# WEB APPLICATION DEVELOPMENT

## SERVER SIDE PROGRAMMING –CONDITIONAL STATEMENTS

PHP SCRIPTING LANGUAGE

MYSQL DATABASE

APACHE SERVER

# CONDITIONAL STATEMENTS

- In PHP we have the following conditional statements:
  - **if statement -** use this statement to execute some code only if a specified condition is true
  - **if...else statement -** use this statement to execute some code if a condition is true and another code if the condition is false
  - **if...elseif....else statement -** use this statement to select one of several blocks of code to be executed
  - **switch statement -** use this statement to select one of many blocks of code to be executed

# THE IF STATEMENT

- Use the if statement to execute some code only if a specified condition is true.
- **Syntax**

```
if (condition) code to be executed if condition is true;
```

- Example

  Write a PHP Program If the date is Friday print Have a nice weekend on screen.

# EXAMPLE

```
<html>
<body>
         <?php
        $d=date("D");
        if ($d=="Fri") echo "Have a nice weekend!";
         ?>
</body>
</html>
```

# THE IF …ELSE STATEMENT

- **Syntax**

```
if (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

- Example

- Write PHP coding if it is hot then turn on the air conditioner otherwise turn on heater.

# EXAMPLE

```
<html>
<body>
      <?php
    if ($a==“hot") {

         echo “Turn on air-conditioner";

    }

    else  {

        echo “Turn on heater”; }

      ?>
</body>
</html>
```

# THE IF…ELSE IF…ELSE STATEMENT

- **Syntax**

```
if (condition)
  code to be executed if condition is true;
else if (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

- Write PHP coding if it is hot then turn on the air conditioner, for warm weather give a message Enjoy the weather otherwise turn on heater.

# EXAMPLE

```
<html>
<body>
        <?php


          if ($a==“hot") {

                echo “Turn on air-conditioner";

          }

          if  else ($a==“warm”) {

               echo “Enjoy the weather”;

          }

          else  {

               echo “Turn on heater”;
          }

        ?>
</body>
</html>
```

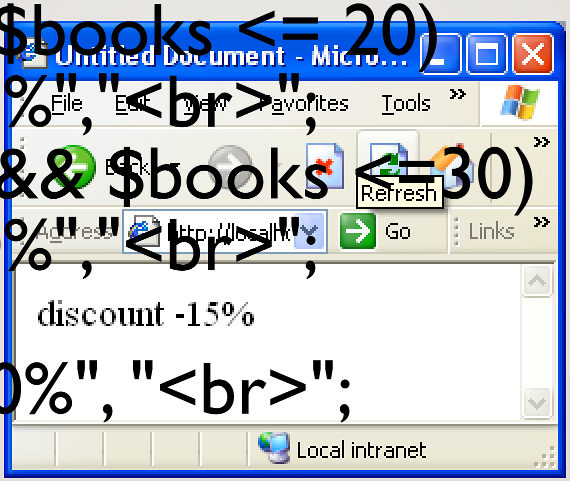
# EXERCISE

- Write a program to show  discount according to number of books.(Number of books =15)
    - 10<=books<=20   Discount-15%
    - 20<books<=30   Discount-20%
    - books>30   Discount-30%

# ANSWER

```
<?php
$books=15;
if($books > 10 &&  $books <= 20)
 echo " discount -15%", "<br>";
 elseif (20 >$books && $books <=30)
 echo " discount -20%", "<br>";
   else
  echo " discount -30%", "<br>";

?>
```

