

- Entrega 1

- Programa 1:
 - Tipos de datos
 - Funciones
 - Variable estáticas
- Programa 2: echo / print
- Programa 3: printf
 - Formato de enteros
 - %d
 - %b
 - %o
 - %x
 - %X
 - Formato de números con Punto Flotante
 - %f
 - %.2f
 - %e
 - %E
 - Formato de caracteres
 - %c
 - Alineación y relleno
 - %5d
- Programa 4:
- Programa 5: Cadenas de texto
- Programa 6: Comprobación y conversión de tipos
 - Desactivamos los warning
- Programa 7: Constantes
- Programa 8: Constantes
- Programa 9: Date
 - d
 - j
 - z
 - N
 - w
 - l
 - D
 - W

- `m`
 - `n`
 - `t`
 - `F`
 - `M`
 - `Y`
 - `y`
 - `L`
 - `h`
 - `H`
 - `g`
 - `G`
 - `a`
 - `A`
 - `r`
- Programa 10: variables superglobales vs constantes predefinidas

Entrega 1

Programa 1:

Tipos de datos

Hemos aprendido como crear un script sencillo, usando echo, html como h1, h2, saltos de línea.

```

<?php
//codigo php
echo "Programa 1";

echo "<h1> Programa 1</h1>";

echo "Esta es la primera línea.\n";
echo "Esta es la segunda línea.\n";
echo "Esta es la tercera línea.\r\n"; // Para mayor compatibilidad

echo "Esta es la primera línea.<br>";
echo "Esta es la segunda línea.<br>";

```

Además hemos aprendido variables básicas como numeros, booleanos, string, array, objetos, recursos y simbolo

```

echo "<h2>PHP Tipos de datos</h2>";
$numero = 42; //Tipo de dato entero
$precio = 19.99; //Tipo de dato float
$saludo = "Hola, mundo!"; //Tipo de dato string
$booleano = true; //Tipo de dato booleano
$colores = array("rojo", "verde", "azul"); //Tipo de dato array
$objeto =(object) [ `nombre` => `Juan` , `edad` => 26]
$nulo = null;
//$recurso = fopen("archivo.txt","r");
$simbolo = "@";

```

Vamos ahora a imprimir todos estos datos y lo hacemos de la siguiente forma

```

echo "<h2>Muestro los tipos de datos creados arriba </h2>";

echo "Tipo de dato entero: " . gettype($numero) . " con valor: $numero<br>";

echo "Tipo de dato flotante: " . gettype($precio) . " con valor: $precio<br>";

echo "Tipo de dato cadena: " . gettype($saludo) . " con valor: $saludo<br>";

echo "Tipo de dato booleano: " . gettype($booleano) . " con valor: $booleano<br>";

echo "Tipo de dato Array: " . gettype($colores) . " con valor: " . implode(", " , $colores) . "<br>";

echo "Tipo de dato objeto: " . gettype($objeto) . " con valor: " . $objeto->nombre . " , " . $objeto->edad . "<br>";

echo "Tipo de dato nulo: " . gettype($nulo) . " con valor: " . var_export($nulo, true) . "<br>";

//echo "Tipo de dato recurso: " . gettype($recurso) . "<br>";

echo "Tipo de dato simbolo: " . gettype($simbolo) . " con valor: $simbolo<br>";

echo "<hr>";

```

Funciones

Vamos a ver también las variables globales y las variables locales

```
echo "<h2>Funciones con variables globales</h2>";  
  
$variableGlobal = "Soy una variable global";  
$x = 10; //Ámbito global  
  
function probarAmbito (){  
    global $variableGlobal; //Acceder a la variable global  
    $variableLocal = "Yo soy una variable local";  
  
    echo $variableGlobal . "<br>"; //Accede a la variable global  
    echo $variableLocal . "<br>"; //Accede a la variable local  
}  
  
function miFuncion(){  
    global $x; //Hace $x accesible dentro de la función  
    echo $x;  
}  
  
miFuncion();  
probarAmbito();
```

```
echo "<h2>Ámbito local</h2>";  
  
function otraFuncion(){  
    $y = 5; //Ámbito local  
    echo $y;  
}  
  
otraFuncion(); //Imprime:5  
echo $y; //Error: $y no está definida fuera de la función |
```

Variable estáticas

```
echo "<h2>Variables estáticas</h2>";  
  
function Contador(){  
    static $contador = 0;  
    $contador++;  
    echo $contador;  
}  
  
Contador(); //Imprime 1  
Contador(); //Imprime 2  
Contador(); //Imprime 3
```

Programa 1

Programa 1

Esta es la primera línea. Esta es la segunda línea. Esta es la tercera línea. Esta es la primera línea.
Esta es la segunda línea.

