PHP Introduction

DWES UD2

3.- Basic PHP Features

Instruction blocks

In PHP you can use braces **{ }** to group statements.

Using control structures, it is possible to decide whether a block of instructions is executed or not, or whether the execution of said block must be repeated.

3.- Basic PHP Features

Control structures

PHP, even though it is a **script** programming language, has, like any other high-level programming language, statements that allow altering the predefined flow of execution (instruction by instruction from top to bottom).

You already know these sentences from other programming languages such as Java.

if **switch** **while** **do/while** **for**

3.- Basic PHP Features

```
<?php
    if ($a < $b)
        echo "a es menor que b";
    else if ($a > $b)
        echo "a es mayor que b";
    else
        echo "a es igual a b";
?>
```

```
<?php
    if ($a < $b) {
        echo "a es menor que b";
    } else if ($a > $b) {
        echo "a es mayor que b";
    } else {
        echo "a es igual a b";
    }
?>
```

3.- Basic PHP Features

```
<?php
        if ($a < $b)
    {
            echo "a es menor que b";
    }
    else if ($a > $b)
    {
            echo "a es mayor que b";
    }
    else
    {
            echo "a es igual a b";
    }
?>
```

3.- Basic PHP Features

```
<?php
	$a = 0;
	switch ($a) {
		case 0:	echo "a vale 0";
			break;
		case 1:	echo "a vale 1";
			break;
		default:	echo "a no vale 0 ni 1";
	}
?>
```

3.- Basic PHP Features

```
<?php
	$a = 0;
	switch ($a) {
		case 0:	echo "a vale 0";
		case 1:	echo "a vale 1";
		case 2:	echo "a vale 2";
			break;
		default:	echo "a no vale 0 ni 1";
	}
?>
```

What does the above script do?

3.- Basic PHP Features

```
<?php
        $a = 1;
        while ($a < 8) {
                $a += 3;
        }
        echo $a;
?>
```

What does the above script do?

3.- Basic PHP Features

```
<?php
    $a = 5;
    do {
        $a -= 3;
    } while ($a > 10);
    echo $a;
?>
```

What does the above script do?

3.- Basic PHP Features

```
<?php
        for ($a = 0; $a<10; $a++) {
                echo $a;
                echo "<br>";
        }
?>
```

What does the above script do?

3.- Basic PHP Features

```
<?php
        for ($a = 5; $a<10; $a+=3) {
                echo $a;
                echo "<br>";
        }
?>
```

What does the above script do?

3.- Basic PHP Features

Functions

Functions are blocks of code that, being found elsewhere, can be executed by making what is known as a **function call**.

A function call has already been used in class:

```
<?php
        phpinfo( );
?>
```

And we have seen how the documentation works to consult functions predefined by the PHP language.

3.- Basic PHP Features

Functions

In addition to the functions predefined by PHP, you can also create your own functions.

It is **NOT** necessary to define the functions before using them. They simply have to be in the **same script** in which the call is made, either directly or in an external file using an **include** or **require** statement.

Let's see an example of a proper function:

3.- Basic PHP Features

Functions

```
<?php
        $precio = 10;
        precio_con_iva();

        function precio_con_iva() {
                global $precio;
                $precio_iva = $precio * 1.21;
                echo "El precio con IVA es ". $precio_iva;
        }
?>
```

*The use of **global** was already explained in class and for this use is **NOT** recommended**.**

3.- Basic PHP Features

Functions - Arguments/parameters

You can pass values to functions by using **arguments**.

The arguments are a comma-separated list of variables.

The data type of the variable is not indicated.

If necessary you can make the function **return a value**.

3.- Basic PHP Features

Functions - Arguments/parameters

```
<?php
        function precio_con_iva($precio) {
                $precio_iva = $precio * 1.21;
                echo "El precio con IVA es ". $precio_iva;
        }

    $precio = 10;
    precio_con_iva($precio);
?>
```

3.- Basic PHP Features

Functions - Arguments/parameters

```
<?php
        function precio_con_iva($precio) {
                $precio_iva = $precio * 1.21;
                return $precio_iva;
        }

    $precio = 10;
    $precio_final = precio_con_iva($precio);

    echo "El precio con IVA es ". $precio_final;
?>
```

3.- Basic PHP Features

Functions - Arguments/parameters

```
<?php
        function precio_con_iva($precio) {
                $precio_iva = $precio * 1.21;
                return $precio_iva;
        }

    $precio = 10;
    echo "El precio con IVA es ". precio_con_iva($precio);
?>
```

3.- Basic PHP Features

Functions - Arguments/parameters

Sometimes it may be interesting to set default values for the arguments, so that if the function call is made no value is indicated, then the default value will be applied.

This behavior was seen with the **date function.**

```
<?php
        echo date("Y");
        echo date("Y", time());
?>
```

In that case the arguments with default values must go at the end.

3.- Basic PHP Features

Functions - Arguments/parameters

```
<?php
        function precio_con_iva($precio, $iva=0.21) {
                $precio = $precio * (1 + $iva);
                return $precio;
        }

        $precio = 10;
        $precio_iva = precio_con_iva($precio);
        echo "El precio con IVA es ".$precio_iva
?>
```

3.- Basic PHP Features

Functions - Arguments/parameters

Attributes can be passed by **value** or by **reference**. In the examples seen so far, the passed by value has been used.

This means that the original variable will not change its value.

The pass can be made by **reference** and then the value of the original variable can change.

It is not recommended to perform this action if you do not have a very high knowledge of the programming language.

