

- Entrega 1
 - Programa 1
 - Funciones PHP
 - Otros Datos
 - Ámbitos de las Variables
 - Programa 2
 - Diferentes ejemplos usando echo y print
 - Printf y especificadores de tipo
 - Programa 3
 - Programa 4
 - Programa 5
 - Programa 6
 - Programa 7
 - Programa 8
 - Programa 9
 - Programa 10
 - Programa 10Sim

Entrega 1

Actualización de la nueva configuración de carpetas

Programa 1

Funciones PHP

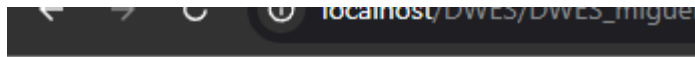
- Podemos alternar en el archivo PHP con código HTML
- Utilización de echo y algunas variables
- Al declarar una variable tienen que comenzar con \$ y a continuación debe ir una letra y no un número

Otros Datos

- **gettype()** --> Sirve para saber qué tipo de dato tiene una variable en tiempo de ejecución.

- **implode** --> Sirve para mostrar una cadena en mi caso seria "\$colores" en una linea separada por comas

Esta captura del Programa1.php despues de declarar unos tipos de variables y mostrarlos



Programa 1

Hola Mundo

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

PHP Tipos de datos

Mostrar tipos de datos creados

Mi nombre es Miguel y tengo 18 años.
Esto es verdadero.
Colores favoritos: rojo, verde, azul.
Marca de mi coche: Toyota.
NULL
Probar el ambito de las variables
Ambito Global

Soy una variable global
Ambito Local

5Ambito Estatica

123|

Ámbitos de las Variables

Podemos encontrar diferentes ámbitos:

- Global: Variables definidas fuera de cualquier función
- Local: Variables definidas dentro de una función
- Estático: Variables locales que persisten su valor entre llamadas a la función usando static

Programa 2

Ejemplo de variable en PHP

El valor de la variable es:

Miguel Torres

```
DWES_miguelTorres > UD2 > Entrega1 > Programa2.php
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 ="Miguel Torres";
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>
```

Como podemos ver en el Programa 2 abajo en el apartado de echo faltaba poner \$nombre y arriba quedaba por asignarle un valor a la variable en este caso "Miguel Torres"

Diferentes ejemplos usando echo y print

```
?>
//Ejemplo
<?php
$nombre = "Juan";
print "Hola, " . $nombre . "!";
?>

<?php
echo "<h1>Hola, mundo!</h1>";
?>

<?php
echo 10 + 20;
?>
```

Ejemplo de variable en PHP

El valor de la variable es:

Miguel Torres //Ejemplo Hola, Juan!

Hola, mundo!

30

Printf y especificadores de tipo

signo (opcional). Indica si se pone signo a los número negativos (por defecto) o también a los positivos (se indica con un signo +).

relleno (opcional). Indica que carácter se usará para ajustar el tamaño de una cadena. Las opciones son el carácter 0 o el carácter espacio (por defecto se usa el espacio).

alineación (opcional). Indica que tipo de alineación se usará para generar la salida: justificación derecha (por defecto) o izquierda (se indica con el carácter -).

ancho (opcional). Indica el mínimo número de caracteres de salida para un parámetro dado.

precisión (opcional). Indica el número de dígitos decimales que se mostrarán para un número real. Se escribe como un dígito precedido por un punto.

tipo (obligatorio). Indica cómo se debe tratar el valor del parámetro correspondiente. En la siguiente tabla puedes ver una lista con todos los especificadores de tipo.

