

Certificado de profesionalidad IFCD0210

Introducción

Los ordenadores requieren de programas o aplicaciones.

Es muy complicado construir programas en el lenguaje nativo de los ordenadores.

Se han creado unos lenguajes de programación que facilita esta tarea.

Lenguaje de programación es un conjunto de símbolos y reglas que combinándolos se usan para crear programas

Introducción

Los lenguajes de programación son un tipo de lenguaje:

- Léxico: símbolos permitidos y vocabulario
- Sintaxis: reglas que indican como se construye el lenguaje
- Semántica: reglas que permiten saber el significado de las expresiones que se construyen.

Introducción

Lenguaje máquina

Lenguaje ensamblador

Lenguaje de alto nivel

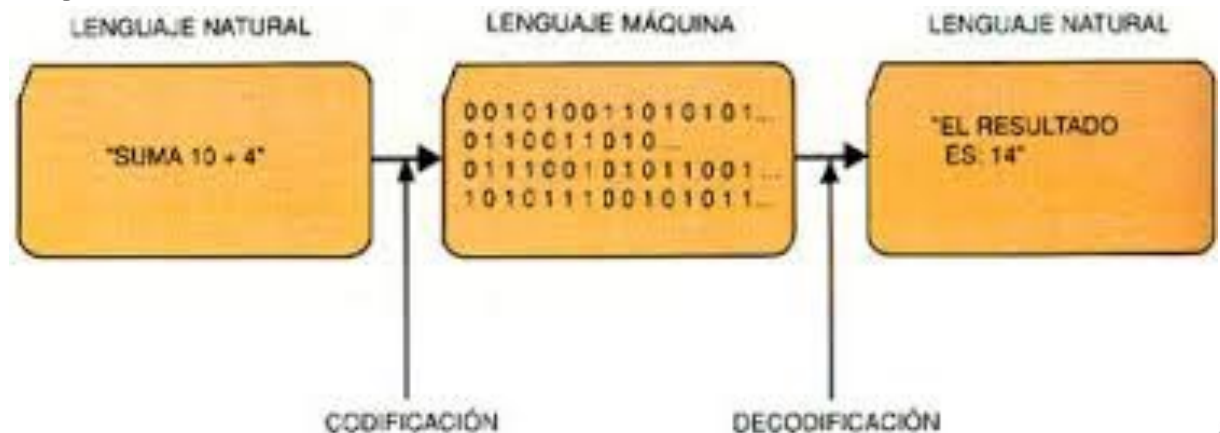
Traductores: compiladores e interpretes

Introducción - Lenguaje máquina

- Es el lenguaje de los ordenadores.
- Estructura adaptada a los ordenadores.
- Programación difícil y requiere conocer la estructura física del ordenador.
- Se obtienen programas muy eficientes.

Características:

- Instrucciones expresadas en binario, hexadecimal o octal.
- Datos referenciados por las direcciones de memoria donde están.
- Instrucciones que hacen operaciones simples
- Instrucciones rígidas



Introducción - Lenguaje ensamblador

- Intento de hacer más humano el lenguaje de los ordenadores.
- Se traduce al lenguaje máquina con el ensamblador.
- Los programadores usan este lenguaje para realizar programas muy eficientes.
- El lenguaje continua estando ligado al procesador.

Características:

Se usa una notación simbólica para expresar los códigos de operación.

Permite un direccionamiento simbólico de variables.



Introducción - Lenguaje de alto nivel

- Se independiza el lenguaje de programación de la máquina.
- Utilizan expresiones parecidas a los humanos.
- Necesitan de un compilador o interprete.

Características:

- Independientes de la arquitectura física de la máquina.
- Requiere de un programa que traduce a LM
- Una sentencia de alto nivel da lugar a varias instrucciones de lenguaje máquina.
- Hay instrucciones potentes de uso frecuente: funciones matemáticas, E/S, tratamiento de variables, etc..
- Son menos eficientes

ejemplo C: Hola Mundo!

```
#include <stdio.h>

int main()
{
    printf("Hola Mundo!\n");
    return 0;
}
```

