

Unit-1: Getting Up and Running, PHP Language Structure

Getting Up and Running: Installation Quick Start Guide with XAMPP5 - Installing and Configuring MySQL - Installing and Configuring Apache - Installing and Configuring PHP - PHP Language Structure: The Building Blocks of PHP - Flow Control Functions in PHP - Working with Functions - Working with Arrays - Working with Objects

1. Installation of XAMPP server

Installing XAMPP on Windows

The latest version can be downloaded from <http://www.apachefriends.org/en/xamppwindows.html>, double-click the file to launch the wizard-based installer program.

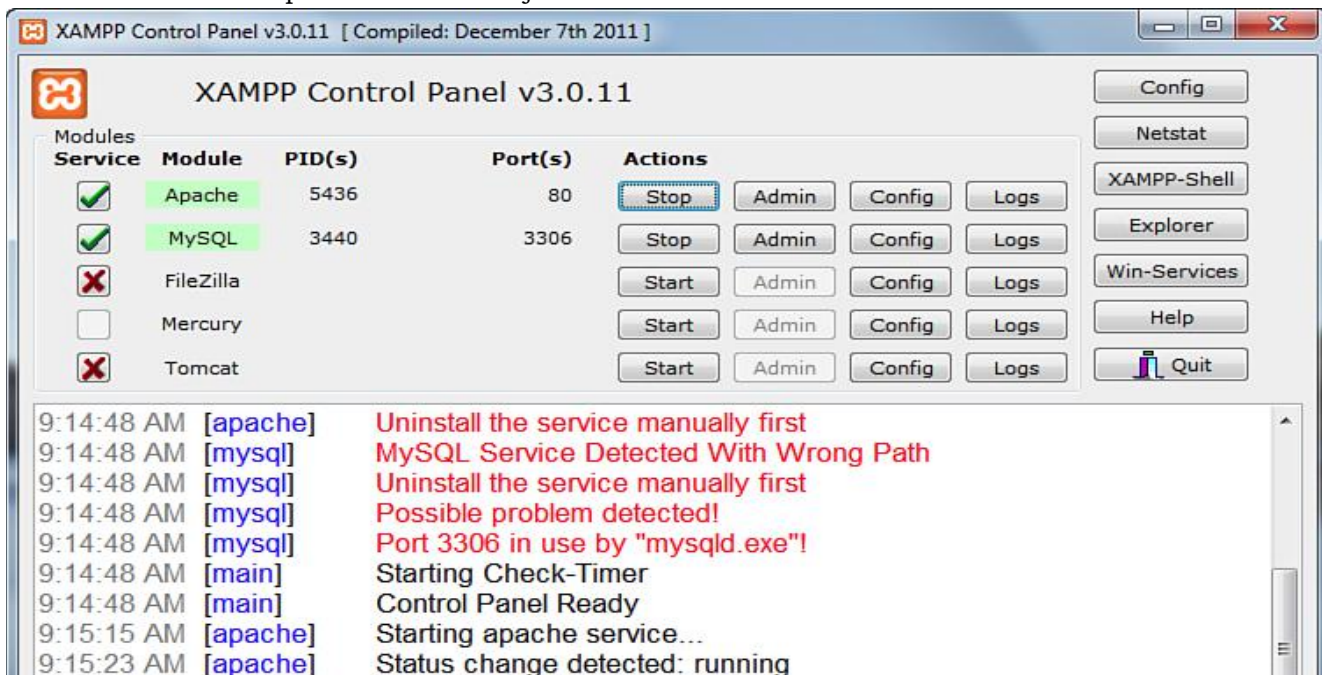
You are first asked to select your language; English is the default selection. After selecting your language and clicking the OK button, you will see the welcome screen of the installer program.



Click the Next button to continue the installation process. As with most wizard-like installations, you are asked to select an installation location and some installation options before moving to the next step.

When the installation process finishes, the installer alerts you to this status; click the Finish button to complete the installation. Before the XAMPP installation process completely closes, it asks whether you want to start the Control Panel for managing the installed services.

The XAMPP Control Panel, provides you with one-click access to starting and stopping the Apache and MySQL server processes running on your machine. If you are running these server processes on your local machine for development purposes only, you might want to turn them on only when you need them; the Control Panel allows quick access to do just that.



To test whether the web server is running, open a web browser and enter **http://localhost/xampp/xampp.php**.

2. Write PHP code to print Hello World program

```
<!DOCTYPE html>
<html>
<head>
<title>A PHP script including HTML</title>
</head>
<body>
<h1><?php echo "hello world"; ?></h1>
</body>
</html>
```

3. Demonstrate 8 basic data types in PHP.

```
<?php
$testing; // declare without assigning
echo "is null? ".is_null($testing); // checks if null
echo "<br/>";
$testing = 5;
echo "is an integer? ".is_int($testing); // checks if integer
echo "<br/>";
$testing = "five";
echo "is a string? ".is_string($testing); // checks if string
echo "<br/>";
$testing = 5.024;
echo "is a double? ".is_double($testing); // checks if double
echo "<br/>";
$testing = true;
echo "is boolean? ".is_bool($testing); // checks if boolean
echo "<br/>";
$testing = array('apple', 'orange', 'pear');
echo "is an array? ".is_array($testing); // checks if array
echo "<br/>";
echo "is numeric? ".is_numeric($testing); // checks if is numeric
echo "<br/>";
echo "is a resource? ".is_resource($testing); // checks if is a resource
echo "<br/>";
echo "is an array? ".is_array($testing); // checks if is an array
echo "<br/>";
?>
```

Changing the Type of a Variable with settype()

```
<?php
$undecided = 3.14;
echo "is ".$undecided." a double? ".is_double($undecided)."<br/>"; // double
settype($undecided, 'string');
echo "is ".$undecided." a string? ".is_string($undecided)."<br/>"; // string
settype($undecided, 'integer');
echo "is ".$undecided." an integer? ".is_integer($undecided)."<br/>"; //integer
settype($undecided, 'double');
echo "is ".$undecided." a double? ".is_double($undecided)."<br/>"; // double
settype($undecided, 'bool');
```

```
echo "is ".$undecided." a boolean? ".is_bool($undecided)."<br/>"; // boolean
?>
```

Casting a Variable

```
<?php
$undecided = 3.14;
$holder = (double) $undecided;
echo "is ".$holder." a double? ".is_double($holder)."<br/>"; // double
$holder = (string) $undecided;
echo "is ".$holder." a string? ".is_string($holder)."<br/>"; // string
$holder = (integer) $undecided;
echo "is ".$holder." an integer? ".is_integer($holder)."<br/>"; // integer
$holder = (double) $undecided;
echo "is ".$holder." a double? ".is_double($holder)."<br/>"; // double
$holder = (boolean) $undecided;
echo "is ".$holder." a boolean? ".is_bool($holder)."<br/>"; // boolean
echo "<hr/>";
echo "original variable type of $undecided: ";
echo gettype($undecided); // double
?>
```

Defining and Accessing a Constant

```
<?php
define("THE_YEAR", "2012");
echo "It is the year ".THE_YEAR;
?>
```

4. Demonstrate the scope of variables declared in PHP code.

```
<?php
// These are all valid declarations
$val = 5;
$val2 = 2;
$x_Y = "gfg";
$_X = "GeeksforGeeks";
// This is an invalid declaration as it begins with a number
$10_val = 56;
// This is also invalid as it contains special character other than _
$f.d = "num";
?>
```

Prog to demonstrate local variable

```
<?php
$num = 60;
function local_var()
{
//This $num is local to this function the variable $num outside this function is a completely different variable
    $num = 50;
    echo "local num = $num \n";
}
local_var();
// $num outside function local_var() is a completely different Variable than that of inside local_var()
echo "Variable num outside local_var() is $num \n";
```

?>

Function to demonstrate use of global variable

```
<?php
$num = 20;
// function to demonstrate use of global variable
function global_var()
{
    // we have to use global keyword before the variable $num to access within the function
    global $num;
    echo "Variable num inside function : $num \n";
}
global_var();
echo "Variable num outside function : $num \n";
?>
```

```
<?php
// function to demonstrate static variables
function static_var()
{
    // static variable
    static $num = 5;
    $sum = 2;
    $sum++;
    $num++;
    echo $num, "\n";
    echo $sum, "\n";
}
// first function call
static_var();
// second function call
static_var();
?>
```

PHP \$ and \$\$ Variables

The \$var (single dollar) is a normal variable with the name var that stores any value like string, integer, float, etc.

