LESSON H3_EN. INTRODUCTION TO PHP LANGUAGE ESPECIALLY FOR eBUSINESS / eCOMMERCE. LEARNING TO MAKE SIMPLE SCRIPTS.

PARENT ENTITY: S.C.FOX I.E. s.r.l. , Bucharest, 4 Aurel Botea street, bl.B8, ground floor, ap.4, sector 3
Authors: Constantin Chersin, software engineer, FOX I.E. s.r.l. , Bucharest, 4 Aurel Botea street, bl.B8, ground floor, ap.4, sector 3
Mihaela Chersin, Educational Coordinator, FOX I.E. s.r.l. , Bucharest, 4 Aurel Botea street, bl.B8, ground floor, ap.4, sector 3
Email: office@fox.ro, Call number: +40-21-3232164, Office hour: Wednesday 10 a.m. – 11 a.m.

After the learning this lesson you will be richer with the following knowledge:

- ***The basic*** knowledge of the PHP scripting language so that developing a simple estore application becomes more easier.

CONTENT OF THE LESSON

1. Few words about programming in PHP
2. Basics in PHP scripting language.
3. How to connect PHP to MySQL

LEARNING OBJECTIVES:

After learning this lesson you will accomplish the ability :

- To do the simplest PHP scripts
- To start web site developing

1. Few words about programming in PHP

When you create a static web page, you simply write HTML code. Writing a dynamic page with PHP is similar, except you embed the PHP code inside of the HTML code. For this reason PHP is called an HTML-embedded scripting language.

2. Basics in PHP scripting language

2.1. Basic syntax

- Opening and closing PHP tags

PHP code is marked with specific tags. <?php opens up a PHP statement. This tells to web server that all statements that follow until the symbol combination ?> are PHP statements.

```
<?php   - opening PHP tag
?>      - closing PHP tag
```

Each statement in PHP must end with a semicolon: **;**.

- Comments in PHP

The purpose of comments is exclusive for your information. In comments you write remarks to the code you are doing. The PHP engine ignores these. Visitors of your site will be not seeing the comments in their browser.
There are two kinds of comment tags PHP:

- comment in one line having following sign symbol: // ... Here is your comment

Example 1:

```
<?php
// Welcome to my eStore!
?>
```

In the browser you will see:

- comment in multiple lines having following sign symbol : /*... Here is your comment… */

Example 2:

```
<?php
 /*
Welcome to my eStore!
Is nothing to view?
Is nothing, because this is a comment.
 */
 ?>
```

In the browser you will see:

2.2. Basic PHP statements

- echo - output one or more strings. echo() is not actually a function (it is a language construct), so you are not required to use parentheses with it.

Let's try to write a simplest script.
Open a new document in HTML-Kit and insert between <body> </body> tags, below PHP code:

Example 3: (
 is a html tag that we used to break the string.)

```
<?php
echo "Welcome to my eStore! <br> I offer some of the best candles on the web.";
// Here I welcome to my visitors
?>
```

(Note: We purpose to create a folder named "my examples" in C:\apachefriends\xampp\htdocs\. In this folder you can save all following examples that you are doing in this lesson. In this way you should see anytime the results of all your examples in the browser to the address: http://localhost/myexamples/.)

Then save it under the name: example3.php. Go to browser to the address: http://localhost/myexamples/. Click on examples3.php (or go directly to: http://localhost/myexamples/example3.php). In fig.2.2.is what you will see in the browser.