Introducción

Compiladores e intérpretes

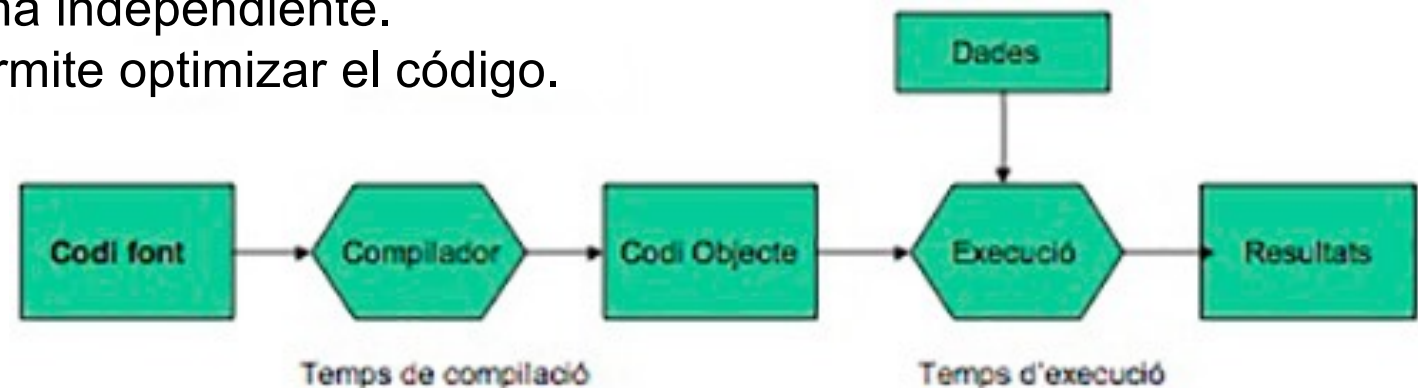
Traductor es un programa que toma como entrada un programa escrito en un lenguaje de alto nivel, llamado código fuente, y proporciona como salida otro programa semánticamente igual llamado código objeto.

Hay de dos tipos:

- Compiladores
- Intérpretes

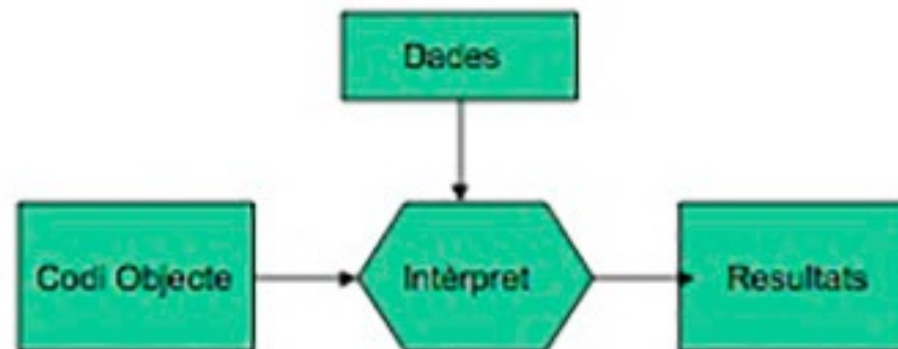
Introducción Compiladores

- Traduce el código fuente a un código objeto escrito en LM
- Durante el proceso de compilación se avisa al programador de los errores detectados y no se genera el código objeto hasta que sea un código fuente correcto.
- Los códigos fuente y objeto se guardan en dos ficheros de forma independiente.
- Permite optimizar el código.



Introducción Intérpretes

- Permite que el código fuente se traduzca y ejecute sentencia a sentencia.
- Se usa directamente el código fuente para ejecutar programas.
- Cada vez que se ejecuta el programa se ejecuta el código.
- Es más lento y la optimización del código es la del propio código.



¿Qué es PHP?

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP es código interpretado que se ejecuta en el servidor y que puede contener HTML.

