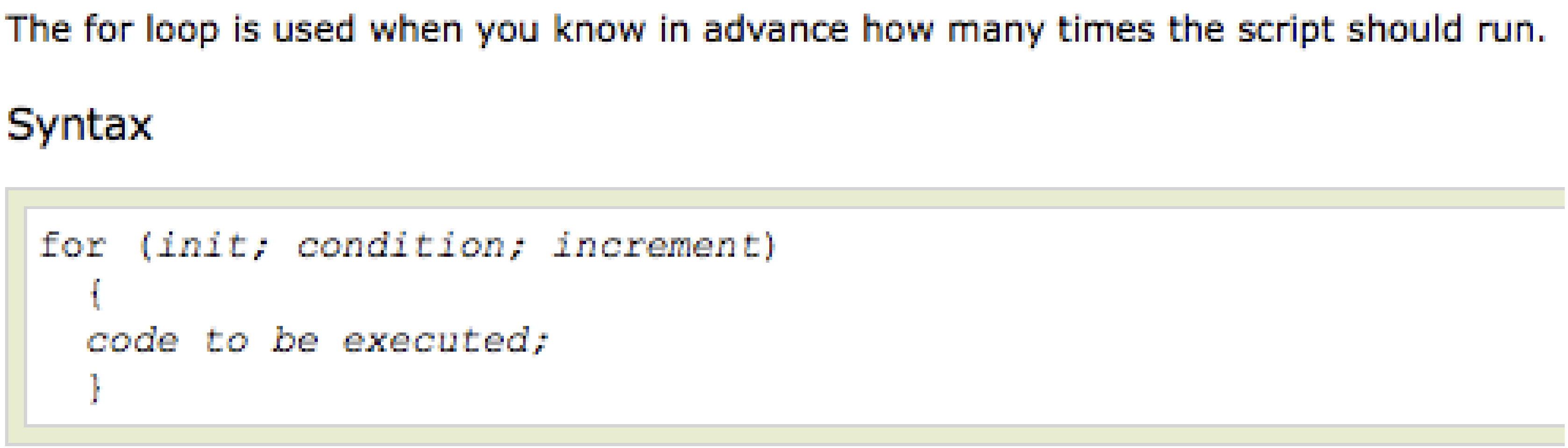
PHP Loops – Do … While



Output:

```
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
```

PHP Loops - For



The for loop is used when you know in advance how many times the script should run.

Syntax

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

PHP Loops - For

Parameters:

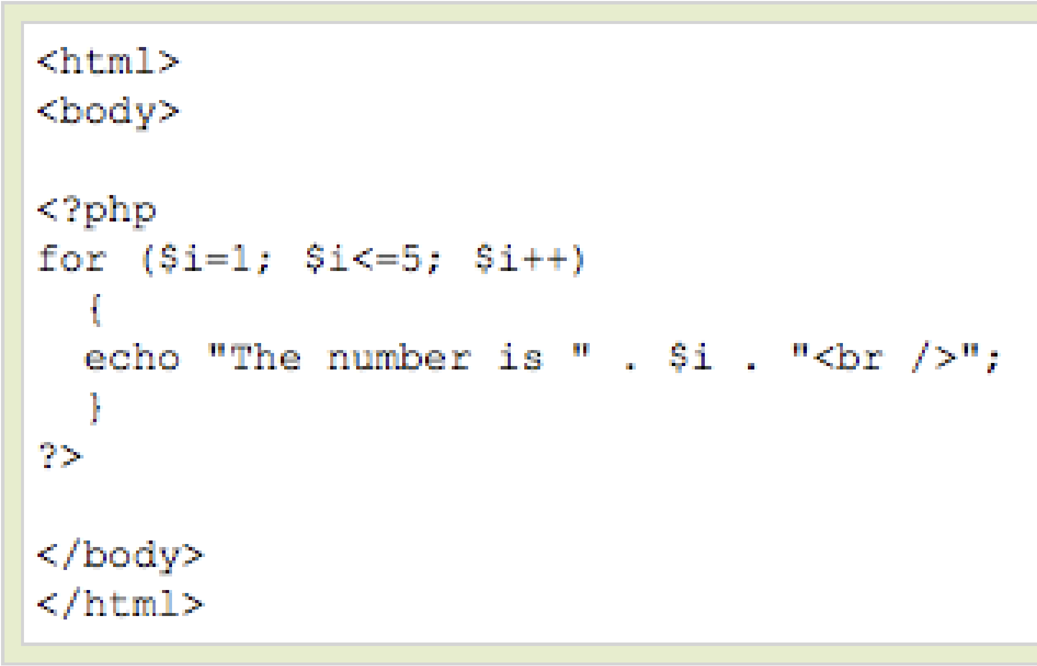
> **init**: Mostly used to set a counter (but can be any code to be executed once at the beginning of the loop)

