Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN, FORMACIÓN PROFESIONAL Y DEPORTES

Plan de Recuperación, Transformación y Resiliencia

Junta de Andalucía
Consejería de Desarrollo Educativo y Formación Profesional

Unit 2. PHP Introduction

I.E.S. Zaidín Vergeles (Granada)

PHP (PHP: Hypertext Preprocessor)

PHP was created in 1994 by Ramus Lerdorf.

Features:

- Scripting language
- Backend
- Free
- Open source

Rasmus Lerdorf

PHP (PHP: Hypertext Preprocessor)

Scripting languages allow to write sequences of instructions which are read, interpreted and executed one by one by the processor.

PHP (PHP: Hypertext Preprocessor)

TRY IT!

1. Write a simple *HTML* document and save it with *.php* extension in a folder inside *htdocs*.
2. Add *<?php echo "Hello!" ?>* inside *<body>* and save the document.
3. Open *XAMPP* and start *Apache*.
4. Open your browser and write *localhost/* followed by the name of the file in the address bar.
5. Read the source code and compare it with the one you wrote. What has happened? What is the purpose of *echo*?

PHP Syntax

To point out the beginning and the end of a script, we write:

```
<?php
    ….
?>
```

Instructions separated by semicolon

PHP Syntax

Comments are parts of the code that are ignored by the application server. However they are essential to describe the meaning of them, for example, when we share code with other developers or when we reuse it after a long time.

PHP Syntax

Comments are written in differents ways depending on their length:

- Comments in one line can be preceded by **//** or **#**
- Comments in more than one lines begin with **/*** and end with ***/**.

PHP Syntax

Take one of the documents you used before and try these three ways of adding comments to the code. Open them in your browser and watch the output you get.

PHP Syntax

The sentences that include content to an HTML document are *echo* and *print*.

They are similar although *echo* is usually executed faster.

Both are accompanied by parameters. When these are literals, they will be written between single or double quotes.

PHP Syntax


```
<?php
  echo “<h1>I.E.S. Zaidín Vergeles</h1>”;
  echo “<h2>News</h2>”;
  echo “<h3>Reunión informativa para familias sobre…</h3>;
  echo “<p>El día 1 de octubre a las 17:00 horas…</p>”;
?>
```

PHP Syntax

Change echo by print in the former code and open the code in your browser. What output do you get?

PHP Syntax

There is a difference between *echo* and *print*:

Only the first one permits including more than one parameter in the sentence.

In this case the parameters will be separated by comma.

PHP Syntax

TRY IT!

```
<?php
  echo “<h1>I.E.S. Zaidín Vergeles</h1>”, “<h2>News</h2>”,
   “<h3>Reunión informativa para familias sobre…</h3>,
   “<p>El día 1 de octubre a las 17:00 horas…</p>”;
?>
```

PHP Syntax

Is it possible to write quotation mark as one of the characters of a parameter of echo or print?

PHP Syntax

While the execution of a script, it is often necessary to store different kinds of data to use them. They can be saved in these two types of “containers”:

- **Variables** which can change their value during the execution.
- **Constants** which stay always the same.

PHP Syntax

Variables and constants are stored in specific places of memory by the operating system and are named by the developer following these rules:

- Their first character is a letter or underscore (_).
- The rest of characters must be letters, digits or underscore.
- Uppercase and lowercase letters are considered different characters (case sensitivity).

PHP Syntax

TRY IT!

```
<html lang="en">
  <head>
    <title>Sum</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <?php
      $a = 4;
      $b = 3;
      echo $a + $b;
    ?>
  </body>
</html>
```

- What are the names of the variables included in this code?
- What character do we need to write before each variable?
- Why didn't we write quotation marks in the parameter of *echo*?

PHP Syntax

In order to differentiate easily from variables, constants are commonly written in uppercase but it is not compulsory to do it.