Mi primer código en PHP

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

Mi primer código en PHP

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>
    <HR>
    <?php echo 'Otro escrito'; ?>
  </body>
</html>
```

Mi primer código en PHP

```
<?php
echo "<html>
    <head>";
echo "        <title>Ejemplo</title>";
echo "    </head>";
echo "    <body>";
echo "¡Hola, soy un script de PHP!";
?>

        <HR>

<?php
echo " </body>";
echo "</html>";
?>
```

Mi primer código en PHP

```
<?php
echo "<html>";
echo "    <head>";
echo "        <title>Ejemplo</title>";
echo "    </head>";
echo "    <body>";
echo "¡Hola, soy un script de PHP!";
// echo "Esto no sale en la web";
?>

    <HR>

<?php
echo " </body>";
echo "</html>";
?>
```

Mi primer código en PHP

```
<?php
echo "<html>";
echo "    <head>";
echo "        <title>Ejemplo</title>";
echo "    </head>";
echo "    <body>";
/* echo "¡Hola, soy un script de PHP!";
echo "Esto no sale en la web"; */
?>

    <HR>

<?php
echo " </body>";
echo "</html>";
?>
```


Ejemplo formulario

formulario.html

```
<form action="procesar.php" method="post">  
  <p>Su nombre: <input type="text" name="nombre" /></p>  
  <p>Su edad: <input type="text" name="edad" /></p>  
  <p><input type="submit" /></p>  
</form>
```

Ejemplo formulario

procesar.php

Hola <?php echo \$_POST['nombre']; ?>.
Usted tiene <?php echo \$_POST['edad']; ?> años.

Ejemplo formulario

procesar.php

```
Hola <?php echo htmlspecialchars($_POST['nombre']); ?>.  
Usted tiene <?php echo (int)$_POST['edad']; ?> años.
```

Ejemplo formulario

procesar.php

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <?php $nombre=htmlspecialchars($_POST['nombre']);?>
    Hola!!<BR>
    <?php
      echo "Su nombre es ";
      echo $nombre;

    ?>.
    Usted tiene <?php echo (int)$_POST['edad']; ?> años.
  </body>
</html>
```

Variables simples

variables.php

```
<?php
$nombre = "Borja";
$edad = 23;
$variable1 = "Mis \"super\" amigos";
$variable2 = 'Mis "super" amigos';
echo 'Mi nombre es: '.$nombre."<BR>";
echo $edad."<BR>";
echo $variable1."<BR>";
echo $variable2."<BR>";
?>
```

ACTIVIDAD 2: ENVIAR MAIL

```
<?php $to='Borja.mulleras@aspasia.es'  
$message = 'Nombre: '.$_POST['nombre'].' Edad:  
'.$_POST['edad'].' Este es un';  
$headers="";
```

```
mail($to,'Recibido por formulario '.$message,$headers);
```

\$to: los destinatarios separados por comas
\$subject: asunto del e-mail
\$message: contenido del email
\$headers: contiene las cabeceras del email

PHP2

Realizar un PHP que permita calcular el área de un rectángulo. Leer la Base y la Altura.

$$\text{AREA} = \text{BASE} * \text{ALTURA}$$

PHP3

Desarrollar un algoritmo que permita leer dos valores distintos y determinar cual de los dos valores es el mayor. Escribir el resultado.

Realizar un algoritmo, además, que sume los dos números.

Si Entonces Sino

SI te llamas Borja ENTONCES

Escribe("Saluda")

SINO

Escribe("No eres Borja")

FSI

```
<?php
    $nombre="Borja";
    if ($nombre=="Borja")
    {
        echo "Saluda";
    }
    else
    {
        echo "No te llamas Borja";
    }
?>
```

```
<?php
    $nombre="Borja1";
    if ($nombre=="Borja"):
        echo "Saluda";
    else:
        echo "No te llamas Borja";
    endif;
?>
```

Si Entonces Sino Si

```
SI (a > b) ENTONCES
    Escribe('a es mayor que b')
SINO SI (a = b) ENTONCES
    Escribe('a es igual que b')
SINO
    Escribe('a es menor que b')
FSI
```

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

SEGÚN - CASO

```
nombre="Borja"  
SEGÜN nombre HACER  
CASO "Borja":  
    Escribir("Soy yo")  
    Salir  
CASO "Javier":  
    Escribir("Es mi  
hermano")  
    Salir  
SINO  
    Escribir("Es otro")
```

FSEGUN

```
<?php  
$nombre="Javier1";  
switch ($nombre)  
{  
    case "Borja": echo "Soy yo";  
                break;  
    case "Javier": echo "Es mi hermano";  
                break;  
    Default:  
                echo "Es otro";  
}  
?>
```

PHP4

Desarrolle un algoritmo que permita convertir calificaciones numéricas a letras según la siguiente tabla de conversión:

NÚMERO	LETRA
<i>19-20</i>	<i>A</i>
<i>16-18</i>	<i>B</i>
<i>12-15</i>	<i>C</i>
<i>9-11</i>	<i>D</i>
<i>0-8</i>	<i>E</i>

Actividad 3

Desarrolle un algoritmo para determinar el pago de entradas de espectáculo en función del número que compren.