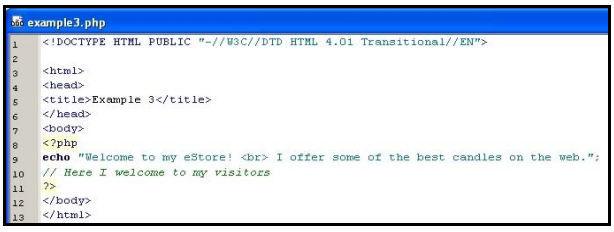

Fig. 2.1.Example3 embedded in html code saw in HTML-Kit

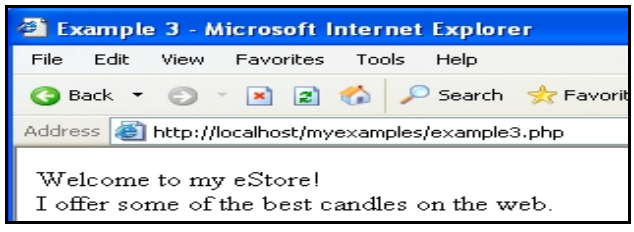

Fig. 2.2. Screenshot showing example3 in the browser

- print() - output a string. print() is not actually a real function (it is a language construct) so you are not required to use parentheses with its argument list.

- print_r() - displays information about a variable in a way that's readable by humans. If given a string, integer or float, the value itself will be printed. If given an array, values will be presented in a format that shows keys and elements.

2.3. Variables in PHP

All variables in PHP start with: $ (symbol). The $ symbol is followed by the name of the variable. The name must not start with a number. It must start only with lowercase or uppercase letter or underscore followed by any letters, numbers or

underscores. The variable name is case-sensitive. For example, the following variables: $words, $Words, $_words, are different

You assign the name of the variable. A suggestive name is better to use for the variable. This will help in your work.

It is not necessary to initialize variables in PHP however it is a very good practice. Un-initialized variables have a default value of their type - FALSE, zero, empty string or an empty array.

Example 4: (inside of echo statement you can use variables.)

```
<?php
$message="Welcome to my eStore!"; // create a variable and assign a string
echo  $message;                   // display the variable
?>
```

In the browser you will see:

```
Welcome to my eStore!
```

Example 5: (inside of print statement you can use variables.)

```
<?php
$name = "John";                      //create a variable and assign a string
print "Welcome $name to my eStore";
?>
```

In the browser you will see:

```
Welcome John to my eStore
```

(Note: using single quotes to the print statement will display the variable name, not its value of.)

Example 6: (you have the possibility to concatenate two or more variables together using the dot **.** operator)

```
<?php
$first_name="John";
$last_name="Doe";
$message="welcome to my eStore";
echo $first_name . " " . $last_name  . ", ". $message . "!"  ;
?>                        //display the variable concatenated
```

In the browser you will see:

```
John Doe, welcome to my eStore!
```

Example 7: (combination between string and a variable is doing with dot**.**)

```
<?php
$words="welcome";
echo  "You are always ".$words." to my eStore!";
?>
```

In the browser you will see:

```
You are always welcome to my eStore!
```

Example 8: a script for calculating the total price of 2 products

```
<?php
$firstprice=1.10;                     //create a variable and assign value 1.10
$secondprice=2.20;                    //create a variable and assign value 2.20
$totalprice=$firstprice+$secondprice;  //create a function witch make the variables total
echo "$totalprice";                   // display the total value
?>
```

In the browser you will see:

```
3.30
```

2.4. Data types and operators

- Data types

PHP supports eight primitive types.

- 4 scalar types: boolean, integer, float and string.
- 2 compound types: array and object.
- 2 special types: resource and NULL.

Most useful data types in PHP are:

- string - a string is series of characters. It is no problem for a string to become very large. A string literal can be specified in three different ways. The easiest way to specify a simple string is to enclose it in single quotes (the character '). You can also specified strings enclosing it in double quotes (the character ").
- array - is an ordered structure. An array can be created by the array( ) language-construct. It takes a certain number *key => value pairs*. These pairs are separated by comma: array([key =>] value , [key1=>]value1, ...) and define keys and values.

An array can be written as: $name_array[key] = value.
You can access each item in the array
(Note - key can be a string or an integer nonnegative and the value can be anything.)
(Note - when key is omitted, an integer index is automatically generated, starting with 0; when key is no omitted and it is an integer, next generate key will be the biggest integer key +1; when two identical key are defined, the last overwrite the first)

The following some examples will make you understand better arrays.