The \$\$var (double dollar) is a reference variable that stores the value of the \$variable inside it.

```
<?php
$x = "abc";
$$x = 200;
echo $x."<br/>";
echo $$x."<br/>";
echo $abc;
?>
```

```
<?php
$x="U.P";
$$x="Lucknow";
echo $x. "<br>";
echo $$x. "<br>";
echo "Capital of $x is " . $$x;
```

```
?>
```

```
<?php
$name="Cat";
${$name}="Dog";
${${$name}}="Monkey";
echo $name. "<br>";
echo ${$name}. "<br>";
echo $Cat. "<br>";
echo ${${$name}}. "<br>";
echo $Dog. "<br>";
?>
```

PHP constant: define()

Use the define() function to create a constant. It defines constant at run time.

```
<?php
define("MESSAGE","Hello AITS PHP");
echo MESSAGE;
?>
```

```
<?php
define("MESSAGE","Hello JavaTpoint PHP",true); //not case sensitive
echo MESSAGE, "</br>";
echo message;
?>
```

PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function. The constant defined using const keyword are case-sensitive.

```
<?php
const MESSAGE="Hello const by AITS PHP";
echo MESSAGE;
?>
```

Constant() function

There is another way to print the value of constants using constant() function instead of using the echo statement.

```
<?php
define("MSG", "JavaTpoint");
echo MSG, "</br>";
echo constant("MSG");
//both are similar
?>
```

5. Demonstrate Arithmetic, Comparison, Logical (or Relational), Assignment and Conditional (or ternary) Operators.

```
<html>
```

```
<head>
<title>Arithmetical Operators</title>
</head>
```

```
<body>
```

```

<?php
    $a = 42;
    $b = 20;
    $c = $a + $b;
    echo "Addition Operation Result: $c <br/>";
    $c = $a - $b;
    echo "Substraction Operation Result: $c <br/>";
    $c = $a * $b;
    echo "Multiplication Operation Result: $c <br/>";
    $c = $a / $b;
    echo "Division Operation Result: $c <br/>";
    $c = $a % $b;
    echo "Modulus Operation Result: $c <br/>";
    $c = $a++;
    echo "Increment Operation Result: $c <br/>";
    $c = $a--;
    echo "Decrement Operation Result: $c <br/>";
?>
</body>
</html>

<html>
    <head>
        <title>Comparison Operators</title>
    </head>
    <body>
        <?php
            $a = 42;
            $b = 20;
            if( $a == $b ) {
                echo "TEST1 : a is equal to b<br/>";
            }else {
                echo "TEST1 : a is not equal to b<br/>";
            }
            if( $a > $b ) {
                echo "TEST2 : a is greater than b<br/>";
            }else {
                echo "TEST2 : a is not greater than b<br/>";
            }
            if( $a < $b ) {
                echo "TEST3 : a is less than b<br/>";
            }else {
                echo "TEST3 : a is not less than b<br/>";
            }
            if( $a != $b ) {
                echo "TEST4 : a is not equal to b<br/>";
            }else {
                echo "TEST4 : a is equal to b<br/>";
            }
            if( $a >= $b ) {
                echo "TEST5 : a is either greater than or equal to b<br/>";
            }else {
                echo "TEST5 : a is neither greater than nor equal to b<br/>";
            }
        </?php>
    </body>
</html>

```

```

}
if( $a <= $b ) {
    echo "TEST6 : a is either less than or equal to b<br/>";
}else {
    echo "TEST6 : a is neither less than nor equal to b<br/>";
}
?>
</body>
</html>

```

```

<html>
<head>
<title>Logical Operators</title>
</head>
<body>
<?php
    $a = 42;
    $b = 0;
    if( $a && $b ) {
        echo "TEST1 : Both a and b are true<br/>";
    }else{
        echo "TEST1 : Either a or b is false<br/>";
    }
    if( $a and $b ) {
        echo "TEST2 : Both a and b are true<br/>";
    }else{
        echo "TEST2 : Either a or b is false<br/>";
    }
    if( $a || $b ) {
        echo "TEST3 : Either a or b is true<br/>";
    }else{
        echo "TEST3 : Both a and b are false<br/>";
    }
    if( $a or $b ) {
        echo "TEST4 : Either a or b is true<br/>";
    }else {
        echo "TEST4 : Both a and b are false<br/>";
    }
    $a = 10;
    $b = 20;
    if( $a ) {
        echo "TEST5 : a is true <br/>";
    }else {
        echo "TEST5 : a is false<br/>";
    }
    if( $b ) {
        echo "TEST6 : b is true <br/>";
    }else {
        echo "TEST6 : b is false<br/>";
    }
    if( !$a ) {
        echo "TEST7 : a is true <br/>";
    }else {

```

```

        echo "TEST7 : a is false<br/>";
    }
    if( !$b ) {
        echo "TEST8 : b is true <br/>";
    }else {
        echo "TEST8 : b is false<br/>";
    }
?>
</body>
</html>

```

```

<html>
<head>
<title>Assignment Operators</title>
</head>
<body>
<?php
    $a = 42;
    $b = 20;
    $c = $a + $b;
    echo "Addition Operation Result: $c <br/>";
    $c += $a;
    echo "Add AND Assignment Operation Result: $c <br/>";
    $c -= $a;
    echo "Subtract AND Assignment Operation Result: $c <br/>";
    $c *= $a;
    echo "Multiply AND Assignment Operation Result: $c <br/>";
    $c /= $a;
    echo "Division AND Assignment Operation Result: $c <br/>";
    $c %= $a;
    echo "Modulus AND Assignment Operation Result: $c <br/>";
?>
</body>
</html>

```

```

<html>
<head>
<title>Arithmetical Operators</title>
</head>
<body>
<?php
    $a = 10;
    $b = 20;
    /* If condition is true then assign a to result otheriwse b */
    $result = ( $a > $b ) ? $a : $b;
    echo "TEST1 : Value of result is $result<br/>";
    /* If condition is true then assign a to result otheriwse b */
    $result = ( $a < $b ) ? $a : $b;
    echo "TEST2 : Value of result is $result<br/>";
?>
</body>
</html>

```


6. Demonstrate if, elseif ...else and switch statements.

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

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

```
<html>
<body>
  <?php
    $d = date("D");
    switch ($d){
      case "Mon":
        echo "Today is Monday";
        break;
      case "Tue":
        echo "Today is Tuesday";
        break;
      case "Wed":
        echo "Today is Wednesday";
        break;
      case "Thu":
        echo "Today is Thursday";
        break;
      case "Fri":
        echo "Today is Friday";
        break;
      case "Sat":
        echo "Today is Saturday";
        break;
      case "Sun":
        echo "Today is Sunday";
        break;
    }
  ?>
</body>
</html>
```

```

        default:
            echo "Wonder which day is this ?";
    }
?>
</body>
</html>

```

7. Demonstrate for, while, do – while, and for each loop.

```

<html>
<body>
<?php
$a = 0;
$b = 0;
for( $i = 0; $i<5; $i++ ) {
    $a += 10;
    $b += 5;
}
echo ("At the end of the loop a = $a and b = $b" );
?>
</body>
</html>

```

```

<html>
<body>
<?php
$i = 0;
$num = 50;
while( $i < 10) {
    $num--;
    $i++;
}
echo ("Loop stopped at i = $i and num = $num" );
?>
</body>
</html>

```

```

<html>
<body>
<?php
$i = 0;
$num = 0;
do {
    $i++;
}
while( $i < 10 );
echo ("Loop stopped at i = $i" );
?>
</body>
</html>

```

```

<html>
<body>
<?php

```

```

    $array = array( 1, 2, 3, 4, 5);
    foreach( $array as $value ) {
        echo "Value is $value <br />";
    }
?>
</body>
</html>

```

```

<html>
<body>
<?php
    $i = 0;
    while( $i < 10) {
        $i++;
        if( $i == 3 )break;
    }
    echo ("Loop stopped at i = $i" );
?>
</body>
</html>

```

```

<html>
<body>
<?php
    $array = array( 1, 2, 3, 4, 5);
    foreach( $array as $value ) {
        if( $value == 3 )continue;
        echo "Value is $value <br />";
    }
?>
</body>
</html>

```

Nesting Two for Loops

```

1: <?php
2: echo "<table style=\"border: 1px solid #000;\"> \n";
3: for ($y=1; $y<=12; $y++) {
4: echo "<tr> \n";
5: for ($x=1; $x<=12; $x++) {
6: echo "<td style=\"border: 1px solid #000; width: 25px;
7: text-align:center;\">";
8: echo ($x * $y);
9: echo "</td> \n";
10: }
11: echo "</tr> \n";
12: }
13: echo "</table>";
14: ?>

```

A Code Block Containing Multiple echo Statements

```

1: <?php
2: $display_prices = true;
3: if ($display_prices) {

```

```
4: echo "<table border=\"1\">\n";
5: echo "<tr><td colspan=\"3\">";
6: echo "today's prices in dollars";
7: echo "</td></tr>";
8: echo "<tr><td>\$14.00</td><td>\$32.00</td><td>\$71.00</td></tr>\n";
9: echo "</table>";
10: }
11: ?>
```

Returning to HTML Mode Within a Code Block

```
1: <?php
2: $display_prices = true;
3: if ($display_prices) {
4: ?>
5: <table border="1">
6: <tr><td colspan="3">today's prices in dollars</td></tr>
7: <tr><td>$14.00</td><td>$32.00</td><td>$71.00</td></tr>
8: </table>
9: <?php
10: }
11: ?>
```

Calling the Built-In abs() Function

```
1: <?php
2: $num = -321;
3: $newnum = abs($num);
4: echo $newnum;
5: //prints "321"
6: ?>
```