NÚMERO	DESCUENTO
<i>1</i>	-
<i>2</i>	<i>10%</i>
<i>3</i>	<i>15%</i>
<i>4</i>	<i>20%</i>
<i>5 o más</i>	<i>25%</i>

Variables asignación por valor

Las variables por defecto siempre se asignan por valor, esto significa que cuando se realiza una asignación se hace una copia del valor. De forma que, si se modifica la copia, la otra variable queda invariable.

```
<?php
    $_1valor=3;
    $copia=$_1valor;
    $_1valor=$_1valor+1;
    echo $copia;
    echo $_1valor;
?>
```

Variables asignación por referencia

Cuando se realiza la asignación simplemente se hace una copia de la dirección de memoria, es decir, que cualquiera de las 2 que modifica, modifica la otra.

Sólo es válido la asignación de nombres de variables.

```
<?php
    $_1valor=3;
    $copia=&$_1valor;
    $_1valor=$_1valor+1;
    echo $copia;
    echo $_1valor;
?>
```

Notas sobre variables

Las variables se inicializan en función del contexto donde se encuentran

4 tipos de variables:

Las booleanas se inicializan a FALSE

Los números enteros como 0

Los números flotantes como 0

Las cadenas vacías

Los arrays como un array vacío.

La función `var_dump($variable)` nos indica el tipo de variable

Notas sobre variables

```
$int_indefinido += 25; // 0 + 25
```

```
$int_indefinido = $int_indefinido + 25;
```

Si utilizamos una que no se ha inicializado, el valor es null

Podemos borrar una variable con unset();

Notas sobre variables

isset(): Determina si una variable ha sido definida y no es nula. Devuelve *FALSE* en caso que le pasemos una variable sin definir, una variable definida pero sin valor o con el valor puesto a *null*, y en el resto de casos devuelve *TRUE*.

empty(): Determina **si la variable tiene un valor vacío**, por llamarlo de alguna forma. En este caso devuelve *TRUE* en muchos supuestos: **una cadena vacía** (es decir ""), **un número 0** (sea tanto un entero, un float o una cadena con el número cero tal que "0"), **una variable con el valor *FALSE***, una **variable con el valor *NULL***, una **variable definida pero sin valor** y un **array vacío**. Al contrario que **isset()**, no puede evaluar variables que no hayan sido definidas.

is_null(): En cierto modo es la complementaria a **isset()**, ya que devuelve *TRUE* en caso de que la variable sea *NULL* o que sea **una variable definida pero sin valor**, y en el resto de los casos *FALSE*. La diferencia es que **no puede evaluar variables que no estén definidas**, provocará un error.

Notas sobre variables

```
<?php  
$a = 1;  
if (empty($a)) {  
    echo 'empty';  
}  
  
if (isset($a)) {  
    echo 'isset';  
}  
?>
```

```
<?php  
$a = 0;  
if (empty($a)) {  
    echo 'empty';  
}  
  
if (isset($a)) {  
    echo 'isset';  
}  
?>
```

Notas sobre variables

El valor especial **NULL** representa una variable sin valor. **NULL** es el único valor posible del tipo [null](#).

Una variable es considerada [null](#) si:

- se le ha asignado la constante **NULL**.
- no se le ha asignado un valor todavía.
- se ha destruido con [unset\(\)](#).