PHP Tipos de datos

Muestro los tipos de datos creados arriba

Tipo de dato entero: integer con valor: 42
Tipo de dato flotante: double con valor: 19.99
Tipo de dato cadena: string con valor: Hola, mundo!
Tipo de dato booleano: boolean con valor: 1
Tipo de dato Array: array con valor: rojo,verde,azul
Tipo de dato nulo: NULL con valor: NULL
Tipo de dato simbolo: string con valor: @

Funciones con variables globales

10Soy una variable global
Yo soy una variable local

Ámbito local

5
Warning: Undefined variable \$y in C:\xampp\htdocs\DWES\Dwes_Maria\UD2PHP\Entrega1\Programa1.php on line 83

Variables estáticas

123

Programa 2: echo / print

Mostrando por pantalla: Hemos completado el siguiente código para que funcione correctamente. Añadimos un valor a la variable, en esta caso será mi nombre. Luego para poder imprimirlo llamamos a la variable

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ejemplo PHP con variables</title>
6  </head>
7  <body>
8      <h1>Ejemplo de variable en PHP</h1>
9
10     <?php
11         // Primer bloque PHP: creamos la variable
12         $nombre = "Maria";
13     ?>
14
15     <p>El valor de la variable es:</p>
16
17     <?php
18         // Segundo bloque PHP: mostramos la variable
19         echo "<strong> $nombre </strong>";
20     ?>
21 </body>
22 </html>
```

Ejemplo de variable en PHP

El valor de la variable es:

Maria

También hemos visto diferentes ejemplos de la función echo y print

```
<h2>1. Uso básico de echo</h2>
<?php
echo "Hola, mundo!";
?>

<h2>2. Uso básico de echo con varias cadenas</h2>
<?php
echo "Hola, ", "mundo!";
?>

<h2>3. Uso de print con una cadena</h2>
<?php
print "Hola, mundo!";
?>

<h2>4. Uso de echo con variables</h2>
<?php
$nombre = "Juan";
echo "...Hola, " . $nombre . "!";
?>

<h2>5. Uso de print con concatenación</h2>
<?php
$nombre = "Juan";
print "...Hola, " . $nombre . "!";
?>

<h2>6. Uso de echo con HTML</h2>
<?php
echo "<h1>Hola, mundo!</h1>";
?>

<h2>7.Uso de print con HTML</h2>
<?php
print "<h1>Hola, mundo!</h1>";
?>

<h2>8. Uso de echo para mostrar números</h2>
<?php
echo 10 + 20;
?>

<h2>9. Uso de print para mostrar números</h2>
<?php
```

```
print 10 + 20;  
?>  
  
<h2>10. Uso de echo con comillas dobles y simples</h2>  
<?php  
echo "Este es un 'ejemplo' con comillas dobles y simples.";  
?>
```

1. Uso básico de echo

Hola, mundo!

2. Uso básico de echo con varias cadenas

Hola, mundo!

3. Uso de print con una cadena

Hola, mundo!

4. Uso de echo con variables

Hola, Juan!

5. Uso de print con concatenación

Hola, Juan!

6. Uso de echo con HTML

Hola, mundo!

7. Uso de print con HTML

Hola, mundo!

8. Uso de echo para mostrar números

30

9. Uso de print para mostrar números

30

10. Uso de echo con comillas dobles y simples

Este es un 'ejemplo' con comillas dobles y simples.

Programa 3: printf

Printf sirve para dar formato a las variables:

Formato de enteros

%d

Formatea un número entero decimal

%b

Formatea un número entero como binario

%o

Formatea un número entero como octal

%x

Formatea un número entero como hexadecimal minúscula

%X

Formatea un número entero como hexadecimal mayúscula

Formato de números con Punto Flotante

%f

Formatea un número de punto flotante (decimal)

%.2f

Limita el número a 2 decimales

%e

Formatea el número de punto flotante en notación científica minúscula

%E

Formatea el número de punto flotante en notación científica mayúscula

Formato de caracteres

%c

Imprime un carácter basado en su valor ASCII

Alineación y relleno

%5d

Alinea el número a la derecha en un campo de 5 caracteres de ancho

```

<!DOCTYPE html>
<html lang="es">
<body>
</html>

<?php
// Variables relacionadas con productos de una empresa
$producto = "Laptop HP EliteBook"; // Cadena
$precio = 1299.99; // 2 decimales
$descuento = 0.1532; // 4 decimales
$stock = 25; // Entero
$codigoBinario = decbin(num: 2025); // Binario
$peso = 2.345; // Real con 3 decimales
$iva = 0.21; // 2 decimales
$precioFinal = $precio * (1 - $descuento) * (1 + $iva); // Cálculo con decimales
$volumen = 1.234567; // 4 decimales
$precioCientifico = sprintf(format: "%-.2e", values: $precio); // Notación científica

// Mostrar variables por pantalla
echo "Producto: $producto<br>";
echo "Precio: " . number_format(num: $precio, decimals: 2) . " €<br>";
echo "Descuento: " . number_format(num: $descuento, decimals: 4) . "<br>";
echo "Stock disponible: $stock unidades<br>";
echo "Código en binario: $codigoBinario<br>";
echo "Peso: " . number_format(num: $peso, decimals: 3) . " kg<br>";
echo "IVA: " . number_format(num: $iva, decimals: 2) . "<br>";
echo "Precio final: " . number_format(num: $precioFinal, decimals: 2) . " €<br>";
echo "Volumen: " . number_format(num: $volumen, decimals: 4) . " m³<br>";
echo "Precio en notación científica: $precioCientifico<br>";

?>

```

You, 4 days ago • Programa 2 terminado, programa 3 en proceso ...