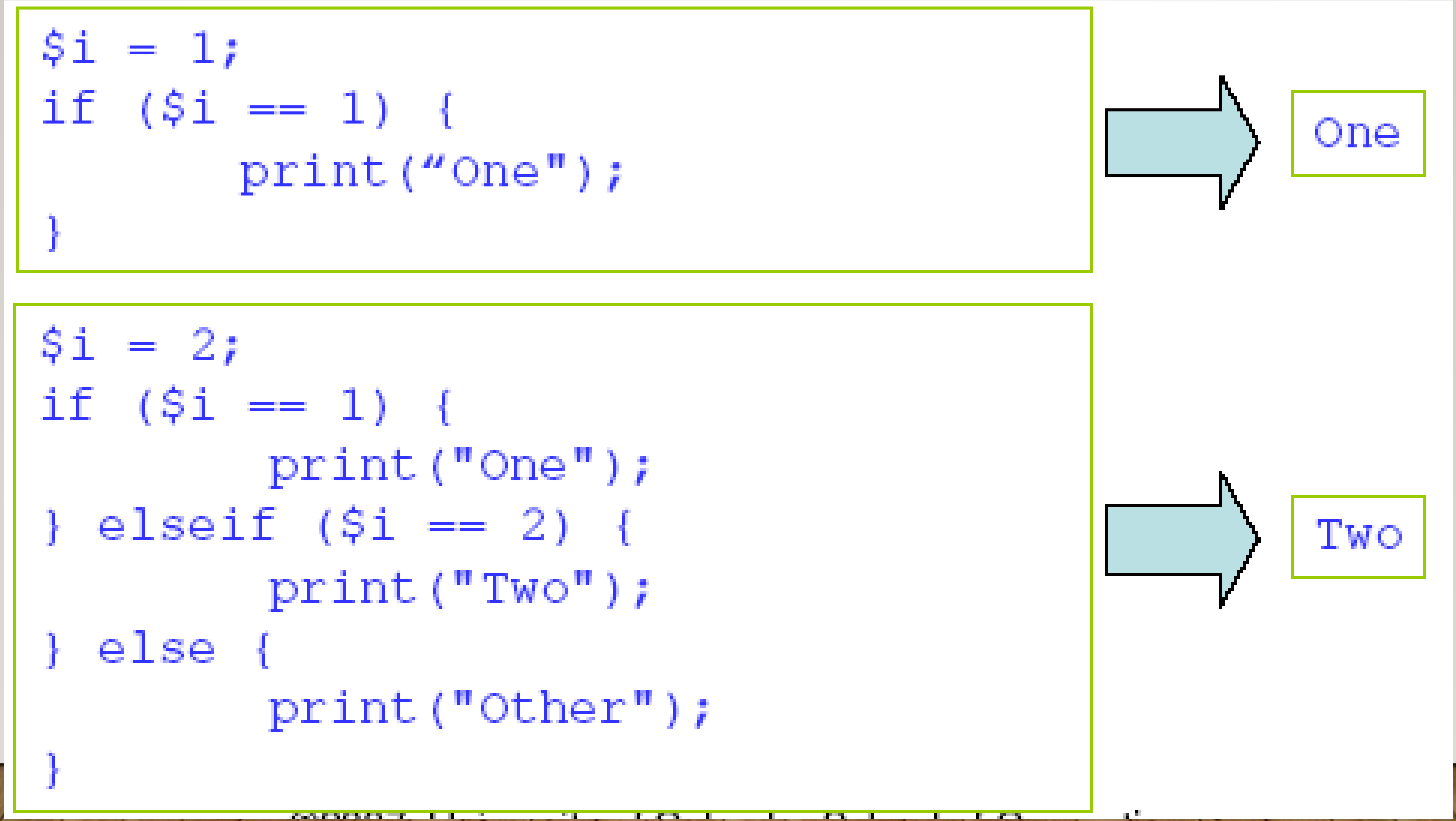


```
discount -15%
```

# EXERCISE

- Write a PHP program to check whether value of assigned variable is one.
- Modify above PHP program to check whether value of assigned variable is 1- one or 2- two or 3-three).