Notas sobre variables

```
<?php  
$primero ? $segundo : $tercero  
?>
```

Si el valor de la primera subexpresión es **TRUE** (no es cero), se evalúa la segunda subexpresión, y ése será el resultado de la expresión condicional. Si no, se evalúa la tercera subexpresión, y ése será el valor.

```
$pagina1= !empty($_POST["pagina"]) ? $_POST["pagina"] : 1;
```

Incremento

```
$numero = 5;  
++$numero;  
echo $numero; // numero = 6
```



```
var $numero = 5;  
$numero = $numero + 1;  
echo $numero; // numero = 6
```

Decremento

```
$numero = 5;  
--$numero;  
echo $numero; // numero = 4
```



```
$numero = 5;  
$numero = $numero - 1;  
echo $numero; // numero = 4
```

Incremento / Decremento

```
$numero1 = 5;  
$numero2 = 2;
```

5

```
$numero3 = $numero1++ + $numero2;  
echo $numero3;  
echo $numero1;  
// $numero3 = 7, $numero1 = 6
```



```
$numero1 = 5;  
$numero2 = 2;  
$numero3 = ++$numero1 + $numero2;  
echo $numero3;  
echo $numero1;  
// numero3 = 8, numero1 = 6
```

NO ES LO MISMO ++6 QUE 6++

Actividad 4

A) Haz una aplicación que en función de la edad devuelva que es una persona joven si tiene menos de 30 años y que es maduro si es superior o igual.

B) Haz una aplicación que en función del número del mes entrado con un formulario devuelva el nombre del mes.

ARRAY

Un [array](#) en PHP es en realidad un mapa ordenado. Un mapa es un tipo de datos que asocia *valores* con *claves*. Este tipo se optimiza para varios usos diferentes; se puede emplear como un array, lista (vector), tabla asociativa (tabla hash - una implementación de un mapa), diccionario, colección, pila, cola, y posiblemente más. Ya que los valores de un [array](#) pueden ser otros [arrays](#), también son posibles árboles y [arrays](#) multidimensionales.

```
array [  
    clave => valor,  
    clave2 => valor2,  
    clave3 => valor3,  
    ...  
]
```

ARRAYS NÚMERICOS

Clave	Valor
0	Pedro
1	Juan
2	Ángel
3	Rubén
4	Ana

Clave	Valor
0	10
1	5
2	15
3	7
4	10

```
<?php
    $tabla = array('Pedro','Juan','Ángel','Rubén', 'Ana');
    $tablaNum = array(10,5,15,7,10);
?>
```

ARRAYS

Clave	Valor
0	Pedro
1	Juan
2	Ángel
3	Rubén
4	Ana

```
<?php
    $tabla[0] = 'Pedro';
    $tabla[1] = 'Juan';
    $tabla[2] = 'Ángel';
    $tabla[3] = 'Rubén';
    $tabla[4] = 'Ana';
?>
```

```
<?php
    $tabla = array();
    $tabla[] = 'Pedro';
    $tabla[] = 'Juan';
    $tabla[] = 'Ángel';
    $tabla[] = 'Rubén';
    $tabla[] = 'Ana';
?>
```

```
<?php
    $tabla = array['Pedro','Juan','Ángel','Rubén','Ana'];
?>
```

ARRAYS

Clave	Valor
A	Pedro
B	Juan
C	Ángel
D	Rubén
E	Ana

```
<?php
    $tabla = array (
        "A" => "Pedro",
        "B" => "Juan",
        "C" => "Ángel",
        "D" => "Rubén",
        "E" => "Ana"
    );?>
```

```
<?php
    $tabla = [
        "A" => "Pedro",
        "B" => "Juan",
        "C" => "Ángel",
        "D" => "Rubén",
        "E" => "Ana",
    ];?>
```

```
print_r($tabla);
```

Actividad 5

El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.

El array de letras es:

```
$letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'];
```

Por tanto, si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A.

Actividad 5

Con estos datos, elaborar un pequeño script que:

Almacene en una variable el número de DNI indicado por el usuario y en otra variable la letra del DNI que se ha indicado.

En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 999999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.

Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente. ($\$resto = \$operador1 \% \$operador2;$)

Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

PARA

```
PARA i=1 HASTA 10 HACER  
    Escribir("Hola, esta es la vez", i)  
fPARA
```

```
<?php  
    for ($i = 1; $i <= 10; $i++)  
    {  
        echo "Hola, esta es la vez ".$i."<BR>";  
    }  
  
?>
```

PHP5

El factorial de un número entero n es una operación matemática que consiste en multiplicar todos los factores $n \times (n-1) \times (n-2) \times \dots \times 1$. Así, el factorial de 5 (escrito como $5!$) es igual a: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Utilizando la estructura for, crear un script que calcule el factorial de un número entero.