There are two ways to declare a constant:

- define()
- const

PHP Syntax

TRY IT!

```
<?php
 define (“GREETING1”, “Hello”);
 echo GREETING1;
?>
```

```
<?php
 const GREETING2= “Bye”;
 echo GREETING2;
?>
```

PHP Syntax

PHP uses these types of data:

- String
- Integer
- Float
- Boolean
- NULL
- Array
- Object
- Resource

PHP Syntax

TRY IT!

```
<?php
    $x = 5985;
    var_dump($x);
    $y = 8.75;
    var_dump($y);
    $z = false;
    var_dump($z);
    $w = “Hi!”;
    var_dump($w);
?>
```

- What is the type of each variable in this code?
- What is the purpose of ***var_dump***?
- What parameter do we write with this function?

PHP Syntax

Operators in PHP

- Arithmetic
- For strings
- Assignment
- Incrementing/Decrementing
- Comparison
- Logical

Have a look at this *web page* in order to read about PHP operators.

PHP Syntax

Write an HTML document with the PHP code necessary to display:

- The length of a circumference with a radius of 12 centimeters and the area of its circle.
- The results of the addition, subtraction, multiplication, division, modulus and power of 7 and 4.
- A greeting using two variables. The first one contains “Good” and the second one “morning”.

PHP Control statements

Until now, our scripts have been executed line by line, from the first one to the last one.

However, there are many cases in which we need to change the order of execution of the sentences of a script. We can manage it by using ***control structures***, also called ***control statements***.

PHP Control statements

In a ***simple conditional statement***, a code is executed if a condition is true.

```
if (condition)
 {
    instructions that will be executed if the condition is true
 }
```

PHP Control statements

TRY IT!

```
<?php
   $h = date("H");
      if($h >= "22") {
        echo "Good night";
       }
?>
```

- What is the purpose of this code?
- What can we use ***date()*** for?
- Visit this web to find out the meaning of its parameter.

PHP Control statements

In a ***double conditional statement***, a code is executed if a condition is true and another code if it is false.

```
if (condition)  {
    instructions that will be executed if the condition is true  }
else {
    instructions that will be executed if the condition is false }
```

PHP Control statements

Write an HTML document with a PHP script that stores a person's name and age in two variables and adds a paragraph telling if the person is an adult or not.

PHP Control statements

In the blocks of instructions to execute in a simple or double conditional statement, it is possible to write other simple or double conditional statements, in other words, we can **nest** several of them.

PHP Control statements

```
if (condition1)  { instructions to execute if condition1 is true  }
else {
    if (condition2)  {
       instructions to execute if condition1 is false and condition2 is true  }
    else {instructions to execute if condition1 and condition2 are false }
}
```

PHP Control statements

An equivalent code to the former one is this:

```
if (condition1)  { instructions to execute if condition1 is true  }
elseif (condition2)  {
        instructions to execute if condition1 is false and condition2 is true  }
    else {instructions to execute if condition1 and condition2 are false }
}
```

PHP Control statements

Write a PHP script that gets the time from the operating system and inserts the text:

- “Good morning”, if it is before midday.
- “Good afternoon”, from 12:00 until 21:59.
- “Good night”, from 22:00 onwards.

PHP Control statements

In a ***multiple conditional statement***, a block is executed depending on the value of a expression.

The word ***break*** forces the processor out of the *switch*.

The word ***default*** allows to add a block that will be executed when the expression doesn’t match any of the values. It is optional.

PHP Control statements

```
switch (expression)  {
   case "value1": {instructions if expression equals value1;
     break;  }
   case "value2": {instructions if expression equals value2;
     break;  }
   …
   default { instructions if there is no match}
}
```

PHP Control statements

Write a PHP script that takes the day of the week from the system and inserts this phrase in a paragraph:

“Hoy es “ followed by the name of the day in Spanish.

PHP Control statements

Sometimes we need to repeat the execution of a block a specific number of times or while a condition is true or false. In these cases, we will use **loops**. There are different kinds of them so it is important to choose the best for each situation in order to take advantage of the possibilities of the language.