Example 9: key is omitted (the number listing of the array starts at zero, so the first item is 0, second is 1, third is 2)

```
<?php
$products = array('candle1', 'candles2', 'candles3');
echo $products[0];
echo $products [1];
echo $products [2];
?>
```

In the browser you will see:

```
candle1candles2candles3
```

Example 10: key is not omitted and is 3 (the number listing of the array starts at 3, so the first item is 3 second is 4, third is 5.)

```
<?php
$products = array(3 =>'candle1', 'candles2', 'candles3');
echo $products [3];
echo $products [4];
echo $products [5];
?>
```

In the browser you will see:

```
candle1candles2candles3
```

Example 11: (key is omitted)

```
<?php
 $products = array ('candle1', 'candle2', 'candle3');
print_r ($products);
?>
```

In the browser you will see:

```
Array ( [0] => candle1 [1] => candle2 [2] => candle3 )
```

Example 12: (key is not omitted and is 4)

```
<?php
$products = array (4 => 'candle1', 'candle2', 'candle3');
print_r ($products);
?>
```

In the browser you will see:

```
Array ( [4] => candle1 [5] => candle2 [6] => candle3) )
```

- Operators

  - Arithmetic operators

$a + $b	addition	is a sum of variable *a* and variable *b*
$a - $b	substraction	is a difference of variable *a* and variable *b*
$a * $b	multiplication	is a product of variable *a* and variable *b*
$a / $b	division	is a division of variable *a* and variable *b*
$a % $b	modulus	rest of variable *a* divided by variable *b*
-$a	negation	is a opposite of variable *a*

  - Comparison Operators

$a == $b	Equal	**TRUE** if $a is equal to $b.
$a === $b	Identical	**TRUE** if $a is equal to $b, and they are of the same type
$a < $b	Less than	**TRUE** if $a is strictly less than $b
$a > $b	Greater than	**TRUE** if $a is strictly greater than $b.
$a <= $b	Less than or equal to	**TRUE** if $a is less than or equal to $b.
$a >= $b	Greater than or equal to	**TRUE** if $a is greater than or equal to $b

  - Incrementing/Decrementing Operators

++$a	Pre-increment	Increments $a by one, then returns $a.
$a++	Post-increment	Returns $a, then increments $a by one.
--$a	Pre-decrement	Decrements $a by one, then returns $a.
$a--	Post-decrement	Returns $a, then decrements $a by one.

  - Assignment Operators

$a =b	basic assignment	means that the left operand gets set to the value of the expression on the rights

2.5. Constants

A constant is an identifier (name) for a simple value. As the name suggests, that value cannot change during the execution of the script. A constant is case-sensitive by default. By convention, constant identifiers are always uppercase. The name of a constant follows the same rules as any label in PHP.
The name of a constant follows the same rules as any label in PHP.

2.6. Control structures

- if - is the simplest conditional statement. It execute two operations:
  1. estimate a conditional expression and
  2. execute a specific instruction if and only if the conditional expression value is true

In PHP, looks like:

Example 13: (to see how works this script, give to variables “a” and “b” different values)

```
<?php
$a=4;
$b=3;
if ($a>$b) {             // apply a control structure function and if is true
echo "Is true";
 }
 ?>
```

In the browser you will see:

```
Is true
```

Example 14:

```
<?php
 $variable=3;
if($variable==1) {               // apply a control structure function
echo "Value of variable is 1";   // if is true write on the web page
 }
 ?>
```

In the browser you will see:

So: "if" expression is true, than PHP will execute echo; "if" is false, than PHP will ignore it.

- else - extends an” if" statement. Execute a statement in case the expression in "if" statement evaluates to be false.

Example 15: (to see how works this script, give to variables “a” and “b” different values)

```
<?php
$a=4;                   // create a variable and assign value 4
$b=9;                   // create a variable and assign value 9
if ($a> $b) {           // apply a control structure function
echo "Is true";         // if is true display on the web page
} else {                //otherwise
echo "Is not true";     // if is false display on the web page
}                       //end function
?>
```

In the browser you will see:

```
Is not true
```

- elseif – is a combination between “else” and “if”. The `if` statement evaluates the truth value of it’s argument. If the argument evaluated as true, the code following the `if` statement and will be executed. And if the argument evaluates is not true, is false and there is an `else` statement, then the code following the `else` statement and will be executed.