```

</html>
</body>

```

//Crea un script PHP con diferentes variables numéricas y de cadena relacionadas con productos de alguna empresa y muéstralas por pantalla, dos de ellas con 2 decimales, otras dos con 4 decimales, un entero (sin decimales), una tipo cadena, una de ellas en binario, en notación científica, almacena una variable real, con 3 decimales.. todo lo que quieras usar basándote en los ejemplos anteriores Producto: Laptop HP EliteBook
 Precio: 1,299.99 €
 Descuento: 0.1532
 Stock disponible: 25 unidades
 Código en binario: 1111101001
 Peso: 2.345 kg
 IVA: 0.21
 Precio final: 1.332.01 €
 Volumen: 1.2346 m³
 Precio en notación científica: 1.30e+3

Programa 4:

```
You, 3 days ago | 1 author (You)
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Actividad - printf incompleto</title>
6  </head>
7  <body>
8      <h1>Actividad con printf</h1>
9
10     <?php
11         $aerolinea      = "AirGlobal";;
12         $numVuelo       = 1205;
13         $precioBase     = 245.5;
14         $tasaCombustible= 12.3456;
15         $ocupacion       = 87;           // porcentaje de ocupación
16         $codigoInterno   = 0b101101;
17         $distancia       = 5.6e3;        // 5600 km
18         $velCrucero      = 902.456;      // km/h
19
20         // Imprime el nombre de la aerolínea en 10 espacios, alineado a la izquierda
21         printf(format: "Aerolínea: [%-10s]<br>", values: $aerolinea);
22         // Número de vuelo como entero
23         printf(format: "Vuelo Nº: %d<br>", values: $numVuelo);
24
25         // Precio base con 2 decimales, en un campo de 8 caracteres
26         printf(format: "Precio base: [%8.2f €]<br>", values: $precioBase);
27
28         // Tasa combustible con 4 decimales
29         printf(format: "Tasa combustible: %.4f €<br>", values: $tasaCombustible);
30
31         // Porcentaje de ocupación (mostrar el símbolo %)
32         printf(format: "Ocupación: %d%<br>", values: $ocupacion);
33
34         // Código interno en binario
35         printf(format: "Código interno: %b<br>", values: $codigoInterno);
36         You, 4 days ago • Programa 2 terminado, programa 3 en proceso ...
37         // Distancia en notación científica
38         printf(format: "Distancia: %e km<br>", values: $distancia);
39
40         // Mezcla de variables en la misma línea (vuelo y velocidad)
41         printf(format: "El vuelo %d de %s vuela a %.3f km/h<br>", values: $numVuelo, $aerolinea, $velCrucero);
42     ?>
43
44     <?php
45     echo "<h2>Uso de sprintf</h2>";
46     // Ejemplo con sprintf (genera una cadena formateada sin imprimir)
47     $mensaje = sprintf(
48         format: "El vuelo %d de %s recorrerá %.1f km con un %d% de ocupación.",
49         values: $numVuelo,
50         $aerolinea,
51         $distancia,
52         $ocupacion
53     );
54
55     // Ahora lo mostramos con echo
56     echo "<p><em>Mensaje generado con sprintf:</em> $mensaje</p>";
57
58 ?>
59     </body>
60     </html>
```

Actividad con printf

Aerolínea: [AirGlobal]

Vuelo N°: 1205

Precio base: [245.50 €]

Tasa combustible: 12.3456 €

Ocupación: 87%

Código interno: 101101

Distancia: 5.600000e+3 km

El vuelo 1205 de AirGlobal vuela a 902.456 km/h

Uso de sprintf

Mensaje generado con sprintf: El vuelo 1205 de AirGlobal recorrerá 5600.0 km con un 87% de ocupación.

Programa 5: Cadenas de texto

You, 4 days ago | 1 author (You)

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Cadenas y secuencias de escape en PHP</title>
6      <style>
7          body { font-family: Arial, sans-serif; padding:20px; line-height:1.5; }
8          pre { background:#f4f4f4; padding:10px; border-radius:5px; }
9          h2 { margin-top:25px; }
10     </style>
11 </head>
12 <body>
13     <h1>Estudio de cadenas y secuencias de escape en PHP</h1>
14
15     <?php
16         // Ejemplos con comillas dobles (interpreta escapes)
17         $dobles = "Hola\nMundo\t(esto está tabulado)\nComilla doble: \" Barra invertida: \\\" Variable: \$valor";
18
19         // Ejemplos con comillas simples (no interpreta escapes, salvo \' y \\)
20         $simples = 'Hola\nMundo\t(esto aparece literal)\nComilla simple: '\` Barra invertida: \\\' Variable: $valor';
21
22         // Unicode y Hex
23         $unicode = "Avión: \u{2708} (símbolo avión)";
24         $hex    = "Códigos hexadecimales: \x41\x42\x43 = ABC";           You, 4 days ago • Programa 2 terminado, programa 3 en p
25         // Heredoc (interpreta escapes)
26         $heredoc = <<<TEXT
27 Cadena con heredoc:
28 - Nueva línea \n (se interpreta)
29 - Tabulación \t (se interpreta)
30 - Variable \$unicode: $unicode
31 TEXT;
32
33         // Nowdoc (no interpreta escapes)
34         $nowdoc = <<<'TEXT
35 Cadena con nowdoc:
36 - Nueva línea \n (literal)
37 - Tabulación \t (literal)
38 - Variable $unicode (no sustituida)
39 TEXT;
40     ?>
41
42     <h2>Con comillas dobles</h2>
43     <pre><?php echo $dobles; ?></pre>
44
45     <h2>Con comillas simples</h2>
46     <pre><?php echo $simples; ?></pre>
47
48     <h2>Unicode y Hex</h2>
49     <pre><?php echo $unicode . "\n" . $hex; ?></pre>
50
51     <h2>Heredoc</h2>
52     <pre><?php echo $heredoc; ?></pre>
53
54     <h2>Nowdoc</h2>
55     <pre><?php echo $nowdoc; ?></pre>
56 </body>
57 </html>
```



