

UD-02b: Introducción a PHP

Desarrollo Web en Entorno Servidor

Curso 2020/2021

Características básicas del lenguaje

Generación de código HTML

Existen varias formas. La más sencilla es usando **echo**, que genera como salida el texto de los parámetros que recibe.

```
echo "La nota calculada es ".$nota."<br>;"
```

Otra posibilidad es **print**, que funciona de forma similar.

Características básicas del lenguaje

Ejemplo

```
<?php
    $modulo="DWES";
    print "<p>Módulo: ";
    print $modulo;
    print "</p>";
?>
```



```
<?php
    $modulo="DWES";
    echo "<p>Módulo: ";
    echo $modulo;
    echo "</p>";
?>
```



```
<?php
    $modulo="DWES";
    echo "<p>Módulo: ",$modulo,"</p>";
?>
```



```
<?php
    $modulo="DWES";
    print "<p>Módulo: ",$modulo,"</p>";
?>
```



Nos puede servir de ayuda...

Ver el código que recibe el navegador y que se muestra en pantalla:

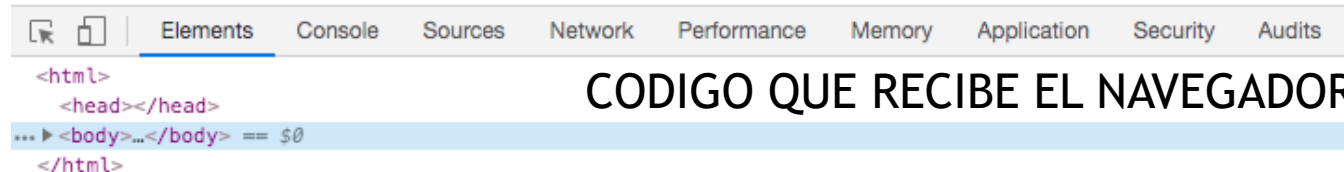
- Botón derecho del ratón sobre una parte de la pagina web que esté vacía (en blanco).
En el menú emergente que aparece, elige **inspeccionar**.
- Otra opción más rápida. Pulsa F12.

Ejemplo

```
<?php
    $modulo="DWES";
    print "<p>Módulo: ";
    print $modulo;
    print "</p>";
?>
```

Módulo: DWES

LO QUE SE MUESTRA EN PANTALLA



CODIGO QUE RECIBE EL NAVEGADOR

Características básicas del lenguaje

printf es otra opción para generar una salida desde PHP. Puede recibir varios parámetros, el primero de los cuales es siempre una cadena de texto que indica el formato que se ha de aplicar.

Esa cadena debe contener un especificador de conversión por cada uno de los demás parámetros que se le pasen a la función, y en el mismo orden en que figuran en la lista de parámetros.

Ejemplo

```
<?php
    $ciclo="DAW";
    $modulo="DWES";
    print "<p>";
    printf("%s es un módulo de %d curso de %s", $modulo, 2, $ciclo);
    print "</p>";
?>
```

Características básicas del lenguaje

Al definir el valor de las variables de tipo string podemos emplear tanto comillas simples como dobles.

Una de las diferencias es que al definir un string con comillas dobles, podemos incorporar variables al contenido entre las comillas, y el valor de estas será expandido automáticamente.

```
<?php
    $usuari = 'Salvador';
    $salutacio = "Hola $usuari";
    echo $salutacio;
?>
```

Características básicas del lenguaje

Si no queremos hacer uso de las comillas dobles, podemos hacer concatenaciones mediante el operador de cadenas '.' (punto)

```
<?php
    $nom = 'Salvador';
    $cognom = 'Mira';
    $salutacio = 'Hola ' . $nom . ' ' . $cognom;
    echo '<p>' . $salutacio . '</p>';
?>
```

Características básicas del lenguaje

```
<?php
```

```
    $modulo="DWES";
```

```
    echo "<p>Módulo: $modulo</p>";
```

```
    echo '<p>Módulo: $modulo</p>';
```

```
?>
```

RESULTADO:

Módulo: DWES

Módulo: \$modulo

Características básicas del lenguaje

Para que PHP distinga correctamente el texto que forma la cadena del nombre de la variable, a veces, es necesario rodearla entre llaves.

Ejemplo

```
<?php
    $modulo="DWES";
    print "<p>Módulo: ${modulo}DAW</p>";
?>
```

Módulo: DWESDAW

Características básicas del lenguaje

Secuencias de escape

- Cuando se encuentra la secuencia de caracteres `\'`, se muestra en la salida una comilla simple
- Cuando se encuentra la secuencia `\\`, se muestra en la salida una barra invertida.

En esta tabla puedes ver las secuencias de escape que se pueden utilizar, y cuál es su resultado.