Example 16: (to see how works this script, give to variables “a” and “b” different values)

```
<?php
$a=8;                          // create a variable and assign value 8
$b=7;                          // create a variable and assign value 7
if ($a < $b) {                 // apply a control structure function “if”
echo "candle1";                // if is true display on the web page
} else if ($a == $b) {         //  otherwise if true
echo "candle2";                // display on web page
} else {                       //otherwise
echo "we have not candles";    //display on the web page
}
?>
```

In the browser you will see:

```
we have not candles
```

- for - is a most complex loop in PHP. Is an iterative instruction (execute in repeatable way the associated instructions). With other words, does something for many times. This instruction has three parameters: initialization, condition and increment.

Example17:

```
<?php
for ($variable = 1; $variable <= 10; $variable++)
{
echo $variable;
}
?>
```

In the browser you will see:

```
12345678910
```

The above script executes something for ten times.
Let’s explain this code:
for ($variable = 1; $variable <= 10; $variable++):

$variable = 1 assign an initial value 1 to $variable;
$variable <= 10 establish 10 as the maximum value of $variable;
$variable++ increment value of $variable with 1.
For each value of the variable between 1 and 10 will be executed the statement between curly brackets { }

Example 18:

```
<?php
for ($variable = 1; $variable <= 3; $variable++)  {
echo "Welcome!";
}
?>
```

In the browser you will see:

```
Welcome! Welcome! Welcome!
```

 while – is the simplest loop in PHP and makes exactly what makes “if” statement, but in other way. The while statement will execute a part of code repeatedly if and as long as the while statement evaluates as true. In PHP looks like:

Example 19:

```
<?php
$variable = 1;           // assign an initial value 1
while ($variable <= 10)  // as long as  $variable is less or equal with 10
{                        // will be executed the statement between curly brackets
 echo  $variable ;       // display variable
$variable++;             // incremented with 1
}
?>
```

In the browser you will see:

```
12345678910
```

 foreach - is a construction which work only with array .

Example 20:

```
<?php
$employ= array(
"John" => manager,
"George" => programmer,
"Michael" => economist,  //for each employ name is assigned a job
);
foreach ($employ as $name => $job) {
echo "$name =>is the $job<br>";          //display the name with assigned job
}
?>
```

In the browser you will see:

```
John =>is the manager
George =>is the programmer
Michael =>is the economist
```

 include - the include() statement includes and evaluates the specified file. In PHP, looks like:

<?php includes “filename.php"; // will include filename.php which reside in the same folder with the file that requires it ?>

This function is useful, especially, in making a good structure and reducing the code of the web site. That means that all what is repeatable in the web site pages, can be written in a single file. (Example: the menu of a web site can be a repeatable code in all pages). So, instead to rewrite the code in every page you can just include the file which contains menu code in every page.

 require - require an external file on our current file. In PHP, looks like:

<?php require “filename.html"; ?>

It is similar with “include”. The difference is in the way they handle failures:

- "include" failure generates and advertising but it follows up to work
- "require" failure stops the execution of the script

2.7. Predefined variables

PHP provides a large number of predefined variables to any script, which it runs.
Predefined variables are variables that have been set by PHP and usually provide information about the HTTP request or response. PHP scripts can access them. To see all these variables use the script: <?php phpinfo(); ?> and you will see them in the browser, among other information. They are also listed in the PHP manual under 'Predefined Variables'.

- $_SESSION

Session support in PHP consists of a way to preserve certain data across subsequent accesses. This enables you to build more customized applications and especially eCommerce web sites.
$_SESSION, are associative array variables, which are currently registered to a script's session.

- $_COOKIE

Is available in all scopes throughout a script.An associative array of variables passed to the currentscript via HTTP cookies.