PHP Control statements

To execute a block of instructions a specific number of times, we use the ***for*** sentence.

```
for (initializing, test, increment)
 {
    instructions that will be executed in each iteration
 }
```

PHP Control statements

TRY IT!

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

- Explain the purpose of this code.
- Improve it in order not to insert the last space character.

PHP Control statements

Write PHP scripts to produce:

- All the even numbers between 50 and 80.
- All the integers from 17 to -17.
- The multiplication table of 6.

PHP Control statements

To execute a block of instructions while a condition is true, we use the ***while*** sentence.

```
while(condition)
 {
    instructions that will be executed in each iteration
 }
```

PHP Control statements

Rewrite the scripts in slide 39 using ***while*** loops.

PHP Control statements

In the previous loop the condition is examined at the beginning of the block of instructions, but it is possible to do it at the end of it, with this other expression:

```
do
 {
    instructions that will be executed in each iteration
 }
while (condition)
```

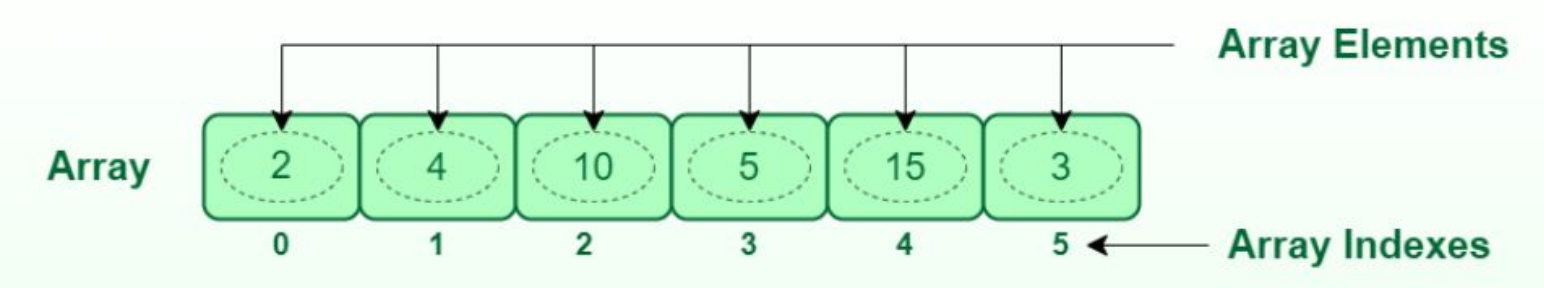
PHP Control statements

Rewrite the scripts in slide 39 using **do** … ***while*** loops.

What is the minimum number of times that a block of instructions is executed in each kind of loop?

PHP Control statements

An array is a collection of elements stored in contiguous memory locations.



PHP Control statements

To loop through an **indexed** array we can use this statement:

```
foreach ($array as $valor)
 {
    instructions that will be executed for each element of the array
 }
```

PHP Control statements

TRY IT!

```
<?php
    $drinks = array("Coffee","Tea","Water","Beer","Wine","Soft drink");
        echo "<h1>DRINKS</h1>";
        echo "<ul>";
        foreach ($drinks as $b) {
           echo "<li>", $b, "</li>";
        }
        echo "</ul>";
?>
```

PHP Control statements

- What is the purpose of the former code?
- At the end of the script, add an instruction to apply the *var_dump* function to *$drinks* and explain the output you get.

PHP Control statements

A store has saved its last fortnight sales in an array. Write a script to calculate the total amount and to carry it to the output.

PHP Control statements

Associative arrays have indexes named by the developer. To loop through them, we can write this:

```
foreach ($array as $index => $valor)
 {
    instructions that will be executed for each element of the array
 }
```

PHP Control statements

TRY IT!

```
<?php
    $computers = array(“A210” => 28, ”A211” => 30, ”A212” => 32, ”A213” => 28);
    echo “<table> <caption>COMPUTERS IN CLASSROOMS</caption>”;
    echo “<tr> <th>Classroom</th> <th>Number of computers</th></tr>”;
    foreach ($computers as $classroom => $numcomputers) {
            echo “<tr><td>”, $classroom, “</td><td>”, $numcomputers, “</td></tr>” ;
     }
    echo “</table>”;
?>
```

PHP Control statements

- What is the purpose of the former code?
- At the end of the script, add an instruction to apply the *var_dump* function to *$computers* and explain the output you get.
- Try to make a drawing of this array.