Declaring and Calling a Function

```
1: <?php
2: function bighello()
3: {
4: echo "<h1>HELLO!</h1>";
5: }
6: bighello();
7: ?>
```

Declaring a Function That Requires an Argument

```
1: <?php
2: function printBR($txt)
3: {
4: echo $txt."<br/>";
5: }
6: printBR("This is a line.");
7: printBR("This is a new line.");
8: printBR("This is yet another line.");
9: ?>
```

A Function That Returns a Value

```
1: <?php
```

```
2: function addNums($firstnum, $secondnum)
3: {
4: $result = $firstnum + $secondnum;
5: return $result;
6: }
7: echo addNums(3,5);
8: //will print "8"
9: ?>
```

Variable Scope: A Variable Declared Within a Function Is Unavailable Outside the Function

```
1: <?php
2: function test()
3: {
4: $testvariable = "this is a test variable";
5: }
6: echo "test variable: ".$testvariable."<br/>";
7: ?>
```

Variables Defined Outside Functions Are Inaccessible from Within a Function by Default

```
1: <?php
2: $life = 42;
3: function meaningOfLife()
4: {
5: echo "The meaning of life is ".$life";
6: }
7: meaningOfLife();
8: ?>
```

Accessing Global Variables with the global Statement

```
1: <?php
2: $life=42;
3: function meaningOfLife()
4: {
5: global $life;
6: echo "The meaning of life is ".$life";
7: }
8: meaningOfLife();
9: ?>
```

Using the global Statement to Remember the Value of a Variable Between Function Calls

```
1: <?php
2: $num_of_calls = 0;
3: function numberedHeading($txt)
4: {
5: global $num_of_calls;
6: $num_of_calls++;
7: echo "<h1>".$num_of_calls." ".$txt."</h1>";
8: }
9: numberedHeading("Widgets");
10: echo "<p>We build a fine range of widgets.</p>";
11: numberedHeading("Doodads");
12: echo "<p>Finest in the world.</p>";
13: ?>
```

Using the static Statement to Remember the Value of a Variable Between Function Calls

```
1: <?php
2: function numberedHeading($txt)
3: {
4: static $num_of_calls = 0;
5: $num_of_calls++;
6: echo "<h1>".$num_of_calls." ". $txt."</h1>";
7: }
8: numberedHeading("Widgets");
9: echo "<p>We build a fine range of widgets.</p>";
10: numberedHeading("Doodads");
11: echo "<p>Finest in the world.</p>";
12: ?>
```

A Function Requiring Two Arguments

```
1: <?php
2: function fontWrap($txt, $fontsize)
3: {
4: echo "<span style=\"font-size:$fontsize\">".$txt."</span>";
5: }
6: fontWrap("A Heading<br/>","24pt");
7: fontWrap("some body text<br/>","16pt");
8: fontWrap("smaller body text<br/>","12pt");
9: fontWrap("even smaller body text<br/>","10pt");
10: ?>
```

A Function with an Optional Argument

```
1: <?php
2: function fontWrap($txt, $fontsize = "12pt")
3: {
4: echo "<span style=\"font-size:$fontsize\">".$txt."</span>";
5: }
6: fontWrap("A Heading<br/>","24pt");
7: fontWrap("some body text<br/>");
8: fontWrap("smaller body text<br/>");
9: fontWrap("even smaller body text<br/>");
10: ?>
```

Passing an Argument to a Function by Value

```
1: <?php
2: function addFive($num)
3: {
4: $num += 5;
5: }
6: $orignum = 10;
7: addFive($orignum);
8: echo $orignum;
9: ?>
```

Using a Function Definition to Pass an Argument to a Function by Reference

```
1: <?php
2: function addFive(&$num)
```

```

3: {
4: $num += 5;
5: }
6: $orignum = 10;
7: addFive($orignum);
8: echo $orignum;
9: ?>

```

Testing for a Function's Existence

```

1: <?php
2: function tagWrap($tag, $txt, $func = "")
3: {
4: if ((!empty($txt)) && (function_exists($func))) {
5: $txt = $func($txt);
6: return "<".$tag.">".$txt."</".$tag."><br/>";
7: } else {
8: return "<strong>".$txt."</strong><br/>";
9: }
10: }
11:
12: function underline($txt)
13: {
14: return "<span style=\"text-decoration:underline;\">".$txt."</span>";
15: }
16: echo tagWrap('strong', 'make me bold');
17: echo tagWrap('em', 'underline and italicize me', "underline");
18: echo tagWrap('em', 'make me italic and quote me',
19: create_function('$txt', 'return "&quot;$txt&quot;";'));
20: ?>

```

8. Write code to create and access numeric arrays.

```

<html>
<body>
<?php
/* First method to create array. */
$numbers = array( 1, 2, 3, 4, 5);
foreach( $numbers as $value ) {
    echo "Value is $value <br />";
}
/* Second method to create array. */
$numbers[0] = "one";
$numbers[1] = "two";
$numbers[2] = "three";
$numbers[3] = "four";
$numbers[4] = "five";
foreach( $numbers as $value ) {
    echo "Value is $value <br />";
}
?>

```

9. Demonstrate the usage of associative arrays.

```

<html>
<body>

```

```

<?php
/* First method to associate create array. */
$salaries = array("mohammad" => 2000, "qadir" => 1000, "zara" => 500);
echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
echo "Salary of qadir is ". $salaries['qadir']. "<br />";
echo "Salary of zara is ". $salaries['zara']. "<br />";

/* Second method to create array. */
$salaries['mohammad'] = "high";
$salaries['qadir'] = "medium";
$salaries['zara'] = "low";
echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";
echo "Salary of qadir is ". $salaries['qadir']. "<br />";
echo "Salary of zara is ". $salaries['zara']. "<br />";
?>
</body>
</html>

```

10. Implement Multi-dimensional arrays

```

<html>
<body>
<?php
$marks = array(
    "mohammad" => array (
        "physics" => 35,
        "maths" => 30,
        "chemistry" => 39
    ),

    "qadir" => array (
        "physics" => 30,
        "maths" => 32,
        "chemistry" => 29
    ),

    "zara" => array (
        "physics" => 31,
        "maths" => 22,
        "chemistry" => 39
    )
);

/* Accessing multi-dimensional array values */
echo "Marks for mohammad in physics : ";
echo $marks['mohammad']['physics'] . "<br />";

echo "Marks for qadir in maths : ";
echo $marks['qadir']['maths'] . "<br />";

echo "Marks for zara in chemistry : ";
echo $marks['zara']['chemistry'] . "<br />";
?>
</body>
</html>

```


11. Create a multidimensional array of movies organized by genre. This should take the form of an associative array with genres as keys, such as Science Fiction, Action, Adventure, and so forth. Each of the array's elements should be an array containing movie names, such as Alien, Terminator 3, Star Wars, and so on. After creating your arrays, loop through them, printing the name of each genre and its associated movies.

Rough Draft of the code:

```
<?php
$genres = array('Science Fiction' => array('Star Trek', 'Star Wars', 'Alien'), 'Drama' => array('Les Amant de Pont
Neuf', 'War & Peace', 'Bridehead Revisited'), 'Crime' => array('Heat', 'Pulp Fiction', 'Messerine'));

foreach ($genres as $genre => $movies)
{
    print $genre . " - <br/>";
    foreach ($movies as $movie)
    {
        print $movie . "<br/>";
    }
    print "</br>";
}
?>
```

Try this too:

```
function multiarray_keys($ar) {
    foreach($ar as $k => $v) {
        $keys[] = $k;
        if (is_array($ar[$k]))
            $keys = array_merge($keys, multiarray_keys($ar[$k]));
    }
    return $keys;
}
$gKeys = multiarray_keys($genres);
echo "<pre>";
print_r(multiarray_keys($array));
echo "</pre>";
```

Working with objects

Proof That Your Object Exists

```
<?php
class myClass {
//code will go here
4: }
$object1 = new myClass();
echo "\$object1 is an ".gettype($object1)."<br/>";
if (is_object($object1)) {
echo "Really! I swear \$object1 is an object!";
}
?>
```

Showing Object Properties

```
1: <?php
2: class myCar {
3: public$color = "silver";
```

```
4: public$make = "Mazda";
5: public$model = "Protege5";
6: }
7: $car = new myCar();
8: echo "I drive a: ".$car -> color." ".$car -> make." ".$car -> model;
9: ?>
```

Changing Object Properties

```
1: <?php
2: class myCar {
3: public$color = "silver";
4: public$make = "Mazda";
5: public$model = "Protege5";
6: }
7: $car = new myCar();
8: $car -> color = "red";
9: $car -> make = "Porsche";
10: $car -> model = "Boxter";
11: echo "I drive a: ".$car -> color." ".$car -> make." ".$car -> model;
12: ?>
```

A Class with a Method

```
1: <?php
2: class myClass {
3: function sayHello() {
4: echo "HELLO!";
5: }
6: }
7: $object1 = new myClass();
8: $object1 -> sayHello();
9: ?>
```