> **condition**: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

> **increment**: Mostly used to increment a counter (but can be any code to be executed at the end of the loop)

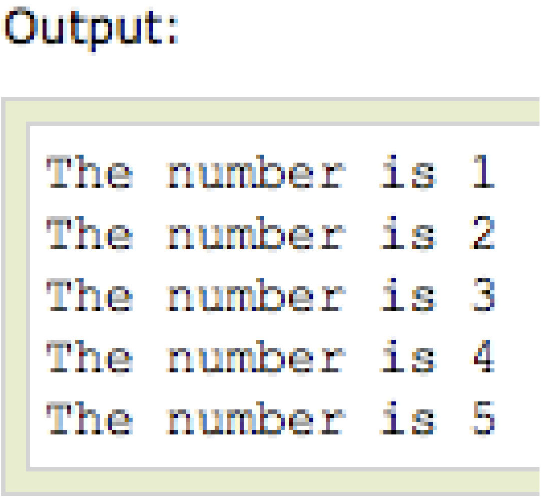
PHP Loops - For

The example below defines a loop that starts with i=1. The loop will continue to run as long as i is less than, or equal to 5. i will increase by 1 each time the loop runs:



```
<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
  {
  echo "The number is " . $i . "<br />";
  }
?>

</body>
</html>
```

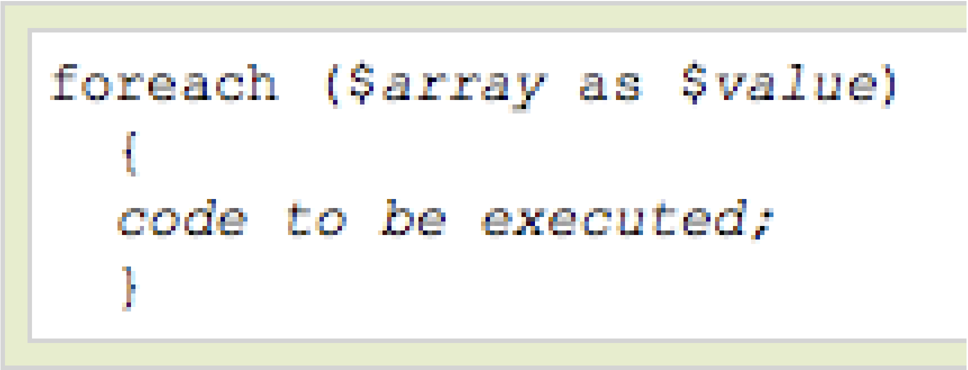
PHP Loops - For



Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
```

PHP Loops - Foreach

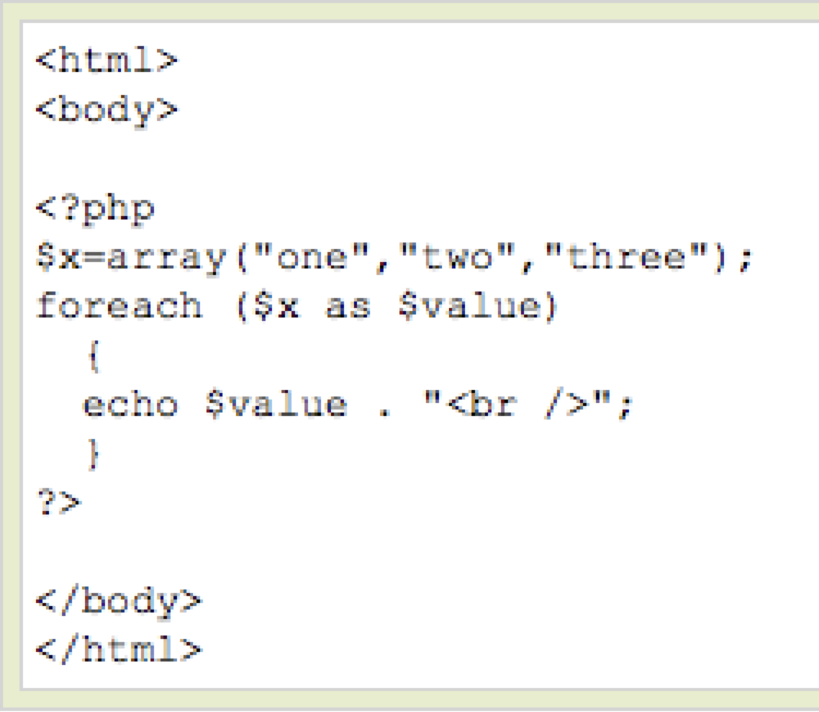