Including files

When writing an application, it is common to divide the code in several files in order to make it easier to read, update and reuse. PHP allows to link these files by including the code of one of them inside another during the execution time.

Including files

There are four instructions to include the code of a file inside another:

include *“file_name”*

include_once *“file_name”*

require *“file_name”*

require_once *“file_name”*

Including files

They work in a different way in case of failure:

STATEMENT	THE FILE DOESN'T EXIST	THE FILE HAS ALREADY BEEN USED
include	A warning is produced	The code is included again
include_once	A warning is produced	The code is not included
require	A fatal error is produced	The code is included again
require_once	A fatal error is produced	The code is not included

Including files

Write these documents:

- A file with a script containing the header of an HTML document.
- A second file with a script containing its footer.
- Another file with an HTML document which includes the previous ones.

Functions

PHP offers more than one thousand functions for very different purposes but, apart from them, it allows us to create our own functions.

- What functions have you used until now?
- How do we reference them?

Functions

A **function** is a block of instructions with a specific purpose that can be used many times during the execution of a program.

Functions are executed *only* in case they are called.

Functions

When calling a function any number of parameters can be passed to it in order to be used. After working with them, the function can also return a result.

Functions

To create a function, we will write:

```
function function_name(argument1, argument2, …)
 {
    instructions that will be executed
 }
```

Functions

The name of a function is made up by the developer following the same rules as for variables. The only difference between them is that function names are not case sensitive.

In order to make the code easy to understand it is advisable to give every function a **name that reflects the task it does**.

Functions

TRY IT!

```
<?php
    function greeting() {
        echo “Nice to meet you, “;}

    $name1 = “Daniel”;
    echo “<div>”;
    greeting();
    echo $name1 , “</div>”;

    $name2 = “Patricia”;
    echo “<div>”;
    greeting();
    echo $name2 , “</div>”;
?>
```

What is the output of this code?

Functions

Arguments are written in these ways:

PLACE	HOW TO WRITE THEM
Function declaration	Beginning with $ (like variables) and separated by comma.
Function calling	Specific values we need to send to the function separated by comma and in the same order as in the declaration

Functions

Write a script that contains a function which gets the pieces of data necessary and produce phrases like this:

Lia was born in Cork in 2004

Functions

It is possible to assign a default value to one or more function arguments. In this case, these will be the values that the function will use if it is called without them.

Functions

TRY IT!

```
<?php
    function greeting($name = “guy”) {
        echo “<div>Nice to meet you, “, $name, <div>;}

    greeting(“Daniel”);
    greeting(“Patricia”);
    greeting();
?>
```

What is the output of this code?

Functions

When writing a function, it is also possible to point out the type of data of each argument, like in this example:

```
function message(int $a, float $b, string $c)
{
    instructions that will be executed
}
```

Functions

In the previous case, if the types of the values used when calling the function are different from those written in the declaration, they will be transformed in order to avoid provoking errors.

Functions

TRY IT!

```
<?php
    function sum(int $a, int $b) {
        echo $a + $b;
    }

    sum(10,15.52);
?>
```

What happens when running this code?

Functions

These conversions of types of data can cause errors in results so, in order to avoid them, we can add this sentence as the first line of the code:

declare(strict_types=1);

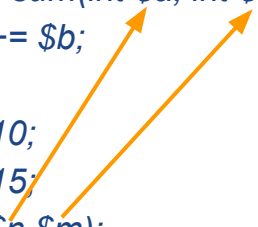
Add it to the previous script and explain what happens when running the code.

Functions

Until now, we have **passed arguments by value**, that is to say, the values in the call of a function are copied in the arguments of its declaration.