Accessing Class Properties Within a Method

```
1: <?php
2: class myClass {
3: public$name = "Jimbo";
4: function sayHello() {
5: echo "HELLO! My name is ".$this->name;
6: }
7: }
8: $object1 = new myClass();
9: $object1 -> sayHello();
10: ?>
```

Changing the Value of a Property from Within a Method

```
1: <?php
2: class myClass {
3: public$name = "Jimbo";
4: function setName($n) {
5: $this->name = $n;
6: }
7: function sayHello() {
8: echo "HELLO! My name is ".$this->name;
```

```

9: }
10: }
11: $object1 = new myClass();
12: $object1 -> setName("Julie");
13: $object1 -> sayHello();
14: ?>

```

A Class Inheriting from Its Parent

```

1: <?php
2: class myClass {
3: public$name = "Matt";
4: function myClass($n) {
5: $this->name = $n;
6: }
7: function sayHello() {
8: echo "HELLO! My name is ".$this->name;
9: }
10: }
11: class childClass extends myClass {
12: //code goes here
13: }
14: $object1 = new childClass("Baby Matt");
15: $object1 -> sayHello();
16: ?>

```

The Method of a Child Class Overriding That of Its Parent

```

1: <?php
2: class myClass {
3: public$name = "Matt";
4: function myClass($n) {
5: $this->name = $n;
6: }
7: function sayHello() {
8: echo "HELLO! My name is ".$this->name;
9: }
10: }
11: class childClass extends myClass {
12: function sayHello() {
13: echo "I will not tell you my name.";
14: }
15: }
16: $object1 = new childClass("Baby Matt");
17: $object1 -> sayHello();
18: ?>

```

12. Create a function that accepts four string variables and returns a string that contains an HTML table element, enclosing each of the variables in its own cell.

```

<?PHP
function addTable($one, $two, $three, $four){
    $data = "<table><tr>\n";
    $data .= "<td>$one</td>\n";
    $data .= "<td>$two</td>\n";
    $data .= "<td>$three</td>\n";
}

```

```

        $data .= "<td>$four</td>\n";
        $data .= "</tr></table>\n";
        return $data;
    }
    echo addTable("one","two","three","four");
?>

```

13. Create a class called baseCalc() that stores two numbers as properties. Next, create a calculate() method that prints the numbers to the browser.

Watch this youtube video and write the solution:

<https://www.youtube.com/watch?v=d6dnCQS8Dck>

14. Create classes called addCalc(), subCalc(), mulCalc(), and divCalc() that inherit functionality from baseCalc() but override the calculate() method and print appropriate totals to the browser.

```

<?php
class MyCalculator {
private $_fval, $_sval;
public function __construct( $fval, $sval ) {
$this->_fval = $fval;
$this->_sval = $sval;
}
public function add() {
return $this->_fval + $this->_sval;
}
public function subtract() {
return $this->_fval - $this->_sval;
}
public function multiply() {
return $this->_fval * $this->_sval;
}
public function divide() {
return $this->_fval / $this->_sval;
}
}
$mycalc = new MyCalculator(12, 6);
echo $mycalc-> add()."\n"; // Displays 18
echo $mycalc-> multiply()."\n"; // Displays 72
echo $mycalc-> subtract()."\n"; // Displays 6
echo $mycalc-> divide()."\n"; // Displays 2
?>

```

Assignments:

1. Sum of Digits
2. Even Odd
3. Prime Number
4. Table of Number
5. Factorial
6. Armstrong Number
7. Palindrome Number
8. Fibonacci Series
9. Reverse Number
10. Reverse String
11. Swapping Two Numbers

12. Adding Two Numbers
13. Subtracting Two Numbers
14. Area of a Triangle
15. Area of Rectangle
16. Leap Year
17. Alphabet Triangle Method
18. Alphabet Triangle
19. Number Triangle
20. Star Triangle

Unit-2: Getting Involved with the Code,
Working with Strings, Dates, and Time - Working with Forms - Working with Cookies and User Sessions -
Working with Files and Directories - Working with Images

Demonstrating Some Type Specifiers

```
1: <?php
2: $number = 543;
3: printf("Decimal: %d<br/>", $number);
4: printf("Binary: %b<br/>", $number);
5: printf("Double: %f<br/>", $number);
6: printf("Octal: %o<br/>", $number);
7: printf("String: %s<br/>", $number);
8: printf("Hex (lower): %x<br/>", $number);
9: printf("Hex (upper): %X<br/>", $number);
10: ?>
```

```
<?php
printf("%04d", 36);
// prints "0036"
?>
```

```
<?php
printf("% 4d", 36)
// prints " 36"
?>
```

```
<?php
printf("%x4d", 36);
// prints "xx36"
?>
```

```
<?php
$red = 1;
$green = 1;
$blue = 1;
printf("#%02X%02X%02X", $red, $green, $blue);
// prints "#010101"
?>
```

```
<?php
echo "<pre>";
printf("%20s\n", "Books");
printf("%20s\n", "CDs");
printf("%20s\n", "DVDs");
printf("%20s\n", "Games");
printf("%20s\n", "Magazines");
echo "</pre>";
?>
```

```
<?php
printf("%.2f", 5.333333);
// prints "5.33"
?>
```

Using printf() to Format a List of Product Prices

```
1: <?php
2: $products = array("Green armchair" => "222.4",
3: "Candlestick"=> "4",
4: "Coffee table"=> "80.6");
5: echo "<pre>";
6: printf("%-20s%20s\n", "Name", "Price");
7: printf("%'-40s\n", "");
8: foreach ($products as $key=>$val) {
9: printf( "%-20s%20.2f\n", $key, $val );
10: }
11: echo "</pre>";
12: ?>
```

local_format.php

```
<?php
return "%02d/%02d/%d<br/>";
?>
```

assume the dates are in a multidimensional array and you are using printf() to format the output:

```
<?php
$dates = array(
array('mon'=> 12, 'mday'=>25, 'year'=>2011),
array('mon'=> 1, 'mday'=>23, 'year'=>2012),
array('mon'=> 10, 'mday'=>29, 'year'=>2011)
);
$format = include("local_format.php");
foreach($dates as $date) {
printf("$format", $date['mon'], $date['mday'], $date['year']);
}
?>
```

Access the individual characters of a string as if they were elements of an array:

```
<?php
$test = "phpcoder";
echo $test[0]; // prints "p"
echo $test[4]; // prints "o"
?>
```

Finding the Length of a String with strlen()

```
<?php
$membership = "pAB7";
if (strlen($membership) == 4) {
echo "<p>Thank you!</p>";
} else {
echo "<p>Your membership number must be four characters long.</p>";
}
?>
```

Finding a Substring Within a String with strstr()

```
<?php
$membership = "pAB7";
if (strstr($membership, "AB")) {
```

```

echo "<p>Your membership expires soon!</p>";
} else {
echo "<p>Thank you!</p>";
}
?>

```

Finding the Position of a Substring with strpos()

```

<?php
$membership = "mz00xyz";
if (strpos($membership, "mz") === 0) {
echo "Hello mz!";
}
?>

```

Extracting Part of a String with substr()

```

<?php
$test = "phpcoder";
echo substr($test,3)."<br/>"; // prints "coder"
echo substr($test,3,2)."<br/>"; // prints "co"
?>

```

```

<?php
$test = "pierre@wanadoo.fr";
if ($test = substr($test, -3) == ".fr") {
echo "<p>Bonjour! We have special prizes for you.</p>";
} else {
echo "<p>Welcome to our store.</p>";
}
?>

```

Tokenizing a String with strtok()

```

1: <?php
2: $test = "http://www.google.com/search?";
3: $test .= "hl=en&ie=UTF-8&q=php+development+books&btnG=Google+Search";
4: $delims = "?&";
5: $word = strtok($test, $delims);
6: while (is_string($word)) {
7: if ($word) {
8: echo $word."<br/>";
9: }
10: $word = strtok($delims);
11: }
12: ?>

```

Cleaning Up a String with trim(), ltrim(), and strip_tags()

```

<?php
$text = "\t\tlots of room to breathe ";
echo "<pre>$text</pre>";
// prints " lots of room to breathe ";
$text = trim($text);
echo "<pre>$text</pre>";
// prints "lots of room to breathe";
?>

```



```

<?php
$text = "\t\tlots of room to breathe ";
echo "<pre>$text</pre>";
// prints "      lots of room to breathe ";
$text = rtrim($text);
echo "<pre>$text</pre>";
// prints "      lots of room to breathe";
?>

```

```

<?php
$text = "\t\tlots of room to breathe ";
echo "<pre>$text</pre>";
// prints " lots of room to breathe ";
$text = ltrim($text);
echo "<pre>$text</pre>";
// prints "lots of room to breathe ";
?>

```

```

<?php
$string = "<p>\"I <em>simply</em> will not have it,\" <br/>said Mr Dean.</p>";
echo strip_tags($string, "<br/><p>");
?>