# ANSWER



```
$i = 1;
if ($i == 1) {
        print(“One");
}
```

One

```
$i = 2;
if ($i == 1) {
        print("One");
} elseif ($i == 2) {
        print("Two");
} else {
        print("Other");
}
```

Two

# THE SWITCH STATEMENT

- **Syntax**

```
switch (n)
{
      case label1:
     code to be executed if n=label1;
     break;
     case label2:
     code to be executed if n=label2;
      break;
     default:
      code to be executed if n is different from both label1  and label2;
}
```

# TRY

```
<html>
<head>
<title>switch</title>
 </head>
<body>
<?php
 $mood = "sad";
switch ( $mood )
{
 case "happy": print "Hooray, I'm in a good mood"; break;
case "sad": print "Awww. Don't be down!"; break;
 default: print "Neither happy nor sad but $mood";
 }
 ?>
</body>
 </html>
```

# EXERCISE

- Write PHP code for finding the number of days in a month.
- Write PHP code for display the day  when give  as short name of day.(ex Mon -> Monday)

# ANSWER

```
<?php
            /* write PHP code for finding the number of days in a
month*/
            $a= 3;
            switch ($a)
            {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
echo "The number of days = 31";
break;
case 2:
echo "The number of days = 28";
break;
case 4:
case 6:
case 9:
case 11:
echo "The number of days = 30";
break;
default:
echo "Error";
break;
            }
?>
```

# ANSWER

```
<?php
$day="Thu";
switch($day){
case "Mon":{
print("Monday");
break;}
case "Tue":{
print("Tuesday");
break;}
case "Wed":{
print("Wednessday");
break;}
case "Thu":{
print("Thursday");
break;}
case "Fri":{
print("Friday");
break;}
case "Sat":{
print("Saturday");
break;}
case "Sun":{
print("Sunday");
break;}
default: {
print("error");}
}
?>
```

# PHP LOOPING STATEMENTS

- Loops execute a block of code a specified number of times, or while a specified condition is true.
- In PHP, we have the following looping statements:
  - **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

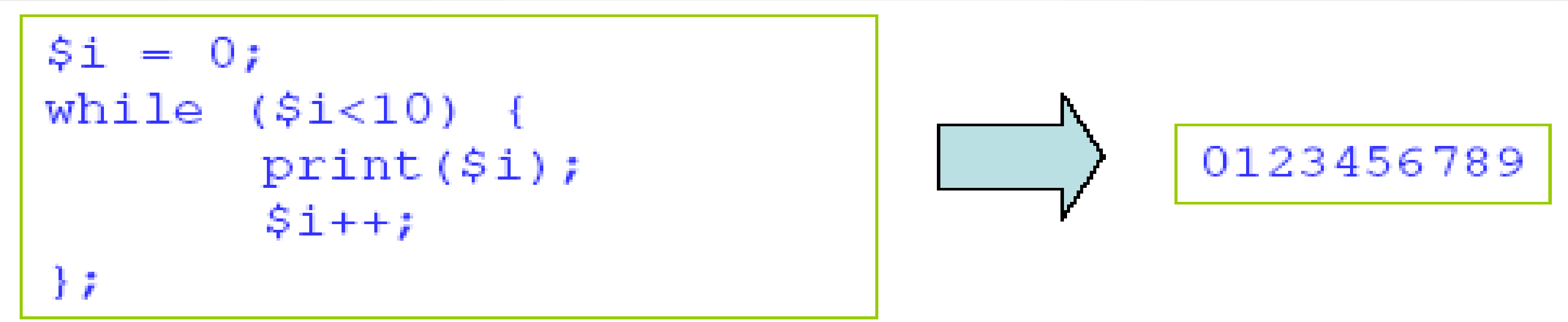
# WHILE LOOP

- The while loop executes a block of code while a condition is true.
- **Syntax**

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

- Write a PHP coding to print 0 – 9 digits on the screen.