Estudio de cadenas y secuencias de escape en PHP

Con comillas dobles

```
Hola  
Mundo    (esto está tabulado)  
Comilla doble: "  Barra invertida: \  Variable: $valor
```

Con comillas simples

```
Hola\nMundo\t(esto aparece literal)\nComilla simple: '  Barra invertida: \  Variable: $valor
```

Unicode y Hex

```
Avión: ✈ (símbolo avión)  
Códigos hexadecimales: ABC = ABC
```

Heredoc

```
Cadena con heredoc:  
- Nueva línea  
(se interpreta)  
- Tabulación      (se interpreta)  
- Variable $unicode: Avión: ✈ (símbolo avión)
```

Nowdoc

```
Cadena con nowdoc:  
- Nueva línea \n (literal)  
- Tabulación \t (literal)  
- Variable $unicode (no sustituida)
```

Programa 6: Comprobación y conversión de tipos

Hemos probado funciones como isset, settype, unset,...

```

Dwes_Maria > UD2PHP > Entrega1 > Programa6.php
1  <?php
2 // Programa 6- Comprobación y conversión de tipos
3 /*
4 En este ejercicio trabajaremos con las funciones de **tipos de datos en PHP** .
5
6 1. Declara las siguientes variables:
7   * Una cadena con el valor `Hola mundo` .
8   * Un número entero con el valor `25` .
9   * Un número decimal con el valor `12.34` .
10  * Un array con los valores `avión, helicóptero, dron` .
11  * Una variable con valor `null` .
12 2. Muestra el **tipo de cada variable** utilizando la función `gettype()` .
13 3. Comprueba si las variables son de un tipo concreto utilizando al menos 5 de las siguientes funciones:
14   * `is_array()`, `is_bool()`, `is_float()`, `is_integer()`, `is_null()`, `is_numeric()`, `is_object()`, `is_resource()`, `is_scalar()`, `is_string()` .
15 4. Convierte la variable decimal en cadena usando `settype()` y muestra antes y después el tipo de dato.
16 5. Comprueba si la variable entera está definida y no es `null` con `isset()` .
17 6. Elimina la variable entera con `unset()` y demuestra que ya no existe.
18 */
19
20 // 1. Declaración de variables
21 $cadena = "Hola mundo";
22 $entero = 25;
23 $decimal = 12.34;
24 $lista = array("avión", "helicóptero", "dron");
25 $nulo = null;
26
27 $cadenaPrueba = "hola";
28
29 // 2. Mostrar tipo con gettype
30 echo "Tipo de cadena: " . gettype(value: $cadena) . "<br>";
31 echo "Tipo de entero: " . gettype(value: $entero) . "<br>";
32 echo "tipo de cadenaPrueba: " . gettype(value: $cadenaPrueba) . " $cadenaPrueba <br>";
33
34 // 3. Comprobaciones con is_...
35 if (is_string(value: $cadena)) {
36   echo "La variable es una cadena<br>";
37 }
38 if (is_array(value: $lista)) {
39   echo "La variable es un array<br>";
40 }
41
42 // 4. Conversión con settype
43 echo "Antes de convertir: " . gettype(value: $decimal) . "<br>";
44 settype(var: &$decimal, type: "integer");
45 echo "Después de convertir: " . gettype(value: $decimal) . "<br>";
46
47 // 5. isset
48 if (isset($lista)) {
49   echo "La variable está definida y no es null<br>";
50 }
51
52 // 6. unset
53 unset($entero);
54 echo "Después de unset: " . (isset($entero) ? "existe" : "no existe") . "<br>";
55
56 // 7 mostramos variable No definida
57 echo "<br>Mostramos variable no definida:$entero"; // Notice: Undefined variable: entero
58 ?>