```

Replacing a Portion of a String Using substr_replace()

```

<?php
$membership = "mz11xyz";
$membership = substr_replace($membership, "12", 2, 2);
echo "New membership number: $membership";
// prints "New membership number: mz12xyz"
?>

```

Replacing Substrings Using str_replace

```

<?php
$string = "<h1>The 2010 Guide to All Things Good in the World</h1>";
$string .= "<p>Site contents copyright 2010.</p>";
echo str_replace("2010","2012",$string);
?>

```

```

<?php
$source = array(
    "The package which is at version 4.2 was released in 2005.",
    "The year 2005 was an excellent time for PointyThing 4.2!");
$search = array("4.2", "2005");
$replace = array("6.3", "2012");
$source = str_replace($search, $replace, $source);
foreach($source as $str) {
    echo "$str<br>";
}
?>

```

Converting Case

```
<?php
$membership = "mz11xyz";
$membership = strtoupper($membership);
echo "$membership"; // prints "MZ11XYZ"
?>
```

```
<?php
$membership = "MZ11XYZ";
$membership = strtolower($membership);
echo "$membership"; // prints "mz11xyz"
?>
```

```
<?php
$full_name = "violet elizabeth bott";
$full_name = ucwords($full_name);
echo $full_name; // prints "Violet Elizabeth Bott"
?>
```

```
<?php
$full_name = "VIoLEt eLIZaBeTH bOTt";
$full_name = ucwords(strtolower($full_name));
echo $full_name; // prints "Violet Elizabeth Bott"
?>
```

```
<?php
$myString = "this is my string.";
$myString = ucfirst($myString);
echo $myString; // prints "This is my string."
?>
```

Wrapping Text with wordwrap() and nl2br()

```
<?php
$string = "one line\n";
$string .= "another line\n";
$string .= "a third for luck\n";
echo nl2br($string);
?>
```

```
<?php
$string = "Given a long line, wordwrap() is useful as a means of ";
$string .= "breaking it into a column and thereby making it easier to read";
echo wordwrap($string);
?>
```

```
<?php
$string = "As usual you will find me at http://www.witteringonaboutit.com/";
$string .= "chat/eating_green_cheese/forum.php. Hope to see you there!";
echo wordwrap($string, 24, "<br/>\n", 1);
?>
```

Breaking Strings into Arrays with explode()

```
<?php
$start_date = "2012-02-19";
```

```
$date_array = explode("-", $start_date);  
// $date_array[0] == "2012"  
// $date_array[1] == "02"  
// $date_array[2] == "19"  
echo $date_array[0].".".$date_array[1].".".$date_array[2];  
//prints 2012-02-19  
?>
```

Acquiring Date Information with getdate()

```
1: <?php  
2: $date_array = getdate(); // no argument passed so today's date will be used  
3: foreach ($date_array as $key => $val) {  
4: echo "$key = $val<br>";  
5: }  
6: ?>  
7: <hr/>  
8: <?php  
9: echo "<p>Today's date: ".$date_array['mon']."/".$date_array['mday']."/".  
10: $date_array['year']. "</p>";  
11: ?>
```

Formatting a Date with date()

```
1: <?php  
2: $time = time(); //stores the exact timestamp to use in this script  
3: echo date("m/d/y G:i:s e", $time);  
4: echo "<br/>";  
5: echo "Today is ";  
6: echo date("jS \of F Y, \a\\t g:ia \i\\n e", $time);  
7: ?>
```

Creating Timestamps with mktime()

```
1: <?php  
2: // make a timestamp for Jan 17 2012 at 9:34 pm  
3: $ts = mktime(21, 34, 0, 1, 17, 2012);  
4: echo date("m/d/y G:i:s e", $ts);  
5: echo "<br/>";  
6: echo "The date is ";  
7: echo date("jS \of F Y, \a\\t g:ia \i\\n e", $ts );  
8: ?>
```

Creating a Simple Input Form

```
1: <!DOCTYPE html>  
2: <html>  
3: <head>  
4: <title>A simple HTML form</title>  
5: </head>  
6: <body>  
7: <form method="post" action="send_simpleform.php">  
8: <p><label for="user">Name:</label><br/>  
9: <input type="text" id="user" name="user"></p>  
10: <p><label for="message">Message:</label><br/>  
11: <textarea id="message" name="message" rows="5" cols="40"></textarea></p>  
12: <button type="submit" name="submit" value="send">Send Message</button>
```

```
13: </form>
14: </body>
15: </html>
```

Reading Input from a Form

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>A simple response</title>
5: </head>
6: <body>
7: <p>Welcome, <strong><?php echo $_POST['user']; ?></strong>!</p>
8: <p>Your message is:
9: <strong><?php echo $_POST['message']; ?></strong></p>
10: </body>
11: </html>
```

Accessing Form Input with User-Defined Arrays

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>An HTML form with checkboxes</title>
5: </head>
6: <body>
7: <form action="send_formwithcb.php" method="POST">
8: <p><label>Name:</label><br />
9: <input type="text" name="user" /></p>
10: <fieldset>
11: <legend>Select Some Products:</legend><br />
12: <input type="checkbox" id="tricorder"
13: name="products[]" value="Tricorder">
14: <label for="tricorder">Tricorder</label><br />
15:
16: <input type="checkbox" id="ORAC_AI"
17: name="products[]" value="ORAC AI">
18: <label for="ORAC_AI">ORAC AI</label><br />
19:
20: <input type="checkbox" id="HAL_2000"
21: name="products[]" value="HAL 2000">
22: <label for="HAL_2000">HAL 2000</label>
23: </fieldset>
24: <button type="submit" name="submit" value="submit">Submit Form</button>
25: </form>
26: </body>
27: </html>
```

Reading Input from the Form

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>Reading checkboxes</title>
5: </head>
6: <body>
```

```

7: <p>Welcome, <strong><?php echo $_POST['user']; ?></strong>!</p>
8: <p>Your product choices are:
9: <?php
10: if (!empty($_POST['products'])) {
11: echo "<ul>";
12: foreach ($_POST['products'] as $value) {
13: echo "<li>$value</li>";
14: }
15: echo "</ul>";
16: } else {
17: echo "None";
18: }
19: ?>
20: </body>
21: </html>

```

Combining HTML and PHP Code on a Single Page

```

1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>An HTML form that calls itself</title>
5: </head>
6: <body>
7: <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
8: <p><label for="guess">Type your guess here:</label> <br/>
9: <input type="text" id="guess" name="guess" /></p>
10: <button type="submit" name="submit" value="submit">Submit</button>
11: </form>
12: </body>
13: </html>

```

```

1: <?php
2: $num_to_guess = 42;
3: if (!isset($_POST['guess'])) {
4: $message = "Welcome to the guessing machine!";
5: } elseif (!is_numeric($_POST['guess'])) { // is not numeric
6: $message = "I don't understand that response.";
7: } elseif ($_POST['guess'] == $num_to_guess) { // matches!
8: $message = "Well done!";
9: } elseif ($_POST['guess'] > $num_to_guess) {
10: $message = $_POST['guess']. " is too big! Try a smaller number.";
11: } elseif ($_POST['guess'] < $num_to_guess) {
12: $message = $_POST['guess']. " is too small! Try a larger number.";
13: } else { // some other condition
14: $message = "I am terribly confused.";
15: }
16: ?>
17: <!DOCTYPE html>
18: <html>
19: <head>
20: <title>A PHP number guessing script</title>
21: </head>
22: <body>

```

```

23: <h1><?php echo $message; ?></h1>
24: <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
25: <p><label for="guess">Type your guess here:</label><br/>
26: <input type="text" is="guess" name="guess" /></p>
27: <button type="submit" name="submit" value="submit">Submit</button>
28: </form>
29: </body>
30: </html>

```

Saving State with a Hidden Field

```

1: <?php
2: $num_to_guess = 42;
3: $num_tries = (isset($_POST['num_tries'])) ? $num_tries + 1 : 1;
4: if (!isset($_POST['guess'])) {
5: $message = "Welcome to the guessing machine!";
6: } elseif (!is_numeric($_POST['guess'])) { // is not numeric
7: $message = "I don't understand that response.";
8: } elseif ($_POST['guess'] == $num_to_guess) { // matches!
9: $message = "Well done!";
10: } elseif ($_POST['guess'] > $num_to_guess) {
11: $message = $_POST['guess'] . " is too big! Try a smaller number.";
12: } elseif ($_POST['guess'] < $num_to_guess) {
13: $message = $_POST['guess'] . " is too small! Try a larger number.";
14: } else { // some other condition
15: $message = "I am terribly confused.";
16: }
17: ?>
18: <!DOCTYPE html>
19: <html>
20: <head>
21: <title>A PHP number guessing script</title>
22: </head>
23: <body>
24: <h1><?php echo $message; ?></h1>
25: <p><strong>Guess number:</strong> <?php echo $num_tries; ?></p>
26: <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
27: <p><label for="guess">Type your guess here:</label><br/>
28: <input type="text" id="guess" name="guess" /></p>
29: <input type="hidden" name="num_tries" value="<?php echo $num_tries; ?>" />
30: <button type="submit" name="submit" value="submit">Submit</button>
31: </form>
32: </body>
33: </html>

```

Using header() to Redirect User