# ANSWER



```
$i = 0;
while ($i<10) {
      print($i);
      $i++;
};
```

```
0123456789
```

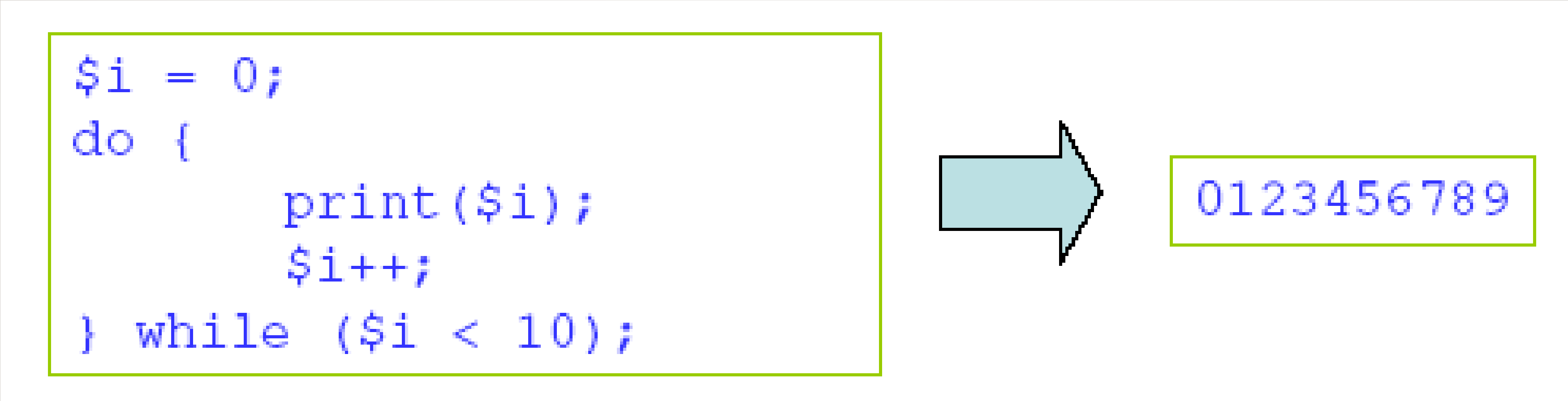
# THE DO… WHILE STATEMENT

- 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.
- **Syntax**

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

- Write code for print 1-9 digits using do…while loop.

# ANSWER



```
$i = 0;
do {
        print($i);
        $i++;
} while ($i < 10);
```

```
0123456789
```

- Note: Due to the expression being evaluated at the end of the structure, statement(s) in the do…while loop are executed at least once, even if the condition is false

# FOR LOOP

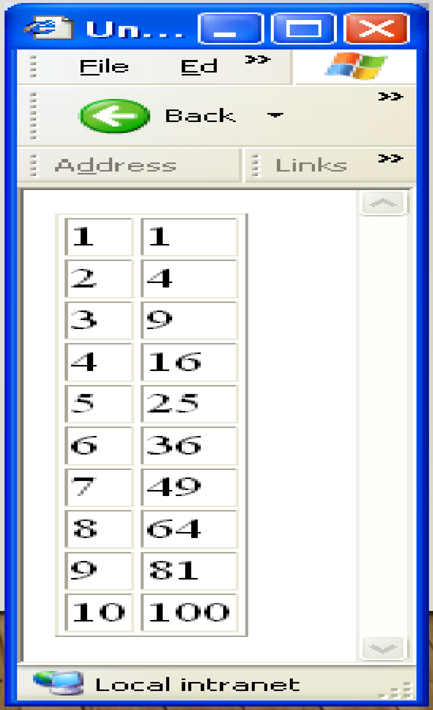
- Loops execute a block of code a specified number of times, or while a specified condition is true.
- **Syntax**

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

- 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)

# EXERCISE

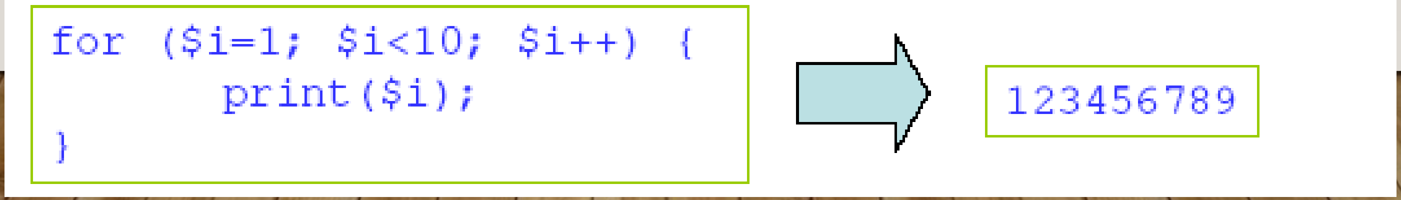
- write code to print 0 to 9 using for loop.
- Write a program to display  following using for loop.(square of 1 to 10)