Especificador	Significado
b	el argumento es tratado como un entero y presentado como un número binario.
c	el argumento es tratado como un entero, y presentado como el carácter con dicho valor ASCII.
d	el argumento es tratado como un entero y presentado como un número decimal.
u	el argumento es tratado como un entero y presentado como un número decimal sin signo.
o	el argumento es tratado como un entero y presentado como un número octal.
x	el argumento es tratado como un entero y presentado como un número hexadecimal (con minúsculas).
X	el argumento es tratado como un entero y presentado como un número hexadecimal (con mayúsculas).

f	el argumento es tratado como un doble y presentado como un número de coma flotante (teniendo en cuenta la localidad).
F	el argumento es tratado como un doble y presentado como un número de coma flotante (sin tener en cuenta la localidad).
e	el argumento es presentado en notación científica, utilizando la e minúscula (por ejemplo, 1.2e+3).
E	el argumento es presentado en notación científica, utilizando la e mayúscula (por ejemplo, 1.2E+3).
g	se usa la forma más corta entre %f y %e.
G	se usa la forma más corta entre %f y %E.
s	el argumento es tratado como una cadena y es presentado como tal.
%	se muestra el carácter %. No necesita argumento..

Programa 3

```

1  <?php
2  // Variables numéricas
3  $precioProducto1 = 12.34567;
4  $precioProducto2 = 7.89123;
5  $descuentoProducto1 = 0.12345678;
6  $descuentoProducto2 = 0.98765432;
7  $unidadesDisponibles = 150;
8  $precioCientifico = 12345.6789;
9  $precioBinario = 25;
10 $precioReal = 45.67891;
11
12 // Variables de cadena
13 $nombreProducto = "Cámara";
14 $descripcionProducto = "Cámara digital con zoom 10x";
15
16 // Mostrar por pantalla con especificadores de formato
17 echo "<h2>Información de Productos</h2>";
18
19 printf(format: "Producto: %s<br>", values: $nombreProducto);
20 printf(format: "Descripción: %s<br>", values: $descripcionProducto);
21 printf(format: "Precio Producto 1: %.2f €<br>", values: $precioProducto1);
22 printf(format: "Precio Producto 2: %.2f €<br>", values: $precioProducto2);
23 printf(format: "Descuento Producto 1: %.4f<br>", values: $descuentoProducto1);
24 printf(format: "Descuento Producto 2: %.4f<br>", values: $descuentoProducto2);
25 printf(format: "Unidades disponibles: %d<br>", values: $unidadesDisponibles);
26 printf(format: "Precio en notación científica: %.2e<br>", values: $precioCientifico);
27 printf(format: "Precio en binario: %b<br>", values: $precioBinario);
28 printf(format: "Precio real con 3 decimales: %.3f €<br>", values: $precioReal);
29 >

```

En este script PHP se crean varias variables relacionadas con productos de una empresa, tanto numéricas como de tipo cadena. Se utilizan los **especificadores de formato** de `printf` para mostrarlas por pantalla

Esto permite formatear los datos de forma clara y profesional para su presentación.

Información de Productos

Producto: Cámara

Descripción: Cámara digital con zoom 10x

Precio Producto 1: 12.35 €

Precio Producto 2: 7.89 €

Descuento Producto 1: 0.1235

Descuento Producto 2: 0.9877

Unidades disponibles: 150

Precio en notación científica: 1.23e+4

Precio en binario: 11001

Precio real con 3 decimales: 45.679 €

Programa 4

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Actividad - printf incompleto</title>
</head>
<body>
  <h1>Actividad con printf</h1>

  <?php
    $aerolinea    = "AirGlobal";
    $numVuelo     = 1205;
    $precioBase   = 245.5;
    $tasaCombustible = 12.3456;
    $ocupacion    = 87;           // porcentaje de ocupación
    $codigoInterno = 0b101101;
    $distancia    = 5.6e3;       // 5600 km
    $velCrucero   = 902.456;     // km/h

    // Imprime el nombre de la aerolínea en 10 espacios, alineado a la izquierda
    printf(format: "Aerolínea: [%-10s]<br>", values: $aerolinea);

    // Número de vuelo como entero
    printf(format: "Vuelo Nº: %d<br>", values: $numVuelo);

    // Precio base con 2 decimales, en un campo de 8 caracteres
    printf(format: "Precio base: [%8.2f €]<br>", values: $precioBase);

    // Tasa combustible con 4 decimales
    printf(format: "Tasa combustible: %.4f €<br>", values: $tasaCombustible);

    // Porcentaje de ocupación (mostrar el símbolo %)
    printf(format: "Ocupación: %d%%<br>", values: $ocupacion);