Functions

```
<?php
    function sum(int $a, int $b) {
        $a += $b;
    }
    $n = 10;
    $m =15;
    sum($n,$m);
?>
```



- What are the values of $n, $m, $a and $b when calling the function?
- What are their values after running it?

Functions

It is also possible to **pass arguments by reference**, in which a function does not receive values in its call but the memory positions where they are stored.

Functions

In this case, the function can change the values of the variables that have been sent in its calling.

To point out that an argument is passed by reference, the character **&** is written before it.

Functions

TRY IT!

```
<?php
    function triple(&$number) {
          $number *= 3;
        }

    $num = 2;
    triple($num);
    echo $num;
?>
```

What is the output of this script?

Functions

It is common that functions return the result of the operations they perform. For this purpose they must include the word ***return*** followed by the expression that they deliver.

Functions

Modify the previous code so that the function delivers the result to the place of the script from which it was called to carry it to the output document.

Functions

Similarly to arguments, it is possible to specify the type of data for the value return by a function. In this case, at the end of its header, we will add the semicolon character followed by the type.

Functions

TRY IT!

```
<?php
    function triple($number):int {
          $number *= 3;
          return $number;
        }

    $num = 2.75;
    triple($num);
    echo $num;
?>
```

- What is the output of this script?
- Write ***declare(strict_types=1)*** at the beginning of the script and explain the new output.

Functions

Write a script that sets the values of the base and the height of a rectangle and uses three functions to:

- Return its area.
- Return its perimeter.
- Interchange the values of the base and the height.

Arrays

An array is a variable that contains several values at the same time.

In order to use one of them, we will write the name of the array followed by the value of the index inside square brackets.

Arrays

To create an array, we use the function *array()* with its values as parameters.

It is possible to to know the number of elements in an array, by using the function *count(array_name)*.

Arrays

Explain the meaning of this sentence:

$drinks = array(“Coffee”,”Tea”,”Water”,”Beer”,”Wine”,”Soft drink”);

Arrays

Use the code in slide 46 and add this message:

“Choose one of these XXX drinks:”

where *XXX* will be the number of drinks in the array.

Arrays

In an **indexed array** the index has values between 0 and its number of elements minus one.

Arrays

VALUE	INDEX	REFERENCE
“Coffee”	0	*$drinks[0]*
“Tea”	1	*$drinks[1]*
“Water”	2	*$drinks[2]*
“Beer”	3	*$drinks[3]*
“Wine”	4	*$drinks[4]*
“Soft drink”	5	*$drinks[5]*

Arrays

It is also possible to create the same array writing this code:

```
$drinks[0] = “Coffee”;
$drinks[1] = “Tea”;
$drinks[2] = “Water”;
$drinks[3] = “Beer”;
…
```

Arrays

Write a script that creates an array containing the names of the modules you are studying this year and carries this content to the output document:

1. The title *“Modules of the 2º year”*.
2. The list of modules (using *foreach*).

Arrays

In an **associative array** the developer makes up the values of the index, creating pairs index-value with a logical connection.

Arrays

VALUE	INDEX	REFERENCE
28	V210	*$computers[V210]*
30	V211	*$computers[V211]*
29	V212	*$computers[V212]*
28	V213	*$computers[V213]*

Write two different codes to create this array.

Arrays

Write a script to create an associative array that stores the precipitation (litre per square meter) of each day of this week in Granada and carries the total amount to the output document.

Arrays

Until now we have used arrays with one index, however it is possible to add more indexes in order to create and manage **multidimensional arrays**.

In these arrays, elements contain other arrays.

Arrays

Bidimensional arrays use two indexes to reference a value. To create them, we build an array of arrays.

Arrays

Have a look at this example:

DAY	MINIMUM TEMPERATURE	MAXIMUM TEMPERATURE
1	*12*	*26*
2	*11*	*24*
3	*15*	*24*
…	*…*	*…*

Arrays

The array can be created in this way:

```
$temperatures = array (
    array(1,12,26),
    array(2,11,24),
    array(3,15,24),
      …
);
```

Arrays

To reference an element of a bidimensional array it is necessary to write the value of its two indexes.