WHILE

```
i=0;  
MIENTRAS i<=10 HACER  
    Escribir("Hola, esta es la vez", i)  
    i=i+1  
fMIENTRAS
```

```
<?php  
    $i = 1;  
    while ($i <= 10):  
        echo ("Hola, esta es la vez ".$i."<BR>");  
        $i++;  
    endwhile;  
?>
```

```
<?php  
    $i = 1;  
    while ($i <= 10)  
    {  
        echo ("Hola, esta es  
la vez ".$i."<BR>");  
        $i++;  
    }  
?>
```

DO WHILE

```
i=0  
HACER / REPETIR  
    Escribir("Hola, esta es la vez", i)  
    i=i+1  
MIENTRAS i<10
```

```
<?php  
$i = 0;  
do {  
    echo "Hola, esta es la vez".$i;  
    $i++;  
} while ($i < 10);  
?>
```

DO WHILE

```
<?php  
$i = 0;  
do {  
    echo $i;  
} while ($i > 0);  
?>
```

PHP6

Haz un algoritmo que nos pida 6 datos y
luego los muestre en orden contrario al que se
han introducido

ACTIVIDAD 6

Desarrolle un algoritmo que lea cuatro números distintos y determine cual de los cuatro es mayor.

Pista: Utilizar una tabla

TABLAS ASOCIATIVA

La clave de la tabla es libre

Clave	Valor
Nombre	Borja
Apellido	Mulleras
Curso	Webs
Teléfono	666 666 666

```
<?php
    $tabla = array();
    $tabla['Nombre'] = 'Borja';
    $tabla['Apellido'] = 'Mulleras';
    $tabla['Curso'] = 'Webs';
    $tabla['Telefono'] = '666 666 666';
?>
```

```
<?php
    $tabla = array(
        'Nombre'=>'Borja',
        'Apellido'=>'Mulleras',
        'Curso'=>'Webs',
        'Teléfono'=>'666 666 666'
    );
?>
```

Tamaño de las tablas

Sizeof()

Clave	Valor
0	Pedro
1	Juan
2	Ángel
3	Rubén
4	Ana

```
<?php
    $tabla = array('Pedro','Juan','Ángel','Rubén', 'Ana');
    for ($i = 0; $i <= sizeof($tabla); $i++)
    {
        echo ($tabla[$i]."<BR>");
    }
?>
```

Lo mismo con count()

Recorrido de tablas - foreach

Clave	Valor
0	Pedro
1	Juan
2	Ángel
3	Rubén
4	Ana

```
<?php
    $tabla = array('Pedro','Juan','Ángel','Rubén', 'Ana');
    foreach ($tabla as $val)
    {
        echo ($val."<BR>");
    }
?>
```


Recorrido de tablas - foreach

Clave	Valor
Nombre	Borja
Apellido	Mulleras
Curso	Webs
Teléfono	666 666 666

```
<?php
    $tabla = array(
        'Nombre'=>'Borja',
        'Apellido'=>'Mulleras',
        'Curso'=>'Webs',
        'Teléfono'=>'666 666 666'
    );
    foreach ($tabla as $clave => $val)
    {
        echo "Clave: ".$clave." Valor: ".$val."<BR>";
    }

?>
```

Recorrido de tablas - foreach

Clave	Valor
Nombre	Borja
Apellido	Mulleras
Curso	Webs
Teléfono	666 666 666

```
<?php
    $tabla = array(
        'Nombre'=>'Borja',
        'Apellido'=>'Mulleras',
        'Curso'=>'Webs',
        'Teléfono'=>'666 666 666'
    );

    print_r($tabla);

?>
```

ACTIVIDAD 7

Crear un formulario con campos multiples y enviar el resultado por mail.

Pista y ejemplo:

<https://es.stackoverflow.com/questions/14295/enviar-un-arreglo-asociativo-multidimensional-mediante-el-uso-de-formularios>

Funciones

Una función es un conjunto de instrucciones a la que podemos recurrir siempre que queramos. Éstas pueden recibir parámetros y realizar todo tipo de tareas, ya sean complejas o sencillas.