    // Código interno en binario
    printf(format: "Código interno: %b<br>", values: $codigoInterno);

    // Distancia en notación científica
    printf(format: "Distancia: %e km<br>", values: $distancia);

    // Mezcla de variables en la misma línea (vuelo y velocidad)
    printf(format: "El vuelo %d de %s vuela a %.3f km/h<br>", values: $numVuelo, $aerolinea, $velCrucero);
  ?>
</body>
</html>
```

Como podemos ver en los printf ya hemos cambiado los signos de interrogación por su valor

Actividad con printf

Aerolínea: [AirGlobal]
Vuelo Nº: 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

Esto es lo que mostraria la pagina una vez cambiado

Aqui podemos ver la diferencia de printf y sprintf

```
<?php
// Ejemplo con sprintf (genera una cadena formateada sin imprimir)
$mensaje = sprintf(
    format: "El vuelo %d de %s recorrerá %.1f km con un %d%% de ocupación.",
    values: $numVuelo,
    $aerolinea,
    $distancia,
    $ocupacion
);

// Ahora lo mostramos con echo
echo "<p><em>Mensaje generado con sprintf:</em> $mensaje</p>";
?>
```

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

Programa 5

```

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     $valor = 123;
17
18     // Comillas dobles: interpreta escapes y variables
19     $dobles = "Ejemplo con comillas dobles:\n- Salto de línea (\n)\n- Tabulación (\t)\tAquí\n- Comilla doble (\")\n- Barra invertida (\\)\n- Variable: \$valor = $valor";
20
21     // Comillas simples: muestra escapes y variables literalmente
22     $simples = 'Ejemplo con comillas simples:\n- Salto de línea (\n)\n- Tabulación (\\t)\tAquí\n- Comilla simple (\')\n- Barra invertida (\\)\n- Variable: $valor';
23
24     // Unicode y Hex
25     $unicode = "Símbolos Unicode: Corazón \u{2665}, Avión \u{2708}";
26     $hex = "Hexadecimal: \x48\x6F\x6C\x61 = Hola";
27
28     // Heredoc: interpreta escapes y variables
29     $heredoc = <<<EOT
30 Ejemplo con heredoc:
31 - Nueva línea (\n) y tabulación (\t)
32 - Variable \$valor: $valor
33 - Unicode: $unicode
34 EOT;
35
36     // Nowdoc: no interpreta escapes ni variables
37     $nowdoc = <<<'EOT'
38 Ejemplo con nowdoc:
39 - Nueva línea (\n) y tabulación (\t)
40 - Variable $valor (no sustituida)
41 - Unicode: \u{2665} \u{2708}
42 EOT;
43   >>
44
45   <h2>Comillas dobles</h2>

```

Unas de las cosas vistas en el programa 5 es la diferencia de la utilización de las comillas simples y dobles

Programa 6

```
// 1. Declaración de variables
$cadena = "Hola mundo";
$entero = 25;
$decimal = 12.34;
$lista = array("avión", "helicóptero", "dron");
$nulo = null;

// 2. Mostrar tipo con gettype
echo "Tipo de cadena: " . gettype(value: $cadena) . "<br>";
echo "Tipo de entero: " . gettype(value: $entero) . "<br>";

// 3. Comprobaciones con is_...
if (is_string(value: $cadena)) {
    echo "La variable es una cadena<br>";
}
if (is_array(value: $entero)) {
    echo "La variable es un array<br>";
}

// 4. Conversión con settype
echo "Antes de convertir: " . gettype(value: $decimal) . "<br>";
settype(var: &$decimal, type: "string");
echo "Después de convertir: " . gettype(value: $decimal) . "<br>";

// 5. isset
if (isset($entero)) {
    echo "La variable está definida y no es null<br>";
}

// 6. unset
unset($entero);
echo "Después de unset: " . (isset($entero) ? "existe" : "no existe") . "<br>";

// 7 mostramos variable No definida
echo "<br>Mostramos variable no definida: $entero"; // Notice: Undefined variable: entero
?>
```

En el programa 6 hemos usado nuevos elementos como el unset y algunos mas