- $_GET

Is available in all scopes throughout a script. An associative array of variables passed to the current script via the HTTP GET method. Variables provided to the script via URL query string.

- $_POST

Is available in all scopes throughout a script. An associative array of variables passed to the current script via the HTTP POST method. Variables provided to the script via HTTP POST.

- $_REQUEST

Is available in all scopes throughout a script. An associative array consisting of the contents of $_GET, $_POST, and $_COOKIE. Variables provided to the script via the GET, POST, and COOKIE input mechanisms, and which therefore cannot be trusted.

3. How to connect PHP to MySQL database.

Web server and MySQL are complete different entities. You need to make them to work together. This, you can do with the help of PHP scripting language. MySQL database stored the informations. PHP must extract/insert these informations. Web server sends/take the informations in/from the browser.
To extract/insert the informations, PHP must connect to MySQL, and access the database.

The procedure that must be followed in working of PHP with MySQL database include three phases:

- PHP connect to MySQL and select the MySQL database
- PHP operate with MySQL database
- PHP close the connection with MySQL

3.1 PHP connects to MySQL and selects the database

For the connection to MySQL you need to know:

- dbserver: the name of the server where MySQL is installed
- dbuser: the user having rights to access database
- dbpass: the password of the user

```
mysql_connect ("dbserver", "dbuser", "dbpass") or die(mysql_error());
```

To access (select) the MySQL database you need to know:

- dbname: the name of the database

```
mysql_select_db("database") or die(mysql_error());
```

(Note: you remember that you created your database estore_db with phpMyAdmin in lesson 2)
So, the connection to your database "estore_db" can be done with:

```
<?php
$dbserver = "localhost";    //create the variable: $dbserver and assign
                            //the value: localhost
$dbuser = "root";    //create the variable: $dbuser and assign the value: root
$dbpass = " ";       //create the variable: $dbpass and assign the value:
                     // we remember that our user root was installed with no password
$dbname = "estore_db";   //create the variable: $dbname and assign the value: estore_db
```

```
$connect = mysql_connect($dbserver, $dbuser, $dbpass) or die(mysql_error()) ;  //connect
                                                                                //to MySQL
$select = mysql_select_db($dbname, $connect) or die(mysql_error()) ;  //select the
                                                                      //database

?>
```

You will save this sequence code in a php file named config.inc.php. You will use it, ulterior in your eStore website pages with "include" function (include "config.inc.php";) and also in below examples and exercises.

(Note: whenever you will work with MySQL and need to connect to a database use the same type of syntax. You can change the variables function of your database configuration.)

3.2. PHP operate with MySQL database

There are many operations that can be done with a database. We give you some examples:

- INSERT with this MySQL command you can insert records in database.

INSERT INTO table_name (column1, column2,...) VALUES (value1, value2,...);

1.)- Script PHP, to insert a product: book with the price: 3.5 in "products" table:

```
<?php
include "config.inc.php";      // connect and select the MySQL database
// here we insert the product book with price 3.5
mysql_query("INSERT INTO products(product, price) VALUES('book', 3.5) ") or die(mysql_error());
mysql_close($connect);        // close the connection with database
?>
```

2.)- Script PHP, to insert a product: pencil with the price: 1.2 in the "products" table

```
<?php
include "config.inc.php";             // connect and select the MySQL database
$product = "pencil";          // our intention is to insert the product pencil
$price =       "1.2";                 // with the price 1.2
// here we insert the product pencil and price 1.2
$query="INSERT INTO products (product, price) VALUES ('$product' , '$price')";
if (!mysql_query($query)) {   // if the insert has not success
       die(mysql_error());           // operation to take if not succeeded
} else {
       echo "The product was added"; // operation to take if succeeded
}
mysql_close($connect);        // close the connection with database
?>
```

- SELECT with this MySQL command you can select records from database.