```

← → ⌂ i localhost/DWES/Dwes_Maria/UD2PHP/Entrega1/Programa6.php

Tipo de cadena: string
 Tipo de entero: integer
 tipo de cadenaPrueba: string hola
 La variable es una cadena
 La variable es un array
 Antes de convertir: double
 Despues de convertir: integer
 La variable está definida y no es null
 Después de unset: no existe

Warning: Undefined variable \$entero in C:\xampp\htdocs\DWES\DWES_Maria\UD2PHP\Entrega1\Programa6.php on line 57

Mostramos variable no definida:

Desactivamos los warning

```
// 7 mostramos variable No definida
echo "<br>Mostramos variable no definida antes de desactivar errores :$entero"; // Notice: Undefined variable:
// Oculta todos los warnings (aunque se sigan generando internamente)
error_reporting(error_level: 0); // 7 mostramos variable No definida
echo "<br>Mostramos variable no definida despues de desactivar entero :$entero"; // Notice: Undefined variable
ini_set(option: 'display_errors', value: 0);

echo $variableInexistente; // No muestra nada en pantalla
?>
```

← → ⌂ i localhost/DWES/Dwes_Maria/UD2PHP/Entrega1/Programa6.php

Tipo de cadena: string
Tipo de entero: integer
tipo de cadenaPrueba: double 0
La variable es una cadena
La variable es un array
Antes de convertir: double
Después de convertir: integer
La variable está definida y no es null
Después de unset: no existe

Warning: Undefined variable \$entero in C:\xampp\htdocs\DWES\Dwes_Maria\UD2PHP\Entrega1\Programa6.php on line 58

Mostramos variable no definida antes de desactivar errores :
Mostramos variable no definida despues de desactivar entero :

Programa 7: Constantes

```
Dwes_Maria > UD2PHP > Entrega1 > Programa7.php > ...
1  <?php
2  define(constant_name: "PI", value: 3.14);
3  print "<p>El valor de pi es " . PI . "</p>\n";
4  ?>
5
6  <?php
7  define(constant_name: "LIBRO", value: ["Don Quijote", "Cervantes", 1605]);
8  print "<p>" . LIBRO[1] . " escribió " . LIBRO[0] . " en " . LIBRO[2] . ".</p>\n";
9  ?>
10
11 // Se pueden definir constantes con const
12
13 <?php
14 const PI = 3.14;
15 print "<p>El valor de pi es " . PI . "</p>\n";
16 ?>
17
18 <?php
19 const LIBRO = ["Don Quijote", "Cervantes", 1605];
20 print "<p>" . LIBRO[1] . " escribió " . LIBRO[0] . " en " . LIBRO[2] . ".</p>\n";
21 ?>
22
23 //Las constantes definidas con define() se crean en tiempo de ejecución mientras que las constantes definidas con const se crean en tiempo de compilación.
24
25
26 <?php
27 $decimales = 6;
28 if ($decimales == 6) {
29     define(constant_name: "PI", value: 3.141592);
30 } else {
31     define(constant_name: "PI", value: 3.14);
32 }
33 print "<p>El valor de pi es " . PI . "</p>\n";
34 ?>
35
36 // no podemos usar const
37 <?php
38 $decimales = 6;
39 if ($decimales == 6) {
40     // const PI = 3.141592;
41 } else {
42     // const PI = 3.14;
43 }
44 print "<p>El valor de pi es " . PI . "</p>\n";
45 ?>
46
47
48
49 //Constantes predefinidas
50
51 <?php
52 print "<p>En este servidor los enteros ocupan " . PHP_INT_SIZE . " bytes.</p>\n";
53 ?>
54
55
56 <?php
57 print "<p>En este servidor el mayor entero posible es " . PHP_INT_MAX . ".</p>\n";
58 ?>
59
60 <?php
61 print "<p>En este servidor el menor entero posible es " . PHP_INT_MIN . ".</p>\n";
62 ?>
63
64
65 <?php
66 print "<p>En este servidor el mayor decimal posible es " . PHP_FLOAT_MAX . ".</p>\n";
67 ?>
68
69 <?php
70 print "<p>En este servidor el decimal más próximo a cero posible es " . PHP_FLOAT_EPSILON . ".</p>\n";
71 ?>
```

El valor de pi es 3.14

Cervantes escribió Don Quijote en 1605.