| | |
|---|---|
| 1 | 1 |
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |
| 6 | 36 |
| 7 | 49 |
| 8 | 64 |
| 9 | 81 |
| 10 | 100 |

# ANSWER

```
<?php
    /* write code to print 0 to 9 using for loop. */
      $a=0;
      for ($a=0;$a<10;$a++)
      {
            echo $a."<br>";
      }
?>
```

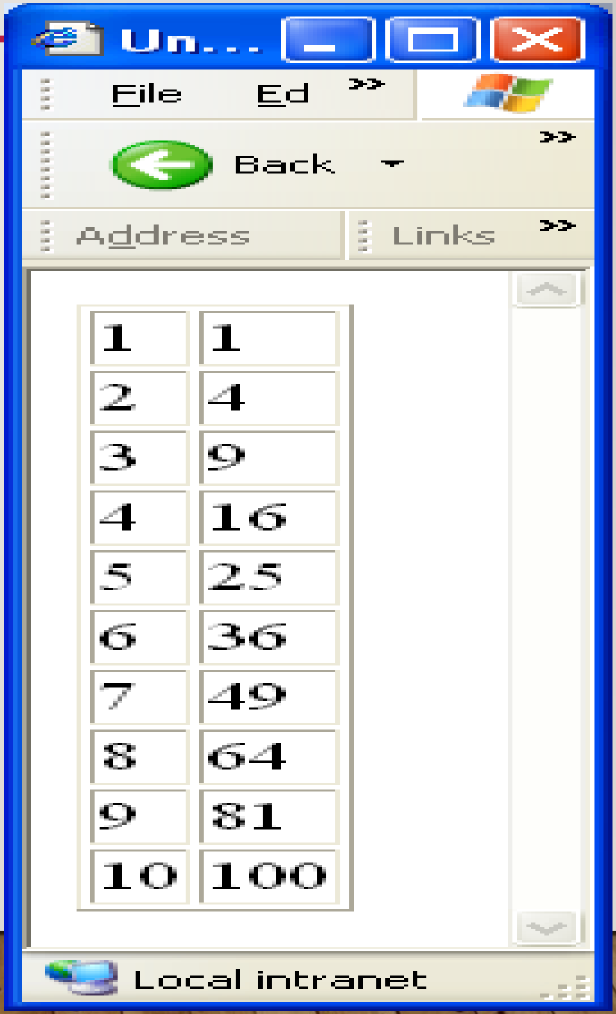


```
for ($i=1; $i<10; $i++) {
        print($i);
}
```

```
123456789
```

# TRY

```
<?php
for ($i=1;$i<11;$i++)
{
 $j=$i*$i;
?>
<tr>
  <td><?echo $i?></td>
   <td><?echo $j ?></td>
</tr>
<?
}
?>
```



| | |
|---|---|
| 1 | 1 |
| 2 | 4 |
| 3 | 9 |
| 4 | 16 |
| 5 | 25 |
| 6 | 36 |
| 7 | 49 |
| 8 | 64 |
| 9 | 81 |
| 10 | 100 |