```
// 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:$entero"; // Notice: Undefined variable: entero
ini_set(option: 'display_errors', value: 0);

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

Añadimos estas lieneas de codigo para ocultar el warning

Programa 7

DWES_miguelTorres > UD2 > Entrega1 > Programa7.php > ...

```
1  <?php
2  define(constant_name: "Telefono", value: 642322234);
3  print "<p>El valor de pi es " . Telefono . "</p>\n";
4  ?>
5
6  <?php
7  define(constant_name: "AUTOR", value: "Don Quejote");
8  print "<p>Mi autor favorito es: " . AUTOR . "</p>\n";
9  ?>
10
11
12 <?php
13 const LIBRO = ["Don Quijote", "Cervantes", 1605];
14 print "<p>" . LIBRO[1] . " escribió " . LIBRO[0] . " en " . LIBRO[2] . ".</p>\n";
15 ?>
16
17
18 <?php
19 $decimales = 6;
20 if ($decimales == 6) {
21     define(constant_name: "PI", value: 3.141592);
22 } else {
23     define(constant_name: "PI", value: 3.14);
24 }
25 print "<p>El valor de pi es " . PI . "</p>\n";
26 ?>
```

Aqui estamos viendo las constantes y viendo las diferentes formas de declararlas y como declararlas de forma correcta

El valor de pi es 642322234

Mi autor favorito es: Don Quejote

Cervantes escribió Don Quijote en 1605.

El valor de pi es 3.141592

Este seria el resultado del codigo de arriba

Programa 8

DWES_miguelTorres > UD2 > Entrega1 > Programa8.php > ...

```
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>
```

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_miguelTorres\UD2\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

Como podemos ver podemos ver varias tablas con un titulo cada uno y unos valores variados viendo variables, cosntantes propias, predefinidas

Programa 9

```
res_miguelonres > BD2 > Entrega1 > Programas9.php
1  <?php
2  // Ejemplo básico de uso de date() y getdate()
3
4  echo "<h2>Uso de date()</h2>";
5  echo "Fecha y hora actual: " . date(format: "d/m/Y H:i:s") . "<br>";
6  echo "Solo el año: " . date(format: "Y") . "<br>";
7  echo "Solo el mes: " . date(format: "F") . "<br>";
8  echo "Solo el día: " . date(format: "l") . "<br>";
9
10 echo "<h2>Uso de getdate()</h2>";
11 $infoFecha = getdate();
12 echo "Array completo:<br>";
13 echo "<pre>";
14 print_r(value: $infoFecha);
15 echo "</pre>";
16 echo "Año: " . $infoFecha['year'] . "<br>";
17 echo "Mes: " . $infoFecha['mon'] . "<br>";
18 echo "Día: " . $infoFecha['mday'] . "<br>";
19 ?>
```

Como podemos ver Date permite obtener la fecha y hora actual en diferentes formatos mientras getdate devuelve un array asociado con los datos de las fechas de cuando se añadieron los datos

Uso de date()

Fecha y hora actual: 29/09/2025 14:28:28

Solo el año: 2025

Solo el mes: September

Solo el día: Monday

Uso de getdate()

Array completo:

Array

```
(  
    [seconds] => 28  
    [minutes] => 28  
    [hours] => 14  
    [mday] => 29  
    [wday] => 1  
    [mon] => 9  
    [year] => 2025  
    [yday] => 271  
    [weekday] => Monday  
    [month] => September  
    [0] => 1759148908  
)
```