// Se pueden definir constantes con const

El valor de pi es 3.14

Cervantes escribió Don Quijote en 1605.

//Las constantes definidas con define() se crean en tiempo de ejecución mientras que las constantes definidas con const se crean en tiempo de compilación.

El valor de pi es 3.141592

// no podemos usar const

El valor de pi es 3.14

//Constantes predefinidas

En este servidor los enteros ocupan 8 bytes.

En este servidor el mayor entero posible es 9223372036854775807.

En este servidor el menor entero posible es -9223372036854775808.

En este servidor el mayor decimal posible es 1.7976931348623E+308.

En este servidor el decimal más próximo a cero posible es 2.2204460492503E-16.

Programa 8: Constantes

```
1  <?php
2  // --- Constantes propias ---
3  const DEPORTE = "Fútbol";
4  const EQUIPO = "Los Tigres";
5  const ESTADIO = "Gran Arena";
6  const AFORO = 50000;
7
8  // --- Constantes predefinidas de PHP ---
9  $constantes_php = [
10    "PHP_VERSION" => PHP_VERSION,
11    "PHP_OS" => PHP_OS,
12    "__FILE__" => __FILE__,
13];
14
15 // --- Constantes propias en array para analizarlas ---
16 $constantes_propias = [
17    "DEPORTE" => DEPORTE,
18    "EQUIPO" => EQUIPO,
19    "ESTADIO" => ESTADIO,
20    "AFORO" => AFORO,
21];
22
23 // --- Variables de ejemplo ---
24 $jugadores = ["Pedro", "Juan", "Luis"]; // array
25 $goles = 3; // entero
26 $capitan = "Carlos"; // string
27 $esLocal = true; // booleano
28 ?>
29 <!DOCTYPE html>
30 <html lang="es">
31 <head>
32   <meta charset="UTF-8">
33   <title><?= EQUIPO ?> - <?= DEPORTE ?></title>
34   <style>
35     table { border-collapse: collapse; margin: 1em 0; }
36     th, td { border: 1px solid #999; padding: .4em .8em; }
37     th { background: #eee; }
38   </style>
39 </head>
40 <body>
41   <h1><?= EQUIPO ?> (<?= DEPORTE ?>)</h1>
42   <p>Estadio: <strong><?= ESTADIO ?></strong></p>
43   <p>Aforo máximo: <strong><?= AFORO ?></strong> espectadores</p>
44
45   <h2>Constantes de PHP</h2>
46   <table>
47     <tr>
48       <th>Nombre</th><th>Valor</th><th>gettype</th><th>is_string</th><th>is_int</th>
49     </tr>
50     <?php foreach ($constantes_php as $nombre => $valor): ?>
51     <tr>
52       <td><?= $nombre ?></td>
53       <td><?= $valor ?></td>
54       <td><?= gettype(value: $valor) ?></td>
55       <td><?= is_string(value: $valor) ? "sí" : "no" ?></td>
56       <td><?= is_int(value: $valor) ? "sí" : "no" ?></td>
57     </tr>
58     <?php endforeach; ?>
59   </table>
60
61   <h2>Constantes propias</h2>
62   <table>
63     <tr>
64       <th>Nombre</th><th>Valor</th><th>gettype</th><th>is_string</th><th>is_int</th>
65     </tr>
66     <?php foreach ($constantes_propias as $nombre => $valor): ?>
67     <tr>
68       <td><?= $nombre ?></td>
```

```

<tr>
    <td><?= $nombre ?></td>
    <td><?= $valor ?></td>
    <td><?= gettype(value: $valor) ?></td>
    <td><?= is_string(value: $valor) ? "sí" : "no" ?></td>
    <td><?= is_int(value: $valor) ? "sí" : "no" ?></td>
</tr>
<?php endforeach; ?>
</table>

<h2>Variables de ejemplo</h2>
<table>
    <tr>
        <th>Nombre</th><th>Valor</th><th>gettype</th><th>is_array</th><th>is_integer</th><th>is_string</th><th>is_bool</th>
    </tr>
    <tr>
        <td>$jugadores</td>
        <td><?= implode(separator: ", ", array: $jugadores) ?></td>
        <td><?= gettype(value: $jugadores) ?></td>
        <td><?= is_array(value: $jugadores) ? "sí" : "no" ?></td>
        <td><?= is_integer(value: $jugadores) ? "sí" : "no" ?></td>
        <td><?= is_string(value: $jugadores) ? "sí" : "no" ?></td>
        <td><?= is_bool(value: $jugadores) ? "sí" : "no" ?></td>
    </tr>
    <tr>
        <td>$goles</td>
        <td><?= $goles ?></td>
        <td><?= gettype(value: $goles) ?></td>
        <td><?= is_array(value: $goles) ? "sí" : "no" ?></td>
        <td><?= is_integer(value: $goles) ? "sí" : "no" ?></td>
        <td><?= is_string(value: $goles) ? "sí" : "no" ?></td>
        <td><?= is_bool(value: $goles) ? "sí" : "no" ?></td>
    </tr>
    <tr>
        <td>$capitan</td>
        <td><?= $capitan ?></td>
        <td><?= gettype(value: $capitan) ?></td>
        <td><?= is_array(value: $capitan) ? "sí" : "no" ?></td>
        <td><?= is_integer(value: $capitan) ? "sí" : "no" ?></td>
        <td><?= is_string(value: $capitan) ? "sí" : "no" ?></td>
        <td><?= is_bool(value: $capitan) ? "sí" : "no" ?></td>
    </tr>
    <tr>
        <td>$esLocal</td>
        <td><?= $esLocal ? "true" : "false" ?></td>
        <td><?= gettype(value: $esLocal) ?></td>
        <td><?= is_array(value: $esLocal) ? "sí" : "no" ?></td>
        <td><?= is_integer(value: $esLocal) ? "sí" : "no" ?></td>
        <td><?= is_string(value: $esLocal) ? "sí" : "no" ?></td>
        <td><?= is_bool(value: $esLocal) ? "sí" : "no" ?></td>
    </tr>
</table>
</body>
</html>