3.- Basic PHP Features

Functions - Arguments/parameters

```
<?php
        function precio_con_iva(&$precio, $iva=0.21) {
                $precio = $precio * (1 + $iva);
        }

        $precio = 10;
        $precio_iva = precio_con_iva($precio);
        echo "El precio con IVA es ".$precio_iva
?>
```

Exercise

Make the web project have the footer in the footer.inc.php file.
Have your first and last name.
Introduce the current date, but with the days and months in Spanish.

For the date, it is recommended to use a switch instruction, in which, knowing the day number and month number, the day and month are displayed written in Spanish.

The date must appear as follows:

Miércoles, 20 de septiembre de 2023

Exercise

Create a document called **count.php** in the web project. This document must have the same header and footer as the rest of the files. A list of numbers from 1 to 30 should appear in the body of the page (use a for loop).

After the list of numbers, the factorial of 5 must appear, use a loop for calculation and data display. The result that should appear on the screen should be:

5! = 5 x 4 x 3 x 2 x 1 = 120

Exercise

```
function calculo_numerico ($a, $b=5, $c) {
        $resultado = $a * $b * $c;
        return $resultado;
}
```

Is the definition of this function correct? Indicate it, explaining why or why not and correct it if it is not so.

3.- Basic PHP Features

Arrays

Arrays allow you to store several values of the same data type. Each member of the array is stored in a position.

Arrays can be of 3 types:

- Numeric: all indexes are integer (whole numbers)
- Associative: all indexes are strings
- Mixed: has integer and string indexes

The position (index/key) can be a **value** or a **string**

There are several ways to declare an array.

3.- Basic PHP Features

Arrays - declaration

```
$colores = array("rojo", "verde", "azul");

$colores = ["rojo", "verde", "azul"];

$ciclos = array("DAW" => "Desarrollo web", "DAM" => "Desarrollo multiplataforma");

$colores = [1 => "rojo", 5 => "verde", "0" => "azul"];

$colores[] = "azul";
```

3.- Basic PHP Features

Arrays

If the index is not indicated, it will be numeric and start at zero.

If the associative array is declared, it can no longer be accessed with numerical positions.

You can mix numeric and associative keys. If the key is associative and is a number, the element could be accessed with the number.

3.- Basic PHP Features

Arrays

Usually, if you want to access an array through its numerical index and also through an associative key, it is usually done in the following way.

In fact, there are system functions that when they return an array they do so in this way.

$ciclos = array(0 => "Desarrollo web", "DAW" => "Desarrollo web", 1 => "Desarrollo multiplataforma", "DAM" => "Desarrollo multiplataforma");

3.- Basic PHP Features

Arrays

You can add elements to the end of an array in the following way:

```
$personajes = ["Luke", "Leia"];
$personajes[] = "Han Solo"; // se asigna la clave 2
```

Using the key (index) you can also add elements to an array:

```
$personajes[3] = "Darth Vader";
```

To access the positions of the arrays, the square bracket notation **[ ]** is used.

```
$colores = ["rojo", "verde", "azul"];
echo $colores[0];
```

3.- Basic PHP Features

Arrays

PHP is very flexible in creating arrays, and it is not necessary to indicate the size of the array.

It is also not necessary to indicate the key, PHP will assign the following one automatically.

```
$colores[0] = "rojo";
$colores[1] = "azul";
$colores[5] = "marron";
$colores[ ] = "verde";   // ¿Qué clave tiene “verde”?

$numeros[ ] = "uno";
$numeros[ ] = "dos";
```

3.- Basic PHP Features

Arrays

Remember that using the **print_r( )** or **var_dump()** function you can display the entire contents of the array, including its keys. Although this is a function only recommended for debugging scripts.

To traverse an array, a **for** loop can be used as is common in all programming languages. In this case, it is necessary to know the length of the array. For this, the **count($array)** function is available, which will return an integer.

3.- Basic PHP Features

Arrays

There is another more advisable way to traverse an array, it is by using the **foreach** function. There are two ways to use it:

```
$numeros = ["uno", "dos", "tres"];

foreach($numeros as $valor) {
        echo $valor ."<br>";

foreach($numeros as $clave => $valor) {
        echo "[". $clave ."]". $valor ."<br>";
```

3.- Basic PHP Features

Arrays

In PHP there are also **multidimensional arrays**. The operation is exactly the same as normal arrays.

To access the data you simply have to use the appropriate key for each dimension.

It must be taken into account that the size of the dimensions of the arrays is not declared and that the key could be mixed with an index or associative.

```
$array =        [[1, 2, 3],
                [4, 5, 6],
                [7, 8, 9]];
echo $array[1][0];
```

3.- Basic PHP Features

Arrays

In PHP there are also **multidimensional arrays**. The operation is exactly the same as normal arrays.

To access the data you simply have to use the appropriate key for each dimension.

It must be taken into account that the size of the dimensions of the arrays is not declared and that the key could be mixed with an index or associative.

```
$array =    [["Ana", "Luis", "Marta"],
            "colores" => ["amarillo", "verde"],
            ["saludo" => "hola", "despedida" => "adiós", "¿Qué tal estás?"]];
```

3.- Basic PHP Features

Arrays

Elements can be removed from the array with the unset function

```
unset($array[2]);
unset($personaje["nombre]);
```

If it is a numeric array, the indexes will have to be recreated using the array_values function, to avoid problems if the entire array is routed:

```
$array = array_values ($array);
```

Exercise

Create a document called **server.php** in the web project.

This document must have the same header and footer as the rest of the files.

The entire content of the **$_SERVER** variable (which, as we have already seen, is an array) must appear in the body of the page **inside a table**.

Cannot use **print_r** or **var_dump** functions. You will have to use a loop to loop through the array.

Exercise

Modify the date generation in the **footer.inc.php** file so that you have the names of the days and months stored in a **multidimensional array.**

This way you will not need a switch to convert to Spanish, you will simply access the corresponding position according to the day or month number.