SELECT * FROM table_name WHERE conditions

3.)- Script PHP, to select all products with their prices from the "products" table:

```
<?php
include "config.inc.php";                          // connect and select the MySQL database
$result = mysql_query("SELECT * FROM products") or die(mysql_error());  // select all records from the
                                                                        //"products" table
while($row = mysql_fetch_array( $result )) {    // keeps selecting the next row until there are no more
                                                //to select
echo $row['product']." ".$row['price']."<br>";  // display in the browser the contents of each row
}
mysql_close($connect);        // close the connection with database
?>
```

4.)- Script PHP, to select all products from the "products" table whose price is 3.5

```
<?php
include "config.inc.php";     // connect and select the MySQL database
// select all the products from the table "products" whose price is 3.5
$result = mysql_query("SELECT product FROM products WHERE price=3.5") or die(mysql_error());
while($row = mysql_fetch_array( $result )) {    // keeps selecting the next row until there are no more
                                                //to select
echo $row['product'] ;        //display in the browser
}
mysql_close($connect);        // close the connection with database
```

?>

- UPDATE with this MySQL command you can update (modify) records from database.

UPDATE table_name SET column1='new value of column1', column2=' new value of column 2' WHERE conditions;

- DELETE

DELETE FROM table_name WHERE conditions;

3.3. PHP closes the connection to MySQL

If you are no more interested to operate with MySQL database, PHP must close the connection with MySQL:

mysql_close();

So, to close the connection with your database "estore_db" you must use following command:

```
mysql_close($connect);
```

mysql_connect - is a function which realize the connection to a MySQL server (it has five facultative parameters: 1. hostname; 2. username; 3. password; 4. logical type parameter and indicate to PHP that if exist a connection to the same server to create an other one not to use the existing one; 5. integer type parameter and will represent the properties of the connection.
mysql_select_db - is a function used to select a database for which you want a connection to a MySQL server (it has two parameters: 1. name of database; 2. an access identification to the connection to MySQL server). This function returns logical value FALSE if the selection failed and logical value TRUE if the database was selected.
mysql_query - is a function used to interrogate a database (it has three parameters: 1. the request to the MySQL server; 2. an access identification to the connection to MySQL server; 3. is integer type and represent the manner in which will be return the result). The third parameter may has following values: MYSQL_USE_RESULT or MYSQL_STORE_RESULT.
mysql_close - is a function which realize the closing of the connection with a MySQL server (it has one parameter: an access identification to the connection to MySQL server)
mysql_create_db - is a function used to create a new database to the MySQL server (it has two parameters: 1. name of database; 2. an access identification to the connection to MySQL server). This function return logical value FALSE if the connection failed and logical value TRUE if the connection is successfully.
mysql_drop_db - is a function used to delete a database (it has two parameters: 1. name of database; 2. an access identification to the connection to MySQL server). This function returns logical value FALSE if the database couldn't be deleted and logical value TRUE if the database was deleted.

Key Point Summary Conclusions and Recommendations

With a database program such as MySQL and with basic knowledge of HTML and PHP you will be capable to create one of most complex eCommerce web site.
We recommend you to download PHP manual, it would help you very much in learning PHP.

Study Guide

ESSENTIAL QUESTIONS FOR THE VERIFICATION OF THE ACCOMPLISHED KNOWLEDGE

1. Make a PHP script to insert in the table"products" of your database "estore_db", following two products: first product: eraser with price 3.50; second product: sharpener with price 1.20
2. Make a PHP script to insert in the table"products" of your database "estore_db", following four products: first product: pen with price: 12.30; second product: satchel with price: 10; third product: sheet with price:0.2; fourth product: refill with 12.30.
3. Using method POST, make a PHP script to insert products in the table "products" of your database "estore_db"
4. Make a PHP script to select all products from the table "products".
5. Make a PHP script to select the price of the product: pen from the table products.

BIBLIOGRAPHY. REFERENCES.

[1.] Larry Ulman, *PHP for the World Wide*, WebPeachpit Press, April 2001 (Publisher)ISBN 0-201--72787-0
[2.] Rasmus Lerdorf, *PHP Pocket Reference*, O'Reilly & Associates, Jan 2000, (Publisher)ISBN 1-56592-769-9

SUPPLEMENTARY IMPORTANT BIBLIOGRAPHY. REFERENCES.