```

Los Tigres (Fútbol)

Estadio: **Gran Arena**

Aforo máximo: **50000** espectadores

Constantes de PHP

Nombre	Valor	gettype	is_string	is_int
PHP_VERSION	8.2.12	string	sí	no
PHP_OS	WINNT	string	sí	no
__FILE__	C:\xampp\htdocs\DWES\Dwes_Maria\UD2PHP\Entrega1\Programa8.php	string	sí	no

Constantes propias

Nombre	Valor	gettype	is_string	is_int
DEPORTE	Fútbol	string	sí	no
EQUIPO	Los Tigres	string	sí	no
ESTADIO	Gran Arena	string	sí	no
AFORO	50000	integer	no	sí

Variables de ejemplo

Nombre	Valor	gettype	is_array	is_integer	is_string	is_bool
\$jugadores	Pedro, Juan, Luis	array	sí	no	no	no
\$goles	3	integer	no	sí	no	no
\$capitan	Carlos	string	no	no	sí	no
\$esLocal	true	boolean	no	no	no	sí

Vamos a modificar ahora algunos datos

```
<?php

// --- Constantes propias ---
const DEPORTE = "Waterpolo";
const EQUIPO = "Los Delfines";
const ESTADIO = "Gran Agua";
const AFORO = 50000;

// --- Constantes predefinidas de PHP ---
$constantes_php = [
    "PHP_VERSION" => PHP_VERSION,
    "PHP_OS"        => PHP_OS,
    "__FILE__"      => __FILE__,
];

// --- Constantes propias en array para analizarlas ---
$constantes_propias = [
    "deporte" => DEPORTE,
    "equipo"  => EQUIPO,
    "estadio" => ESTADIO,
    "aforo"   => AFORO,
];

// --- Variables de ejemplo ---
$jugadores = ["Maria", "Juan", "Hugo"]; // array
$goles     = 5;                         // entero
$capitan   = "Marcos";                 // string
$esLocal   = true;                     // booleano
?>
```

Los Delfines (Waterpolo)

Estadio: **Gran Agua**

Aforo máximo: **50000** espectadores

Constantes de PHP

Nombre	Valor	gettype	is_string	is_int
PHP_VERSION	8.2.12	string	sí	no
PHP_OS	WINNT	string	sí	no
__FILE__	C:\xampp\htdocs\DWES\DWES_Maria\UD2PHP\Entrega1\Programa8.php	string	sí	no

Constantes propias

Nombre	Valor	gettype	is_string	is_int
deporte	Waterpolo	string	sí	no
equipo	Los Delfines	string	sí	no
estadio	Gran Agua	string	sí	no
aforo	50000	integer	no	sí

Variables de ejemplo

Nombre	Valor	gettype	is_array	is_integer	is_string	is_bool
\$jugadores	Maria, Juan, Hugo	array	sí	no	no	no
\$goles	5	integer	no	sí	no	no
\$capitan	Marcos	string	no	no	sí	no
\$esLocal	true	boolean	no	no	no	sí

Programa 9: Date

Formatos fechas y horas:

d

Dia del mes con 2 dígitos

j

Dia del mes con uno o dos dígitos (sin ceros iniciales)

Z

dia del año, comenzando por el cero (0 = 1 de enero)

N

Dia de la semana (1 = lunes)

W

Dia de la semana (0 = domingo)