```
foreach ($array as $value)
  {
  code to be executed;
  }
```

For every loop iteration, the value of the current array element is assigned to $value (and the array pointer is moved by one) - so on the next loop iteration, you'll be looking at the next array value.

PHP Loops - Foreach

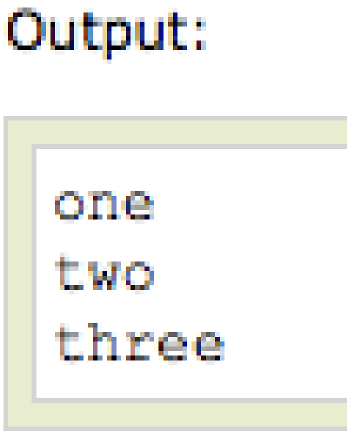
The following example demonstrates a loop that will print the values of the given array:



```
<html>
<body>

<?php
$x=array("one","two","three");
foreach ($x as $value)
  {
  echo $value . "<br />";
  }
?>

</body>
</html>
```

PHP Loops - Foreach

Winner of the most impressive slide award



Output:

```
one
two
three
```

PHP Functions

> We will now explore how to create your own functions.

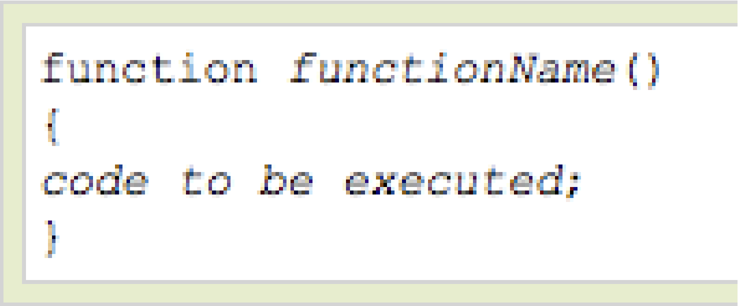
> To keep the script from being executed when the page loads, you can put it into a function.

> A function will be executed by a call to the function.

> You may call a function from anywhere within a page.

PHP Functions

A function will be executed by a call to the function.



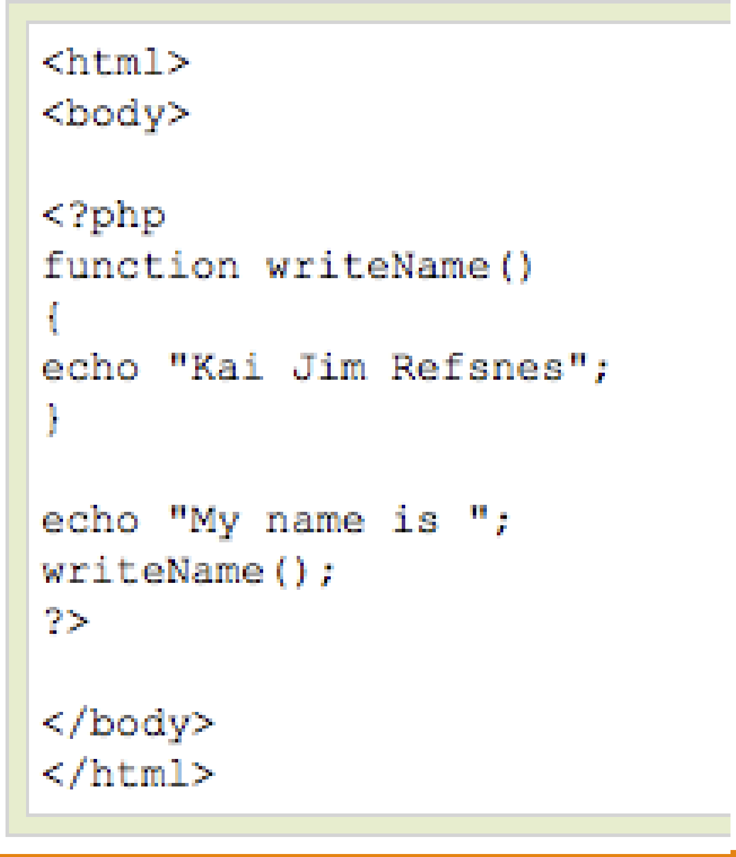
```
function functionName()
{
code to be executed;
}
```

> Give the function a name that reflects what the function does

> The function name can start with a letter or underscore (not a number)

PHP Functions

A simple function that writes a name when it is called:



```
<html>
<body>

<?php
function writeName()
{
echo "Kai Jim Refsnes";
}

echo "My name is ";
writeName();
?>

</body>
</html>
```

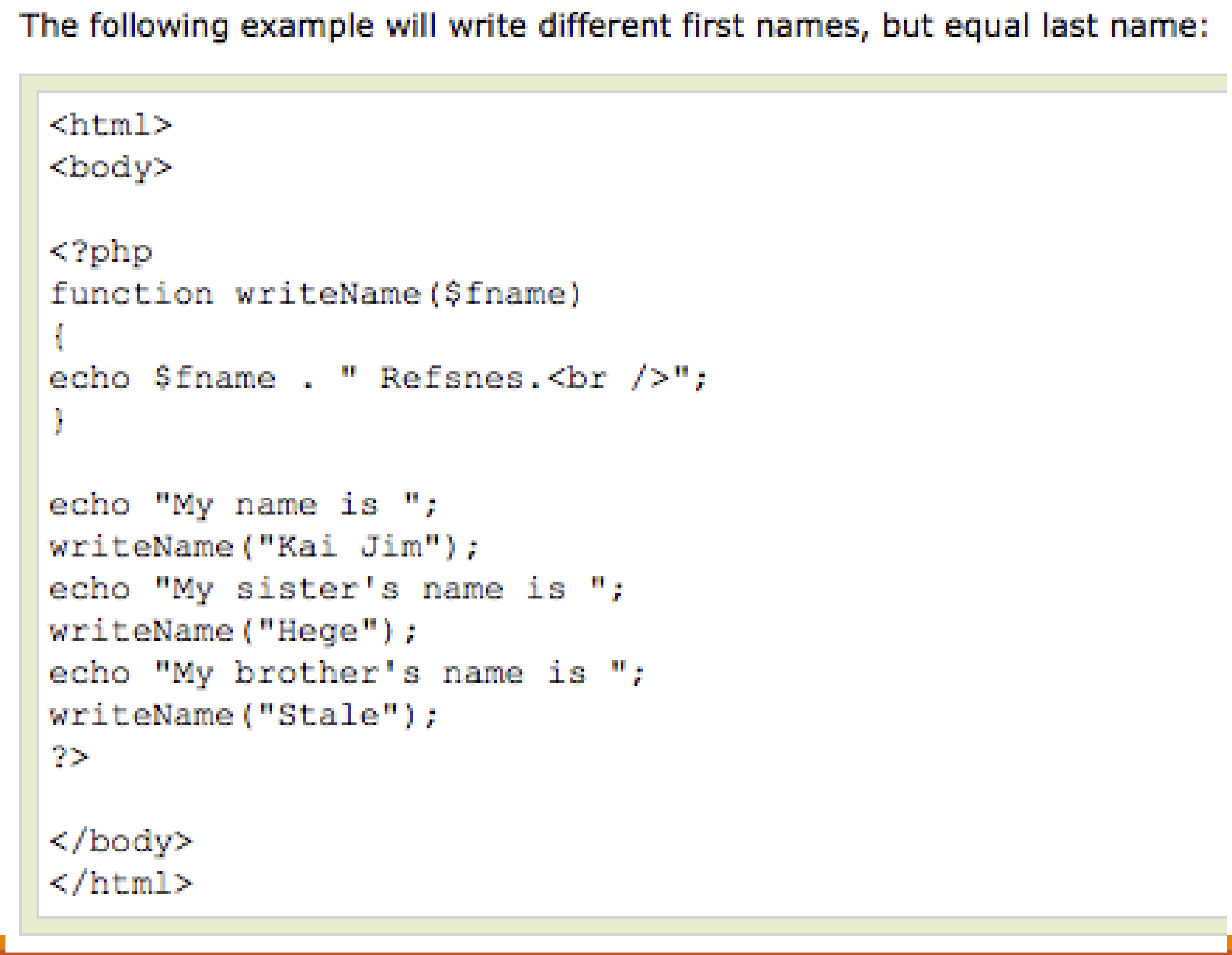
PHP Functions - Parameters

Adding parameters...

> To add more functionality to a function, we can add parameters. A parameter is just like a variable.

> Parameters are specified after the function name, inside the parentheses.

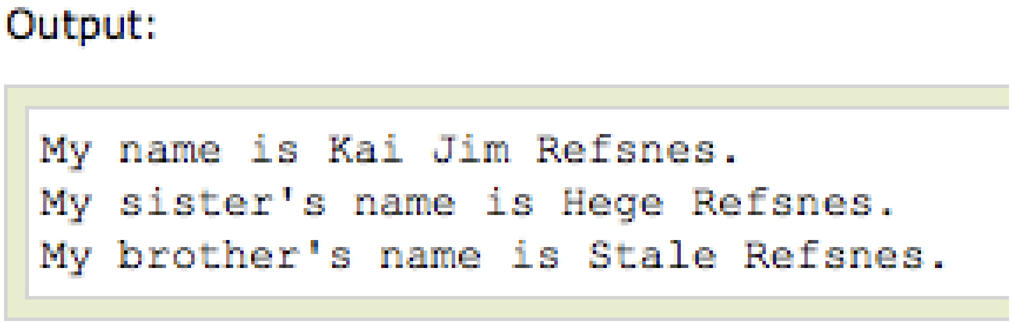
PHP Functions - Parameters



The following example will write different first names, but equal last name:

```
<html>
<body>

<?php
function writeName($fname)
{
echo $fname . " Refsnes.<br />";
}

echo "My name is ";
writeName("Kai Jim");
echo "My sister's name is ";
writeName("Hege");
echo "My brother's name is ";
writeName("Stale");
?>

</body>
</html>
```

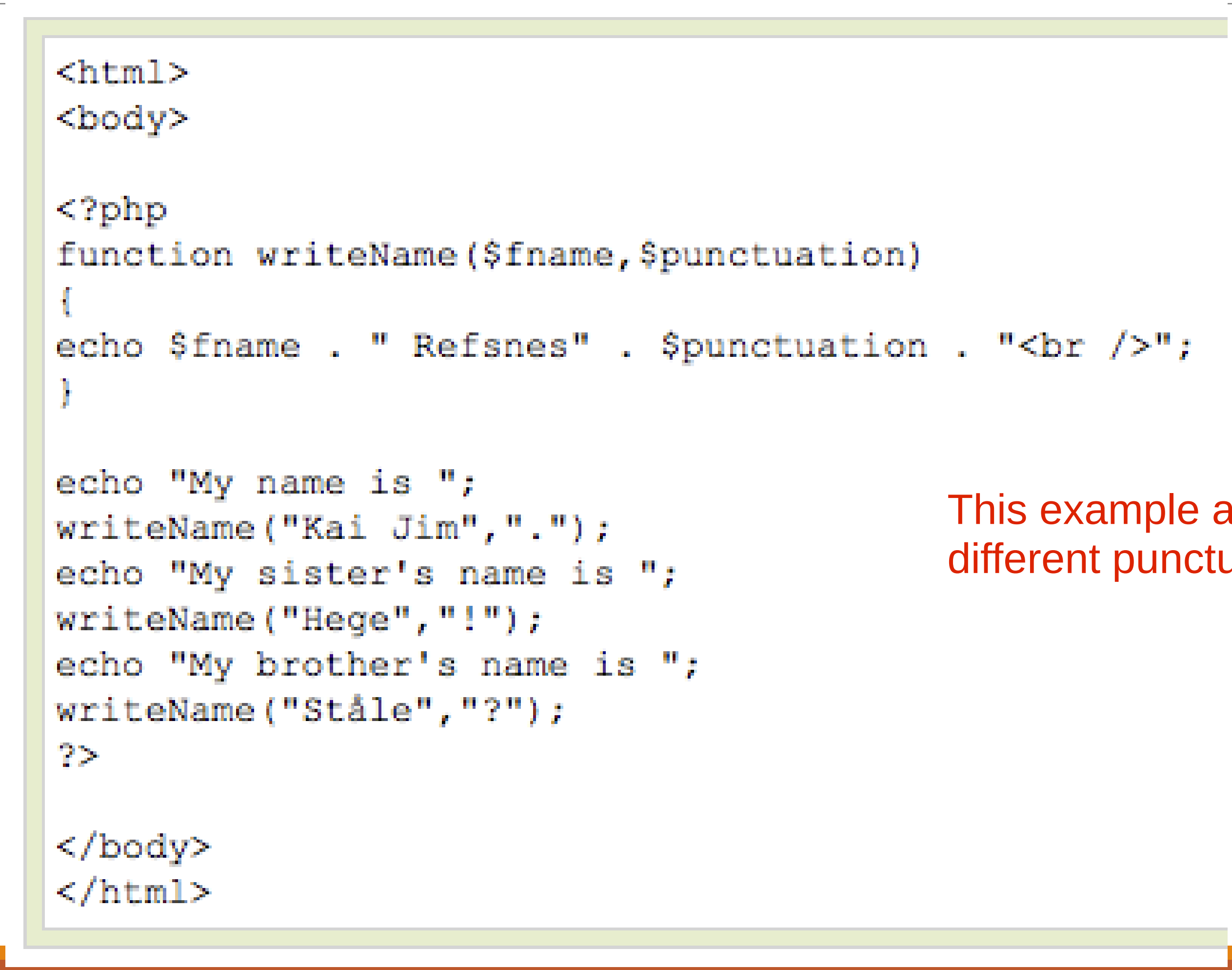
PHP Functions - Parameters



Output:

```
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes.
My brother's name is Stale Refsnes.
```

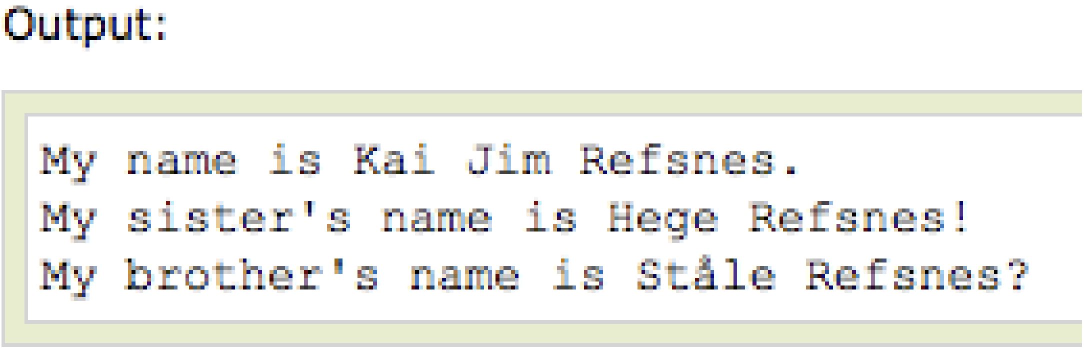
PHP Functions - Parameters



```
<html>
<body>

<?php
function writeName($fname,$punctuation)
{
echo $fname . " Refsnes" . $punctuation . "<br />";
}

echo "My name is ";
writeName("Kai Jim",".");
echo "My sister's name is ";
writeName("Hege","!");
echo "My brother's name is ";
writeName("Ståle","?");
?>

</body>
</html>
```

This example adds different punctuation.

PHP Functions - Parameters



Output:

```
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes!
My brother's name is Ståle Refsnes?
```

thank
YOU . . .
تشکر فراوان

Any Question??

Please write your questions and suggestions on the group!