[SUP.1]http://www.php.net/ - Everything what you want to know about PHP
[SUP.2]http://www.php.net/manual/en/language.control-structures.php - here you will find control structures in PHP
[SUP.3]http://www.php.net/download-docs.php - from here you can download PHP manual (we recommand to downlod it)
[SUP.4]http://www.phpclasses.org/mysql_connection - class - mysql connection
[SUP.5]http://www.phpbuilder.com/ - PHP tutorials, templates, PHP manuals.

SUPPLEMENTARY INDICATIONS ABOUT THE CONTENT OF THE LESSON

To save all the scripts from this lesson in one folder under \htdocs\ , and then to access one by one in browser, to see and analyze the result.

ANSWERS TO THE QUESTIONS

1. The PHP script to insert in the table"products"of your database estore_db, following two products: first product: eraser with price 3.50; second product: sharpener with price 1.20, is:

```
<?php
include "config.inc.php";            // connect and select database
// insert the product: eraser and its price 3.50 and the product: sharpener and its price 1.20 in the table products:
mysql_query("INSERT INTO products(product, price) VALUES('eraser', '3.5' ) ") or die(mysql_error());
mysql_query("INSERT INTO products(product, price) VALUES('sharpener', '1.2' ) ") or die(mysql_error());
mysql_close($connect);
?>
```

2. The PHP script to insert in the table "products"of your database estore_db, following four products: first product pen with price 12.30; second product satchel with price 10; third product sheet with price 0.2; fourth product refill with 12.30, is:

```
<?php
// connect and select database
include "config.inc.php";
// insert all the products each by each in the table "products"
mysql_query("INSERT INTO products(product, price) VALUES('pen', '12.3' ) ") or die(mysql_error());
mysql_query("INSERT INTO products(product, price) VALUES('satchel', '10' ) ") or die(mysql_error());
mysql_query("INSERT INTO products(product, price) VALUES('sheet', '0.2' ) ") or die(mysql_error());
mysql_query("INSERT INTO products(product, price) VALUES('refill', '12.3' ) ") or die(mysql_error());
mysql_close($connect);
?>
```

3. The PHP script using method POST, to insert products in the table "products"of your database estore_db, is:

```
<?php
include "config.inc.php";
$product=$_POST['product'];
$price=$_POST['price'];
//insert products and price
$insert_new_product ="INSERT INTO products (product, price) VALUES ('".$_POST['product']."', '".$_POST['price']."')";
if(!mysql_query($insert_new_product)) {
die(mysql_error());
} else {
?>
<form method="POST" action="insert_ex3.php">     //form to add product and price by method POST
Product: <input type="text" name="product"><br>
Price: <input type="text" name="price"><br>
<input type="submit" value="Add product">
</form>
<?php
 print("Successfully was added product: Product: <b> ".$_POST['product']." </b> . Price : <b> ".$_POST['price']." </b>");
}
?>
```

(Note: save it and go in the browser from where you can insert any product with its price)

4. The PHP script to select all products from the table "products", is:

```
<?php
include "config.inc.php";                    // connect and select the MySQL database
// select all the products from the table "products"
$select = mysql_query("SELECT product FROM products ") or die(mysql_error());
while($row = mysql_fetch_array( $select )) {          // keeps selecting the next row until there are no more to select
echo $row['product']."<br>" ;                    //display in the browser the contents in one column
}
mysql_close($connect);
?>
```

5. The PHP script to select the price of the product pen from the table products, is

```
<?php
include "config.inc.php";                    // connect and select the MySQL database
// select the price for product pen from the table: "products"
$result = mysql_query("SELECT price FROM products WHERE product='pen'") or die(mysql_error());
while($row = mysql_fetch_array( $result )) {         // keeps the next row until there are no more to select
echo $row['price']."<br>";
}
mysql_close($connect);
?>
```

WORDS TO THE LEARNER:

By learning you will teach; by teaching you will learn.