I

Texto del dia de la semana en inglés

D

Texto del dia de la semana, solo tres letras en inglés

W

Numero de la semana del año

m

Número del mes con dos dígitos

n

Numeros del mes con uno o dos dígitos (sin ceros iniciales)

t

Numeros de dias que tiene el mes

F

Texto del dia del mes, en ingles

M

Texto del dia del mes solo 3 letras en ingles

Y

Numero del año

y

dos últimos digitos del numero del año

L

1 si es año bisiesto, 0 si no lo es

h

Formato 12 horas siempre con dos dígitos

H

Formato de 24 horas, siempre con dos digitos

g

Formato de 12 horas con uno o dos digitos sin ceros iniciales

G

Formato de 24 horas con uno o dos dígitos sin ceros iniciales

a

am o pm, en minúsculas

A

AM o PM en mayúsculas

r

Fecha entera con formato RFC 2822

```
Dwes_Maria > UD2PHP > Entrega1 > Programa9.php
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Funciones de Fecha y Hora en PHP</title>
6  </head>
7  <body>
8  <h2>Uso de la función date()</h2>
9  <?php
10 echo "<p>Formato completo (d/m/Y H:i:s): " . date(format: "d/m/Y H:i:s") . "</p>";
11 echo "<p>Dia de la semana (l): " . date(format: "l") . "</p>";
12 echo "<p>Mes (F): " . date(format: "F") . "</p>";
13 echo "<p>Año en 4 dígitos (Y): " . date(format: "Y") . "</p>";
14 echo "<p>Hora en 12h con AM/PM (h:i A): " . date(format: "h:i A") . "</p>";
15 echo "<p>Hora en 24h (H:i): " . date(format: "H:i") . "</p>";
16 echo "<p>Timestamp actual (U): " . date(format: "U") . "</p>";
17 echo "<p>Semana del año (W): " . date(format: "W") . "</p>";
18 ?>
19
20 <h2>Uso de la función getdate()</h2>
21 <?php
22 $info = getdate();
23 echo "<pre>";
24 print_r(value: $info); // Muestra todos los detalles de la fecha/hora actual
25 echo "</pre>";
26
27 // Acceder a valores concretos
28 echo "<p>Dia del mes: " . $info['mday'] . "</p>";
29 echo "<p>Mes numérico: " . $info['mon'] . "</p>";
30 echo "<p>Año: " . $info['year'] . "</p>";
31 echo "<p>Hora actual: " . $info['hours'] . ":" . $info['minutes'] . "</p>";
32
33 /*
34 Con date() muestra al menos 8 formatos distintos (d/m/Y, l, F, Y, h:i A, H:i, U, W).
35
36 Con getdate() imprime todos los valores de la fecha actual y luego algunos campos específicos (mday, mon, year, hours, minutes).
37
38 ¿Quieres que lo prepare también en modo ejercicio para alumnos con huecos (por ejemplo, que deban completar los formatos de date() en vez de verlos r
39 */
40 ?>
41 </body>
42 </html>
```

Programa 10: variables superglobales vs constantes predefinidas

```
Dwes_Maria > UD2PHP > Entrega1 >  Programa10.php
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ejemplo de Variables Predefinidas</title>
6  </head>
7  <body>
8
9  <h2>Variables Predefinidas (Superglobales)</h2>
10 <?php
11 echo "<p>Servidor: " . $_SERVER['SERVER_NAME'] . "</p>";
12 echo "<p>Archivo actual: " . $_SERVER['PHP_SELF'] . "</p>";
13 echo "<p>Dirección IP del cliente: " . ($_SERVER['REMOTE_ADDR'] ?? "Desconocida") . "</p>";
14 echo "<p>Parámetro GET 'ejemplo': " . ($_GET['ejemplo'] ?? "No enviado") . "</p>";
15 ?>
16
17 <h2>Constantes Predefinidas</h2>
18 <?php
19 echo "<p>Versión de PHP: " . PHP_VERSION . "</p>";
20 echo "<p>Sistema operativo del servidor: " . PHP_OS . "</p>";
21 echo "<p>Directorio de instalación de PHP: " . PHP_BINDIR . "</p>";
22 echo "<p>Tamaño máximo de enteros (PHP_INT_MAX): " . PHP_INT_MAX . "</p>";
23 echo "<p>Separador de directorios (DIRECTORY_SEPARATOR): " . DIRECTORY_SEPARATOR . "</p>";
24 ?>
25
26 </body>
27 </html>
```

← → ⌂ ⓘ localhost/DWES/Dwes_Maria/UD2PHP/Entrega1/Programa10.php

Variables Predefinidas (Superglobales)

Servidor: localhost

Archivo actual: /DWES/Dwes_Maria/UD2PHP/Entrega1/Programa10.php

Dirección IP del cliente: ::1

Parámetro GET 'ejemplo': No enviado

Constantes Predefinidas

Versión de PHP: 8.2.12

Sistema operativo del servidor: WINNT

Directorio de instalación de PHP: C:\php

Tamaño máximo de enteros (PHP_INT_MAX): 9223372036854775807

Separador de directorios (DIRECTORY_SEPARATOR): \