Example: To get the maximum temperature of the 2nd day, we will write:

$temperaturas[1][2]

Web Application Implementation

TRY IT!

Arrays

```
<?php
    $temperatures = array (
                    array(1,12,26),
                    array(2,11,24),
                    array(3,15,24)
                );
                echo "<table><caption>Table of temperatures</caption>";
                echo "<tr><th>Day</th><th>Minimum</th><th>Maximum</th></tr>";
                for ($day = 0; $day < 3; $day++) {
                    echo "<tr>";
                    for ($col = 0; $col < 3; $col++) {
                            echo "<td>", $temperatures[$day][$col],"</td>";
                     }
                     echo "</tr>";
                }
    echo "</table>";
?>
```

Arrays

Write a script that uses two 5 x 5 matrices and two functions to create:

1. The transposed matrix.
2. The sum of both matrices.

Arrays

What control statements do we need to go through a bidimensional array?

Arrays

Use the code in slide 96 to get the last day with the highest maximum temperature of the month and carry it to the output document preceded by a message.

Arrays

Write a script that stores the audience in each of the three sessions of the four films that a cinema showed last Saturday and report the number of spectators in each session and in each film.

Arrays

- What is a tridimensional array?
- How can we build it?
- How can we reference each of its element?
- How can we can we go through it?

Arrays

Write a script that delivers the total audience of the former cinema in the last weekend.

Strings

A **string** is sequence of characters that can contain letters, numbers, spaces, punctuation characters and symbols.

Each character in a string can be referenced in the same way as a value in an array.

Strings

$greeting = “Hello”;

CHARACTER	INDEX	REFERENCE
“H”	0	*$greeting[0]*
“e”	1	*$greeting[1]*
“l”	2	*$greeting[2]*
“l”	3	*$greeting[3]*
“o”	4	*$greeting[4]*

Strings

We studied an operator for characters. What is it? What do we use it for?

Strings

FUNCTION	OUTPUT	EXAMPLE
strlen()	Number of characters in the string.	*strlen("See you soon!")* Output: 13
str_word_count()	Number of words in the string.	*str_word_count("See you soon!")* Output: 3
strrev()	Reverse string.	*strrev("See you soon!")* Output: !noos uoy eeS

Strings

FUNCTION	OUTPUT	EXAMPLE
strpos()	Position of a substring inside a string. If it is not contained, it returns FALSE.	*strpos("See you soon!","you")* Output: 4
substr()	Substring that begins in a position and has a number of characters.	*substr("See you soon!",8,4)* Output: "soon"
trim()	String without some characters take out from the beginning and the end.	*trim("See you soon!","!noS")* Output: "ee you s"

Strings

FUNCTION	OUTPUT	EXAMPLE
str_replace()	String with a substring changed.	*str_replace(“See”,”Phone”,“See you soon!”)* Output: “Phone you soon!”
explode()	Array whose elements are substrings separated by a character.	*explode(“ ”,“See you soon!”)* Output: Array(“See”,”you”,”soon!”)
implode() *join()*	String built from the elements of an array joined by a character.	*$a = Array(“See”,”you”,”soon!”);* *implode(“ “,$a)* Output: “See you soon!”

Strings

Write a script that points out if a string is a palindrome. A palindrome is a sequence of characters that reads the same backwards as forwards, such as *madam* or *racecar.*

Strings

Write a script that points out the number of times that a string is contained inside another string.

Strings

Write a script that points out the number of times that a string is contained inside another string.

Exercise

A teacher recorded the marks (integers between 1 and 10) got by her 20 students (named like this: Fernández Gil, María) in a project, a class activity and an exam. Write a script to deliver a table to show the students' term marks bearing in mind that:

- The students' names will appear like this: María Fernández Gil. (Use a function to get this change)
- The class activity was 15% of the final mark, the project was 35% and the exam was 50%.