# FOREACH LOOP

- The foreach loop is used to loop through arrays.
- **Syntax**

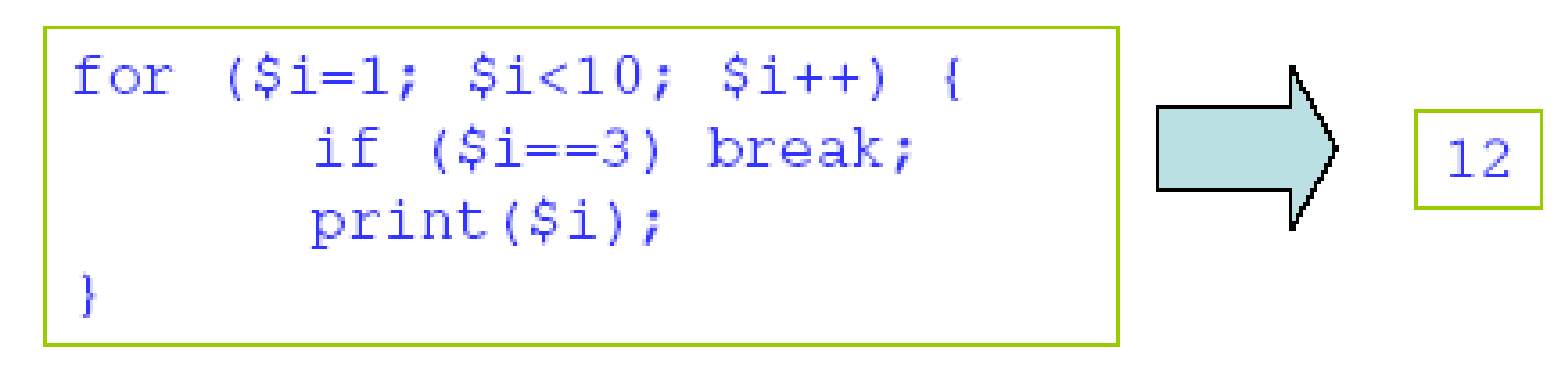
```
foreach ($array as $value)
  {
        code to be executed;
  }
```

# EXAMPLE

```
<?php
    $x=array(“Monday",“ Tuesday ",” Wednesday “,
“Thursday”, “Friday”);
     foreach ($x as $value)
     {
          echo $value . "<br />";
     }
?>
```

# BREAK AND CONTINUE

- The break command will break the loop and continue executing the code that follows after the loop (if any).
- E.g.

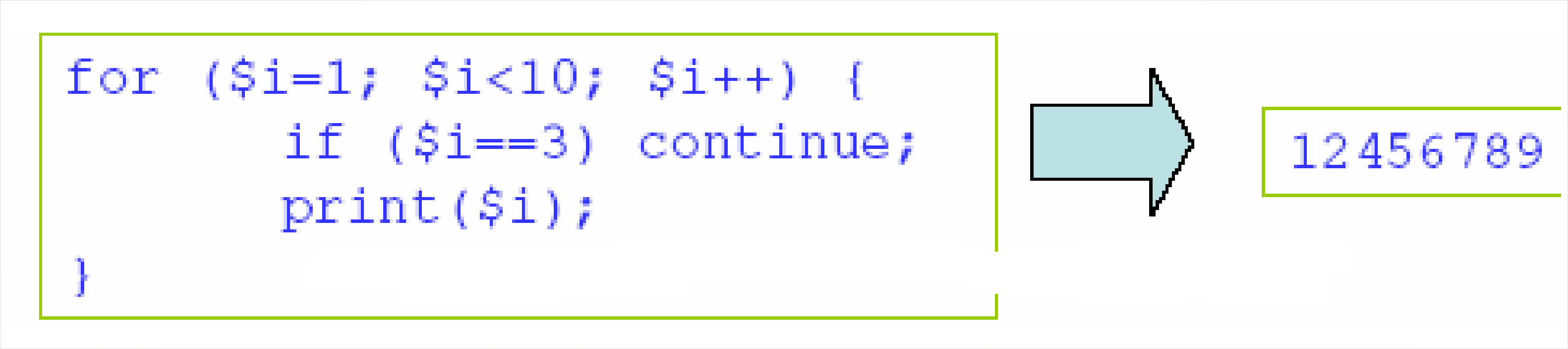


```
for ($i=1; $i<10; $i++) {
        if ($i==3) break;
        print($i);
}
```

```
12
```