- PHP tiene una extensa biblioteca de funciones propias: `sort()`, `count()`, `is_array()`, `isset()`, etc...
- También podemos tener nuestras propias funciones

Funciones de tablas

`in_array($valor_búsqueda,$tabla)`

Buscamos el \$valor_búsqueda en la \$tabla

Ej: `in_array("Borja",$tabla)`

`sort($tabla)`

Ordena los valores de la tabla de menor a mayor.

`asort()`, `rsort()`, `arsort()`, `ksort()`, `krsort()` (pag. 59)

`array_push()`

Permite añadir varios elementos a la vez al final de la tabla.

Funciones de cadenas de caracteres

`$cadena="Borja";`

`strlen($cadena)`

Devuelve la longitud de la cadena. En este caso 5.

`substr($cadena,2,1) - substr($cadena,$posicion_inicio, longitud)`

Te devuelve el trozo de \$cadena que empieza en la posición 2 y que ocupa un carácter. En este caso r.

`str_replace($cadena_buscada,$cadena_que_sustituye,$cadena)`

Sustituye un trozo de cadena por otro.

Ejemplo: `str_replace('rja','ta',$cadena)`

Funciones

```
<?php  
function escribe_encabezado($var1) {  
    $encabezado=$var1;  
    echo $encabezado;  
}  
?>
```

La llamada a la función desde cualquiera de nuestras páginas sería así:

```
<?php  
$mititulo="Mi Proyecto";  
escribe_encabezado($mititulo);  
?>
```

Funciones

```
<?php
function calcular_precio_iva ($precio, $iva) {
    return $precio*$iva;
}
?>
```

La llamada a la función desde cualquiera de nuestras páginas sería así:

```
<?php
$pvp_sin_iva=20;
$IVA=1.21;
$pvp_con_iva=calcular_precio_iva($pvp_sin_iva,$IVA);
echo $pvp_con_iva;
?>
```


Funciones

```
<?php
function calcular_precio_iva ($precio, $iva)
{
    return $precio*$iva;
}
?>
```

```
<?php
$pvp_sin_iva=20;
$IIVA=1.21;
$pvp_con_iva1=calcular_precio_iva($pvp_sin_iva,$IIVA);
echo $pvp_con_iva1;
echo calcular_precio_iva(100,$IIVA);
If (es_producto_iva($producto)) {

}
?>
```

Funciones

```
<?php
function números_pequeños()
{
    return array (0, 1, array(2 ,3));
}
list($cero, $uno, list( $dos, $tres)) = números_pequeños();
echo $cero;
echo $uno;
echo $dos;
?>
```

<https://www.php.net/manual/es/functions.user-defined.php>

Funciones

```
<?php
$a = 1; /* ámbito global */

function test()
{
    echo $a; /* referencia a una variable del ámbito local */
}

test();
?>
```

No imprimirá nada, porque \$a es de ámbito local.

Funciones

```
<?php  
$a = 1;  
$b = 2;
```

```
function Suma()  
{  
    global $a, $b;  
  
    $b = $a + $b;  
}
```

```
Suma();  
echo $b;  
?>
```

Global hace que el valor de las variables \$a y \$b sea global. Es decir, en este caso \$b nos imprimirá 3

Funciones

```
<?php
function test()
{
    $a = 0;
    echo $a;
    $a++;
}
```

```
test(); //0
test(); //0
?>
```

```
<?php
function test()
{
    static $a = 0;
    echo $a;
    $a++;
}
test(); //0 $a=1
test(); //1 $a=2
?>
```

Static hace que solo se ejecute la inicialización la primera vez que se ejecuta la función.

Librerías

```
test.php
<?php

echo "Una $fruta $color"; // Una

include 'vars.php';

echo "Una
$fruta $color"; // Una manzana verde

?>
```

```
vars.php
<?php

$color = 'verde';
$fruta = 'manzana';

?>
```

Librerías

Include

require

include_once

require_once

Composer

Composer es un [sistema de gestión de paquetes](#) para programar en [PHP](#) el cual provee los formatos estándar necesarios para manejar dependencias y [librerías](#) de PHP.


DEDE ESTAR INSTALADO EN EL SERVIDOR

ACCESO SSH

<https://es.wikipedia.org/wiki/Composer>

Composer

aspasia11.md360.es ...