```

1: <?php
2: $num_to_guess = 42;
3: $num_tries = (isset($_POST['num_tries'])) ? $num_tries + 1 : 1;
4: if (!isset($_POST['guess'])) {
5: $message = "Welcome to the guessing machine!";
6: } elseif (!is_numeric($_POST['guess'])) { // is not numeric
7: $message = "I don't understand that response.";
8: } elseif ($_POST['guess'] == $num_to_guess) { // matches!

```

```

9: header("Location: congrats.html");
10: exit;
11: } elseif ($_POST['guess'] > $num_to_guess) {
12: $message = $_POST['guess']. " is too big! Try a smaller number.";
13: } elseif ($_POST['guess'] < $num_to_guess) {
14: $message = $_POST['guess']. " is too small! Try a larger number.";
15: } else { // some other condition
16: $message = "I am terribly confused.";
17: }
18: ?>
19:
20: <!DOCTYPE html>
21: <html>
22: <head>
23: <title>A PHP number guessing script</title>
24: </head>
25: <body>
26: <h1><?php echo $message; ?></h1>
27: <p><strong>Guess number:</strong> <?php echo $num_tries; ?></p>
28: <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
29: <p><label for="guess">Type your guess here:</label><br/>
30: <input type="text" id="guess" name="guess" /></p>
31: <input type="hidden" name="num_tries" value="<?php echo $num_tries; ?>" />
32: <button type="submit" name="submit" value="submit">Submit</button>
33: </form>
34: </body>
35: </html>

```

Creating a Simple Feedback Form

```

1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>E-Mail Form</title>
5: </head>
6: <body>
7: <form action="sendmail.php" method="POST">
8: <p><label for="name">Name:</label><br/>
9: <input type="text" size="25" id="name" name="name" /></p>
10: <p><label for="email">E-Mail Address:</label><br/>
11: <input type="text" size="25" id="email" name="email" /></p>
12: <p><label for="msg">Message:</label><br/>
13: <textarea id="msg" name="msg" cols="30" rows="5"></textarea></p>
14: <button type="submit" name="submit" value="send">Send Message</button>
15: </form>
16: </body>
17: </html>

```

Creating the Script to Send the Mail

```

1: <?php
2: //start building the mail string
3: $msg = "Name: ".$_POST['name']."\n";
4: $msg .= "E-Mail: ".$_POST['email']."\n";
5: $msg .= "Message: ".$_POST['message']."\n";

```

```

6:
7: //set up the mail
8: $recipient = "you@yourdomain.com";
9: $subject = "Form Submission Results";
10: $mailheaders = "From: My Web Site <defaultaddress@yourdomain.com> \n";
11: $mailheaders .= "Reply-To: ".$_POST['email'];
12:
13: //send the mail
14: mail($recipient, $subject, $msg, $mailheaders);
15: ?>
16: <!DOCTYPE html>
17: <html>
18: <head>
19: <title>Sending mail from the form in Listing 11.10</title>
20: </head>
21: <body>
22: <p>Thanks, <strong><?php echo $_POST['name']; ?></strong>,
23: for your message.</p>
24: <p>Your e-mail address:
25: <strong><?php echo $_POST['email']; ?></strong></p>
26: <p>Your message: <br/> <?php echo $_POST['message']; ?> </p>
27: </body>
28: </html>

```

Sending the Simple Feedback Form: HTML Version

```

1: <?php
2: //start building the mail string
3: $msg = "<p><strong>Name:</strong> ".$_POST['name']."</p>";
4: $msg .= "<p><strong>E-Mail:</strong> ".$_POST['email']."</p>";
5: $msg .= "<p><strong>Message:</strong> ".$_POST['message']."</p>";
6:
7: //set up the mail
8: $recipient = "you@yourdomain.com";
9: $subject = "Form Submission Results";
10: $mailheaders = "MIME-Version: 1.0\r\n";
11: $mailheaders .= "Content-type: text/html; charset=ISO-8859-1\r\n";
12: $mailheaders = "From: My Web Site <defaultaddress@yourdomain.com> \n";
13: $mailheaders .= "Reply-To: ".$_POST['email'];
14:
15: //send the mail
16: mail($recipient, $subject, $msg, $mailheaders);
17: ?>
18: <!DOCTYPE html>
19: <html>
20: <head>
21: <title>Sending the Simple Feedback Form - HTML Version</title>
22: </head>
23: <body>
24: <p>Thanks, <strong><?php echo $_POST['name']; ?></strong>,
25: for your message.</p>
26: <p>Your e-mail address:
27: <strong><?php echo $_POST['email']; ?></strong></p>
28: <p>Your message: <br/> <?php echo $_POST['message']; ?> </p>

```



```
29: </body>
30: </html>
```

Working with File Uploads

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>A simple file upload form</title>
5: </head>
6: <body>
7: <form action="do_upload.php" enctype="multipart/form-data" method="POST">
8: <input type="hidden" name="MAX_FILE_SIZE" value="1048576" />
9: <p><label for="fileupload">File to Upload:</label>
10: <input type="file" id="fileupload" name="fileupload" /></p>
11: <button type="submit" name="submit" value="send">Upload File</button>
12: </form>
13: </body>
14: </html>
```

A File Upload Script1: <?php

```
2: $file_dir = "/path/to/upload/directory";
3:
4: foreach($ _FILES as $file_name => $file_array) {
5: echo "path: ".$file_array['tmp_name']."<br/>\n";
6: echo "name: ".$file_array['name']."<br/>\n";
7: echo "type: ".$file_array['type']."<br/>\n";
8: echo "size: ".$file_array['size']."<br/>\n";
9:
10: if (is_uploaded_file($file_array['tmp_name'])) {
11: move_uploaded_file($file_array['tmp_name'],
12: "$file_dir/".$file_array['name'])
13: or die ("Couldn't move file");
14: echo "File was moved!";
15: } else {
16: echo "No file found.";
17: }
18: }
19: ?>
```

Working with Cookies and User Sessions

Setting and Printing a Cookie Value

```
1: <?php
2: setcookie("vegetable", "artichoke", time()+3600, "/", ".yourdomain.com", 0);
3:
4: if (isset($_COOKIE['vegetable'])) {
5: echo "<p>Hello again! You have chosen: ".$_COOKIE['vegetable']."</p>";
6: } else {
7: echo "<p>Hello, you. This may be your first visit.</p>";
8: }
9: ?>
```

Starting or Resuming a Session

```
1: <?php
```

```
2: session_start();
3: echo "<p>Your session ID is ".session_id()."</p>";
4: ?>
```

Storing Variables in a Session

```
1: <?php
2: session_start();
3: $_SESSION['product1'] = "Sonic Screwdriver";
4: $_SESSION['product2'] = "HAL 2000";
5: echo "The products have been registered.";
6: ?>
```

Accessing Stored Session Variables

```
1: <?php
2: session_start();
3: ?>
4: <p>Your chosen products are:</p>
5: <ul>
6: <li><?php echo $_SESSION['product1']; ?></li>
7: <li><?php echo $_SESSION['product2']; ?></li>
8: </ul>
```

Adding an Array Variable to a Session Variable

```
1: <?php
2: session_start();
3: ?>
4: <!DOCTYPE html>
5: <html>
6: <head>
7: <title>Storing an array with a session</title>
8: </head>
9: <body>
10: <h1>Product Choice Page</h1>
11: <?php
12: if (isset($_POST['form_products'])) {
13: if (!empty($_SESSION['products'])) {
14: $products = array_unique(
15: array_merge(unserialize($_SESSION['products']),
16: $_POST['form_products']));
17: $_SESSION['products'] = serialize($products);
18: } else {
19: $_SESSION['products'] = serialize($_POST['form_products']);
20: }
21: echo "<p>Your products have been registered!</p>";
22: }
23: ?>
24: <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
25: <p><label for="form_products">Select some products:</label><br />
26: <select id="form_products" name="form_products[]" multiple="multiple"
size="3">
27: <option value="Sonic Screwdriver">Sonic Screwdriver</option>
28: <option value="Hal 2000">Hal 2000</option>
29: <option value="Tardis">Tardis</option>
```

```

30: <option value="ORAC">ORAC</option>
31: <option value="Transporter bracelet">Transporter bracelet</option>
32: </select></p>
33: <button type="submit" name="submit" value="choose">Submit Form</button>
34: </form>
35: <p><a href="session1.php">go to content page</a></p>
36: </body>
37: </html>

```

Accessing Session Variables

```

1: <?php
2: session_start();
3: ?>
4: <!DOCTYPE html>
5: <html>
6: <head>
7: <title>Accessing session variables</title>
8: </head>
9: <body>
10: <h1>Content Page</h1>
11: <?php
12: if (isset($_SESSION['products'])) {
13: echo "<strong>Your cart:</strong><ol>";
14: foreach (unserialize($_SESSION['products']) as $p) {
15: echo "<li>".$p."</li>";
16: }
17: echo "</ol>";
18: }
19: ?>
20: <p><a href="arraysession.php">return to product choice page</a></p>
21: </body>
22: </html>

```

Working with Files and Directories

An Include File Containing PHP Code - myinclude.php

```

1: <?php
2: echo "I have been included!!<br/>";
3: echo "But now I can add up... 4 + 4 = ".(4 + 4);
4: ?>

```

Using include()

```

1: <?php
2: include 'myinclude.php';
3: ?>

```

An Include File That Returns a Value

```

1: <?php
2: $retval = ( 4 + 4 );
3: return $retval;
4: ?>

```

Using include to Execute PHP and Assign the Return Value

```

1: <?php

```

```

2: $addResult = include 'returnvalue.php';
3: echo "The include file returned ".$addResult;
4: ?>

```

Using include Within a Loop

```

1: <?php
2: for ($x = 1; $x<=3; $x++) {
3: $incfile = "incfile".$x.".txt";
4: echo "Attempting to include ".$incfile."<br/>";
5: include $incfile;
6: echo "<hr/>";
7: }
8: ?>

```

Save the contents in a file named **loopy_include.php** and place it in the document root of the web server, along with three different files: incfile1.txt, incfile2.txt, and incfile3.txt.