# BREAK AND CONTINUE

- The continue command will break the current loop and continue with the next value.
- E.g.



```
for ($i=1; $i<10; $i++) {
        if ($i==3) continue;
        print($i);
}
```

```
12456789
```

# USING THE ? OPERATOR / CONDITIONAL OPERATOR

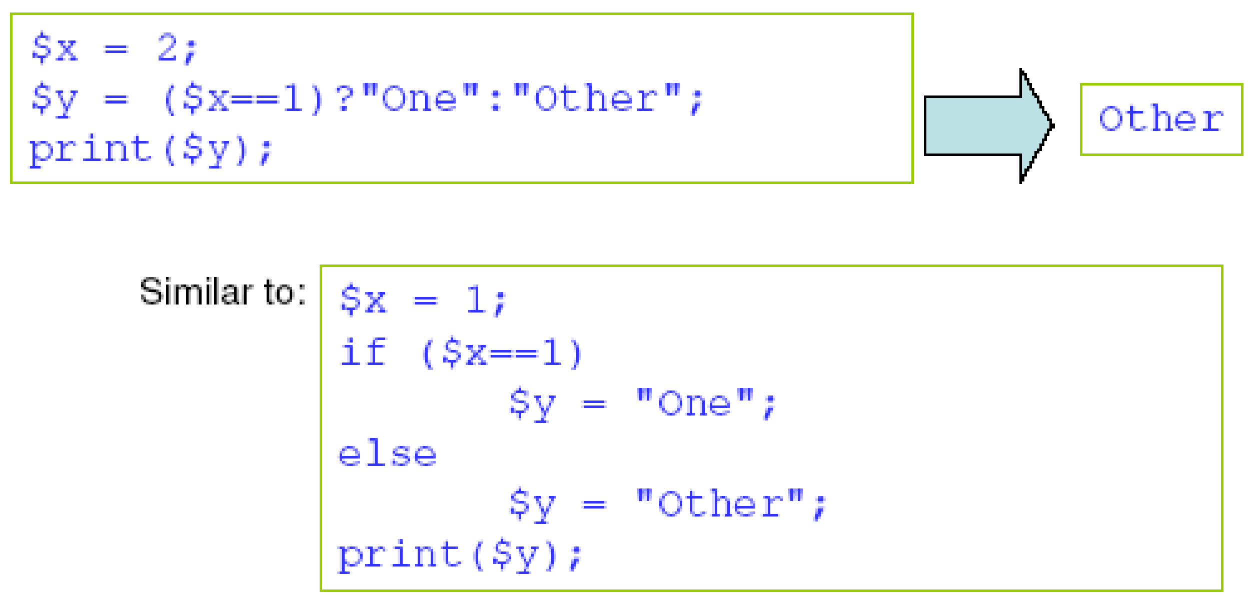
- The ? ~~or ternary operator is similar to the if statement~~ but returns a value derived from one of two expressions separated by a colon. Which expression is used to generate the value returned depends on the result of a test expression

```
(expression)?returned_if_expression_is_true:returned_if_expression_is_false;
```

- If the test expression evaluates to true, the result of the second expression is returned; otherwise, the value of the third expression is returned.

# CONDITIONAL OPERATOR

- PHP also contains a conditional operator that assigns a value to a variable based on some condition.
- E.g.



```
$x = 2;
$y = ($x==1)?"One":"Other";
print($y);
```

```
Other
```

Similar to:

```
$x = 1;
if ($x==1)
        $y = "One";
else
        $y = "Other";
print($y);
```

# TRY

```
<html>
 <head>
<title>Listing 5.5</title>
 </head>
 <body>
 <?php
$mood = "sad";
 $text = ( $mood=="happy" )?"Hooray, I'm in a good mood":"Not happy but $mood";
 print "$text";
?>
</body>
 </html>
```

**$mood is set to "sad". $mood is tested for equivalence to the string "happy". Because this test returns false, the result of the third of the three expressions is returned.The ternary operator can be difficult to read but is useful if you are dealing with only two alternatives and like to write compact code.**