Año: 2025

Mes: 9

Día: 29

Programa 10

```

Entrega1.md M x Programa10.php U x
DWES_miguelTorres > UD2 > Entrega1 > Programa10.php
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ejemplo Superglobales PHP</title>
6  </head>
7  <body>
8      <h1>Ejemplo de Superglobales en PHP</h1>
9
10     <h2>1. $_SERVER</h2>
11     <pre><?php print_r(value: $_SERVER); ?></pre> <!-- HUECO 1 -->
12
13     <h2>2. $_GET</h2>
14     <form method="get">
15         <label>Nombre (GET): <input type="text" name="nombre"></label>
16         <input type="submit" value="Enviar GET">
17     </form>
18     <pre><?php print_r(value: $_GET); ?></pre> <!-- HUECO 2 -->
19
20     <h2>3. $_POST</h2>
21     <form method="post">
22         <label>Edad (POST): <input type="number" name="edad"></label>
23         <input type="submit" value="Enviar POST">
24     </form>
25     <pre><?php print_r(value: $_POST); ?></pre> <!-- HUECO 3 -->
26
27     <h2>4. $_REQUEST</h2>
28     <pre><?php print_r(value: $_REQUEST); ?></pre> <!-- HUECO 4 -->
29
30     <h2>5. $_FILES</h2>
31     <form method="post" enctype="multipart/form-data">
32         <label>Subir archivo: <input type="file" name="archivo"></label>
33         <input type="submit" value="Enviar Archivo">
34     </form>
35     <pre><?php print_r(value: $_FILES); ?></pre> <!-- HUECO 5 -->
36
37     <h2>6. $_ENV</h2>
38     <pre><?php print_r(value: $_ENV); ?></pre> <!-- HUECO 6 -->
39
40     <h2>7. $_GLOBALS</h2>
41     <pre><?php print_r(value: $GLOBALS); ?></pre> <!-- HUECO 7 -->
42
43     <h2>8. $_COOKIE</h2>
44     <pre><?php print_r(value: $_COOKIE); ?></pre> <!-- HUECO 8 -->
45
46     <h2>9. $_SESSION</h2>

```

En este programa tenemos que rellenar los huecos de código vacíos

Ejemplo de Superglobales en PHP

1. \$_SERVER

```
Array
(
    [MIBDIRS] => C:/xampp/php/extras/mibs
    [MYSQL_HOME] => \xampp\mysql\bin
    [OPENSSL_CONF] => C:/xampp/apache/bin/openssl.cnf
    [PHP_PEAR_SYSCONF_DIR] => \xampp\php
    [PHPRC] => \xampp\php
    [TMP] => \xampp\tmp
    [HTTP_HOST] => localhost
    [HTTP_CONNECTION] => keep-alive
    [HTTP_SEC_CH_UA] => "Chromium";v="140", "Not=A?Brand";v="24", "Google Chrome";v="140"
    [HTTP_SEC_CH_UA_MOBILE] => ?0
    [HTTP_SEC_CH_UA_PLATFORM] => "Windows"
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
    [HTTP_SEC_FETCH_SITE] => same-origin
    [HTTP_SEC_FETCH_MODE] => navigate
    [HTTP_SEC_FETCH_USER] => ?1
    [HTTP_SEC_FETCH_DEST] => document
    [HTTP_REFERER] => http://localhost/DWES/DWES_miguelTorres/UD2/Entrega1/
    [HTTP_ACCEPT_ENCODING] => gzip, deflate, br, zstd
    [HTTP_ACCEPT_LANGUAGE] => es-ES,es;q=0.9
    [HTTP_COOKIE] => Idea-9fdabec6=f87960d0-c3d0-4519-90c2-ab741c40dfa9; wp-settings-time-1=1739800202; Idea-9fdabec7=8375562f-b702-4732-9313-a7b5aeabb752; Webstorm-fb5fe7cd=d251
    [PATH] => C:\Program Files\Microsoft\jdk-21.0.6.7-hotspot\bin;C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Common Files\Oracle\Java\java8path;C:\
    [SystemRoot] => C:\WINDOWS
    [COMSPEC] => C:\WINDOWS\system32\cmd.exe
    [PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
    [WINDIR] => C:\WINDOWS
    [SERVER_SIGNATURE] =>
    Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at Localhost Port 80

    [SERVER_SOFTWARE] => Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
    [SERVER_NAME] => localhost
    [SERVER_ADDR] => ::1
    [SERVER_PORT] => 80
```

Programa 10Sim

Programa10Sim.php Entrega1.md

DWES_miguelTorres > UD2 > Entrega1 > Programa10Sim.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>
```

Variables Predefinidas (Superglobales)

Servidor: localhost

Archivo actual: /DWES/DWES_miguelTorres/UD2/Entrega1/Programa10Sim.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): \