Website at  <httpdocs/> IP address: 51.75.196.89 System user: **aspasia11**

 [Hosting Settings](#)  [Open](#)  [Preview](#) [Suspend](#) [Disable](#) [Description](#)



Praise The Sun

Log In

Check Security

Plugins

Themes

Clone

Copy Data

⚠ Plugin updates are available. [Details](#)

⚠ Theme updates are available. [Details](#)



SEO Toolkit

Scan

Add keywords

Scan is not performed yet. Click "Scan" to crawl your site.

⬆ SHOW LESS



[Web Hosting Access](#)



[FTP Access](#)



[Hosting Settings](#)

Composer

System user

Here you can specify an operating system user for managing files and folders within the subscription by FTP or with Fi over SSH (on Linux) or RDP (on Windows) if the corresponding permissions are granted.

Username *

aspasia11

Password

Aspasia20

Medium

(?)

Generate

Hide

Confirm password

Access to the server over SSH

Forbidden

✓ /bin/sh

/bin/dash

/bin/bash

/bin/rbash

/usr/bin/screen

/bin/bash (chrooted)

/usr/bin/tmux

SSH with system user's credentials.

* Required fields

ly

Cancel

Composer

En MAC usamos Terminal
ssh aspasia11@aspasia11.md360.es

[illegible]

Para windows

<https://docs.ovh.com/es/dedicated/introduccion-ssh/>

Composer

Instalamos el paquete DOMPDPF
composer require dompdf/dompdf

```
borja — ssh aspasia11@aspasia11.md360.es — 137x32

Last login: Fri Jul 24 16:50:42 2020 from 151.red-83-41-22.dynamicip.rima-tde.net
$ composer require dompdf/dompdf
Using version ^0.8.3 for dompdf/dompdf
./composer.json has been created
Running composer update dompdf/dompdf
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
- Locking dompdf/dompdf (v0.8.3)
- Locking phenx/php-font-lib (0.5.2)
- Locking phenx/php-svg-lib (v0.3.3)
- Locking sabberworm/php-css-parser (8.3.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
- Downloading sabberworm/php-css-parser (8.3.1)
- Downloading phenx/php-svg-lib (v0.3.3)
- Downloading phenx/php-font-lib (0.5.2)
- Downloading dompdf/dompdf (v0.8.3)
- Installing sabberworm/php-css-parser (8.3.1): Extracting archive
- Installing phenx/php-svg-lib (v0.3.3): Extracting archive
- Installing phenx/php-font-lib (0.5.2): Extracting archive
- Installing dompdf/dompdf (v0.8.3): Extracting archive
1 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
$
```

Composer

Borramos DOMPDF

`composer remove dompdf/dompdf`

```
borja — ssh aspasia11@aspasia11.md360.es — 137x32
$ composer remove dompdf/dompdf
./composer.json has been updated
Running composer update dompdf/dompdf
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 0 updates, 1 removal
- Removing dompdf/dompdf (v0.8.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 0 updates, 1 removal
- Removing dompdf/dompdf (v0.8.3)
Generating autoload files
$
```

¿Y si no tengo composer?

Bajar la librería de <http://eclecticgeek.com/dompdf-info/>

Subimos al servidor y extraemos la carpeta

```
<?php
require_once 'dompdf/autoload.inc.php'; // Se usa sin composer
require '../vendor/autoload.php'; // SE usa si composer

// reference the Dompdf namespace
use Dompdf\Dompdf;

// instantiate and use the dompdf class
$dompdf = new Dompdf();
$dompdf->loadHtml('hello world');

// (Optional) Setup the paper size and orientation
$dompdf->setPaper('A4', 'landscape');

// Render the HTML as PDF
$dompdf->render();

// Output the generated PDF to Browser
$dompdf->stream();
?>
```

PHPMailer

<https://github.com/PHPMailer/PHPMailer>

Subir un fichero

<https://code.tutsplus.com/es/tutorials/how-to-upload-a-file-in-php-with-example--cms-31763>

<https://aspasia11.md360.es/archivos>

Imagick

<https://www.php.net/manual/es/book.imagick.php>

`phpinfo();`

Ejemplo de uso:

https://aspasia11.md360.es/archivos/leer_miniaturas.php

Certificado de profesionalidad IFCD0210