A Function to Output the Results of Multiple File Tests

```

1: <?php
2: function outputFileTestInfo($f) {
3: if (!file_exists($f)) {
4: echo "<p>$f does not exist</p>";
5: return;
6: }
7: echo "<p>$f is ".(is_file($f) ? "" : "not ")."a file</p>";
8: echo "<p>$f is ".(is_dir($f) ? "" : "not ")."a directory</p>";
9: echo "<p>$f is ".(is_readable($f) ? "" : "not ")."readable</p>";
10: echo "<p>$f is ".(is_writable($f) ? "" : "not ")."writable</p>";
11: echo "<p>$f is ".(is_executable($f) ? "" : "not ")."executable</p>";
12: echo "<p>$f is ".(filesize($f))." bytes</p>";
13: echo "<p>$f was accessed on ".date("D d M Y g:i A",fileatime($f))."</p>";
14: echo "<p>$f was modified on ".date("D d M Y g:i A",filemtime($f))."</p>";
15: echo "<p>$f was changed on ".date("D d M Y g:i A",filectime($f))."</p>";
16: }
17: $file = "test.txt";
18: outputFileTestInfo($file);
19: ?>

```

Opening and Reading a File Line by Line

```

1: <?php
2: $filename = "test.txt";
3: $fp = fopen($filename, "r") or die("Couldn't open $filename");
4: while (!feof($fp)) {
5: $line = fgets($fp, 1024);
6: echo $line."<br/>";
7: }
8: ?>

```

Reading a File with fread()

```

1: <?php
2: $filename = "test.txt";
3: $fp = fopen($filename, "r") or die("Couldn't open $filename");
4: while (!feof($fp)) {
5: $chunk = fread($fp, 8);

```

```
6: echo $chunk."<br/>";
7: }
8: ?>
```

Moving Around a File with fseek()

```
1: <?php
2: $filename = "test.txt";
3: $fp = fopen($filename, "r") or die("Couldn't open $filename");
4: $fsize = filesize($filename);
5: $halfway = (int)($fsize / 2);
6: echo "Halfway point: ".$halfway." <br/>\n";
7: fseek($fp, $halfway);
8: $chunk = fread($fp, ($fsize - $halfway));
9: echo $chunk;
10: ?>
```

Moving Around a File with fgetc()

```
1: <?php
2: $filename = "test.txt";
3: $fp = fopen($filename, "r") or die("Couldn't open $filename");
4: while (!feof($fp)) {
5: $char = fgetc($fp);
6: echo $char."<br/>";
7: }
8: ?>
```

Writing and Appending to a File

```
1: <?php
2: $filename = "test.txt";
3: echo "<p>Writing to ".$filename." ... </p>";
4: $fp = fopen($filename, "w") or die("Couldn't open $filename");
5: fwrite($fp, "Hello world\n");
6: fclose($fp);
7: echo "<p>Appending to ".$filename." ...</p>";
8: $fp = fopen($filename, "a") or die("Couldn't open $filename");
9: fputs($fp, "And another thing\n");
10: fclose($fp);
11: ?>
```

Writing and Appending to a File

```
1: <?php
2: $filename = "test.txt";
3: echo "<p>Writing to ".$filename." ... </p>";
4: file_put_contents ($filename, "Hello world\n");
5: echo "<p>Appending to ".$filename." ...</p>";
6: file_put_contents ($filename, "And another thing\n", FILE_APPEND);
7: ?>
```

Listing the Contents of a Directory with readdir()

```
1: <?php
2: $dirname = ".";
3: $dh = opendir($dirname) or die("Couldn't open directory");
4:
```

```

5: while (!(($file = readdir($dh)) === false ) ) {
6: if (is_dir("$dirname/$file")) {
7: echo "(D) ";
8: }
9: echo $file."<br/>";
10: }
11: closedir($dh);
12: ?>

```

Using popen() to Read a File

```

1: <?php
2: $file_handle = popen("/path/to/fakefile 2>&1", "r");
3: $read = fread($file_handle, 2096);
4: echo $read;
5: pclose($file_handle);
6: ?>

```

Working with Images

Creating a New Image

```

1: <?php
2: //create the canvas
3: $myImage = ImageCreate(300,300);
4:
5: //set up some colors for use on the canvas
6: $black = ImageColorAllocate($myImage, 0, 0, 0);
7: $white = ImageColorAllocate($myImage, 255, 255, 255);
8: $red = ImageColorAllocate($myImage, 255, 0, 0);
9: $green = ImageColorAllocate($myImage, 0, 255, 0);
10: $blue = ImageColorAllocate($myImage, 0, 0, 255);
11:
12: //draw some rectangles
13: ImageRectangle($myImage, 15, 15, 95, 155, $red);
14: ImageRectangle($myImage, 95, 155, 175, 295, $white);
15: ImageRectangle($myImage, 175, 15, 255, 155, $red);
16:
17: //output the image to the browser
18: header ("Content-type: image/png");
19: ImagePng($myImage);
20:
21: //clean up after yourself
22: ImageDestroy($myImage);
23: ?>

```

Creating a New Image with Color Fills

```

1: <?php
2: //create the canvas
3: $myImage = ImageCreate(300,300);
4:
5: //set up some colors for use on the canvas
6: $black = ImageColorAllocate($myImage, 0, 0, 0);
7: $white = ImageColorAllocate($myImage, 255, 255, 255);
8: $red = ImageColorAllocate($myImage, 255, 0, 0);
9: $green = ImageColorAllocate($myImage, 0, 255, 0);

```

```
10: $blue = ImageColorAllocate($myImage, 0, 0, 255);
11:
12: //draw some rectangles
13: ImageFilledRectangle($myImage, 15, 15, 95, 155, $red);
14: ImageFilledRectangle($myImage, 95, 155, 175, 295, $white);
15: ImageFilledRectangle($myImage, 175, 15, 255, 155, $red);
16:
17: //output the image to the browser
18: header ("Content-type: image/png");
19: ImagePng($myImage);
20:
21: //clean up after yourself
22: ImageDestroy($myImage);
23: ?>
```

A Basic Pie Chart

```
1: <?php
2: //create the canvas
3: $myImage = ImageCreate(300,300);
4:
5: //set up some colors for use on the canvas
6: $white = ImageColorAllocate($myImage, 255, 255, 255);
7: $red = ImageColorAllocate($myImage, 255, 0, 0);
8: $green = ImageColorAllocate($myImage, 0, 255, 0);
9: $blue = ImageColorAllocate($myImage, 0, 0, 255);
10:
11: //draw a pie
12: ImageFilledArc($myImage, 100, 100, 200, 150, 0, 90, $red, IMG_ARC_PIE);
13: ImageFilledArc($myImage, 100, 100, 200, 150, 90, 180, $green,
IMG_ARC_PIE);
14: ImageFilledArc($myImage, 100, 100, 200, 150, 180, 360, $blue,
IMG_ARC_PIE);
15:
16: //output the image to the browser
17: header ("Content-type: image/png");
18: ImagePng($myImage);
19:
20: //clean up after yourself
21: ImageDestroy($myImage);
22: ?>
```

A 3D Pie Chart

```
1: <?php
2: //create the canvas
3: $myImage = ImageCreate(300,300);
4:
5: //set up some colors for use on the canvas
6: $white = ImageColorAllocate($myImage, 255, 255, 255);
7: $red = ImageColorAllocate($myImage, 255, 0, 0);
8: $green = ImageColorAllocate($myImage, 0, 255, 0);
9: $blue = ImageColorAllocate($myImage, 0, 0, 255);
10: $lt_red = ImageColorAllocate($myImage, 255, 150, 150);
11: $lt_green = ImageColorAllocate($myImage, 150, 255, 150);
```

```

12: $lt_blue = ImageColorAllocate($myImage, 150, 150, 255);
13:
14: //draw the shaded area
15: for ($i = 110;$i > 100;$i--) {
16: ImageFilledArc ($myImage,100,$i,200,150,0,90,$lt_red,IMG_ARC_PIE);
17: ImageFilledArc ($myImage,100,$i,200,150,90,180,$lt_green,IMG_ARC_PIE);
18: ImageFilledArc ($myImage,100,$i,200,150,180,360,$lt_blue,IMG_ARC_PIE);
19: }
20:
21: //draw a pie
22: ImageFilledArc($myImage, 100, 100, 200, 150, 0, 90, $red, IMG_ARC_PIE);
23: ImageFilledArc($myImage, 100, 100, 200, 150, 90, 180, $green, IMG_ARC_PIE);
24: ImageFilledArc($myImage, 100, 100, 200, 150, 180, 360, $blue, IMG_ARC_PIE);
25:
26: //output the image to the browser
27: header ("Content-type: image/png");
28: ImagePng($myImage);
29:
30: //clean up after yourself
31: ImageDestroy($myImage);
32: ?>

```

Creating a New Image from an Existing Image

```

1: <?php
2: //use existing image as a canvas
3: $myImage = ImageCreateFromPng("baseimage.png");
4:
5: //allocate the color white
6: $white = ImageColorAllocate($myImage, 255, 255, 255);
7:
8: //draw on the new canvas
9: ImageFilledEllipse($myImage, 100, 70, 20, 20, $white);
10: ImageFilledEllipse($myImage, 175, 70, 20, 20, $white);
11: ImageFilledEllipse($myImage, 250, 70, 20, 20, $white);
12:
13: //output the image to the browser
14: header ("Content-type: image/png");
15: ImagePng($myImage);
16:
17: //clean up after yourself
18: ImageDestroy($myImage);
19: ?>

```

Stacking Images and Making Them Transparent

```

1: <?php
2: //select an image to start with
3: $baseimage = ImageCreateFromPng("img1.png");
4:
5: //loop through images #2 through the end
6: for($i=2; $i <5; $i++) {
7: //allocate the transparent color, and stack
8: $myImage = ImageCreateFromPng("img".$i.".png");
9: $gray = ImageColorAllocate($myImage, 185, 185, 185);

```



```

10: ImageColorTransparent($myImage, $gray);
11: ImageCopyMerge($baseimage,$myImage,0,0,0,0,150,150,100);
12: }
13:
14: //output the image to the browser
15: header ("Content-type: image/png");
16: ImagePng($baseimage);
17:
18: //clean up after yourself
19: ImageDestroy($baseimage);
20: ?>

```

Creating an Image from User Input

```

1: <?php
2: if (!$_POST) {
3: //show form
4: ?>
5: <!DOCTYPE html>
6: <html>
7: <head>
8: <title>Image Creation Form</title>
9:
10: <style type="text/css">
11: fieldset{border: 0; padding: 0px 0px 12px 0px;}
12: fieldset label {margin-left: 24px;}
13: legend, label {font-weight:bold;}
14: </style>
15:
16: </head>
17: <body>
18: <h1>Create an Image</h1>
19: <form method="POST" action="<?php echo $_SERVER['PHP_SELF']; ?>">
20:
21: <fieldset>
22: <legend>Image Size:</legend><br/>
23: <label for="w">W:</label>
24: <input type="text" id="w" name="w" size="5" maxlength="5" />
25: <label for="h">H:</label>
26: <input type="text" id="h" name="h" size="5" maxlength="5" />
27: </fieldset>
28:
29: <fieldset>
30: <legend>Background Color:</legend><br/>
31: <label for="b_r">R:</label>
32: <input type="text" id="b_r" name="b_r" size="3" maxlength="3" />
33: <label for="b_g">G:</label>
34: <input type="text" id="b_g" name="b_g" size="3" maxlength="3" />
35: <label for="b_b">B:</label>
36: <input type="text" id="b_b" name="b_b" size="3" maxlength="3" />
37: </fieldset>
38:
39: <fieldset>
40: <legend>Text Color:</legend><br/>

```

```

41: <label for="t_r">R:</label>
42: <input type="text" id="t_r" name="t_r" size="3" maxlength="3" />
43: <label for="t_g">G:</label>
44: <input type="text" id="t_g" name="t_g" size="3" maxlength="3" />
45: <label for="t_b">B:</label>
46: <input type="text" id="t_b" name="t_b" size="3" maxlength="3" />
47: </fieldset>
48:
49: <p><label for="string">Text String:</label>
50: <input type="text" id="string" name="string" size="35" /></p>
51:
52: <p><label for="font_size">Font Size:</label>
53: <select id="font_size" name="font_size">
54: <option value="1">1</option>
55: <option value="2">2</option>
56: <option value="3">3</option>
57: <option value="4">4</option>
58: <option value="5">5</option>
59: </select></p>
60:
61: <fieldset>
62: <legend>Text Starting Position:</legend><br/>
63: <label for="x">X:</label>
64: <input type="text" id="x" name="x" size="3" maxlength="3" />
65: <label for="y">Y:</label>
66: <input type="text" id="y" name="y" size="3" maxlength="3" />
67: </fieldset>
68:
69: <button type="submit" name="submit" value="create">Create Image</button>
70: </form>
71: </body>
72: </html>
73: <?php
74: } else {
75: //create image
76: //create the canvas
77: $myImage = ImageCreate($_POST['w'], $_POST['h']);
78:
79: //set up some colors
80: $background = ImageColorAllocate ($myImage, $_POST['b_r'],
81: $_POST['b_g'], $_POST['b_b']);
82: $text = ImageColorAllocate ($myImage, $_POST['t_r'],
83: $_POST['t_g'], $_POST['t_b']);
84:
85: // write the string at the top left
86: ImageString($myImage, $_POST['font_size'], $_POST['x'],
87: $_POST['y'], $_POST['string'], $text);
88:
89: //output the image to the browser
90: header ("Content-type: image/png");
91: ImagePNG($myImage);
92:
93: //clean up after yourself

```

```
94: ImageDestroy($myImage);
95: }
96: ?>
```

Creating an Image with Custom Font and Text

```
1: <?php
2: //create the canvas
3: $myImage = ImageCreate(150,25);
4:
5: //set up some colors for use on the canvas
6: $white = ImageColorAllocate($myImage, 255, 255, 255);
7: $black = ImageColorAllocate($myImage, 0, 0, 0);
8:
9: //load a font
10: $font = imageloadfont("hootie.gdf");
11:
12: // write the string
13: ImageString($myImage, $font, 0, 0, "CAPTCHA!", $black);
14:
15: //output the image to the browser
16: header ("Content-type: image/png");
17: ImagePng($myImage);
18:
19: //clean up after yourself
20: ImageDestroy($myImage);
21: ?>
```

Using Images Created by Scripts

```
1: <!DOCTYPE html>
2: <html>
3: <head>
4: <title>Using Images Created by Scripts</title>
5: </head>
6: </body>
7: <h1>Generated Image Below...</h1>
8: 
9: </body>
10: </html>
```

Assignments:

1. Create a feedback form that accepts a user's full name and an email address. Use case-conversion functions to capitalize the first letter of each name the user submits and print the result back to the browser. Check that the user's email address contains the @ symbol and print a warning otherwise.

Suggestions:

```
<label for="name">Name</label>
```

```
<div>
```

```
  <input type="text" name="name" id="name" value="">
```

```
</div>
```

```
<label for="email">Email</label>
```

```
<div>
```

```
  <input type="text" name="email" id="email" value="">
```

```
</div>
```

```
<?php
```

```
$email = "john.doe@example.com";
```

```
// Remove all illegal characters from email
```

```
$email = filter_var($email, FILTER_SANITIZE_EMAIL);
```

```
// Validate email
```

```
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
```

```
  echo $email." is a valid email address";
```

```
} else {
```

```
  echo $email." is not a valid email address";
```

```
}
```

```
?>
```

```
<div>
```

```
  <input type="submit" name="submit" value="Submit">
```

```
</div>
```

```
<form method="POST" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

2. Create an array of doubles and integers. Loop through the array, converting each element to a floating-point number with a precision of 2. Right-align the output within a field of 20 characters.

3. Create a birthday countdown script. Given form input of month, day, and year, output a message that tells the user how many days, hours, minutes, and seconds until the big day.

4. Create a calculator script that enables the user to submit two numbers and choose an operation (addition, multiplication, division, or subtraction) to perform on them.

5. Use hidden fields with the script you created in activity 1 to store and display the number of requests that the user submitted.

6. Create a script that uses session functions to track which pages in your environment the user has visited.

7. Create a new script that will list for the user all the pages he/she has visited within your environment, and when.

8. Create a form that accepts a user's first and second name. Create a script that saves this data to a file.

9. Create a script that reads the data file you created in the first activity. In addition to writing its contents to the browser (adding a tag to each line), print a summary that includes the number of lines in the file and the file's size.

10. Draw a New Image, shapes and lines.

11. Create a New Image with Color Fills.

12. Draw A Basic Pie Chart and 3D Pie Chart
13. Creating a New Image from an Existing Image.
14. Creating an Image from User Input.
15. Creating an Image with Custom Font and Text