Secuencia	Resultado
<code>\\</code>	muestra una barra invertida.
<code>\'</code>	muestra una comilla simple.
<code>\"</code>	muestra una comilla doble.
<code>\n</code>	muestra un avance de línea (LF o 0x0A (10) en ASCII).
<code>\r</code>	muestra un retorno de carro (CR o 0x0D (13) en ASCII).
<code>\t</code>	muestra un tabulador horizontal (HT o 0x09 (9) en ASCII).
<code>\v</code>	muestra un tabulador vertical (VT o 0x0B (11) en ASCII).
<code>\f</code>	muestra un avance de página (FF o 0x0C (12) en ASCII).
<code>\\$</code>	muestra un signo de dólar.

Características básicas del lenguaje

(.) → Concatenación

(.=) → Asignación y concatenación

Ejemplo

```
<?php
    $a = "Módulo ";
    $b = $a . "DWES"; // ahora $b contiene "Módulo DWES"
    $a .= "DWES"; // ahora $a también contiene "Módulo DWES"
?>
```

Características básicas del lenguaje

Más funciones relacionadas con las cadenas de texto

<http://es.php.net/manual/es/ref.strings.php>

Características básicas del lenguaje

Funciones relacionadas con los tipos de datos

- **gettype**(variable) → devuelve el tipo de dato: *boolean, array, integer...*

También **is_array()**, **is_bool()**, **is_float()**, ...

- **settype**(variable, nuevo tipo) → para cambiarla de tipo

- **isset**(variable) → nos dice si está definida

- **unset**(variable) → destruye la variable

} ¡Las usaremos! 😊

Características básicas del lenguaje

Ejemplo:

```
<?php
    // asignamos a las dos variables la misma cadena de texto
    $a = $b = "3.1416";
    settype($b, "float"); // y cambiamos $b a tipo float
    echo "\$a vale $a y es de tipo ".gettype($a);
    echo "<br />";
    echo "\$b vale $b y es de tipo ".gettype($b);
?>
```

\$a vale 3.1416 y es de tipo string

\$b vale 3.1416 y es de tipo double

Características básicas del lenguaje

Constantes

Su característica principal es que **NO** se pueden sobrescribir. Se crean con ***define***.

Ejemplo:

```
<?php
    $radio = 5;
    define("PI", 3.141592);           // definiendo una constante numérica
    define("FLAG", true);             //definiendo una constante booleana
    echo "Resultado : " . 2* PI * $radio;
    echo "<br/>Flag: " . FLAG;        // Llamada a constante sin comillas.
?>
```

Las constantes son globales, es decir, accesibles desde cualquier parte del código. Sólo se permiten los siguientes tipos para las constantes: **integer**, **float**, **string**, **boolean** y **null**.

Características básicas del lenguaje

Constantes y Arrays

También podemos definir Arrays como constante (más adelante profundizaremos en los arrays).

Ejemplo:

```
<?php
    define('ANIMALES', array( 'perro', 'gato', 'pájaro' ));
    echo ANIMALES[1];
?>
```


Características básicas del lenguaje

Empty (vacía)

Una variable está vacía, **NO** es lo mismo que decir que es **null**.

Una variable se considera vacía si no existe o su valor es igual a FALSE.

Las siguientes expresiones son **consideradas como vacías**:

- "" (una cadena vacía)
- 0 (cero como un integer)
- 0. (cero como un float)
- "0" (cero como un string)
- FALSE
- array() (un array vacío)
- \$var; (una variable declarada, pero sin un valor)

Características básicas del lenguaje

Fechas y horas

En PHP no existe un tipo de datos específico de fecha/hora.

Tenemos la función **date** que permite obtener una cadena de texto con la fecha y hora, en el formato que especifiquemos.

```
string date (string $formato [, int $fechahora]);
```

La función recibe dos parámetros (**el formato** y **un número entero** que identifica la fecha) y devuelve una cadena de texto formateada.

El segundo parámetro es opcional. Si no se indica, se utilizará la hora actual para crear la cadena de texto.

Carácter	Resultado
d	día del mes con dos dígitos.
j	día del mes con uno o dos dígitos (sin ceros iniciales).
z	día del año, comenzando por el cero (0 = 1 de enero).
N	día de la semana (1 = lunes, ..., 7 = domingo).
w	día de la semana (0 = domingo, ..., 6 = sábado).
l	texto del día de la semana, en inglés (Monday, ..., Sunday).
D	texto del día de la semana, solo tres letras, en inglés (Mon, ..., Sun).
W	número de la semana del año.
m	número del mes con dos dígitos.
n	número del mes con uno o dos dígitos (sin ceros iniciales).
t	número de días que tiene el mes.
F	texto del día del mes, en inglés (January, ..., December).
M	texto del día del mes, solo tres letras, en inglés (Jan, ..., Dec).
Y	número del año.
y	dos últimos dígitos del número del año.
L	1 si el año es bisiesto, 0 si no lo es.
h	formato de 12 horas, siempre con dos dígitos.
H	formato de 24 horas, siempre con dos dígitos.
g	formato de 12 horas, con uno o dos dígitos (sin ceros iniciales).
G	formato de 24 horas, con uno o dos dígitos (sin ceros iniciales).
i	minutos, siempre con dos dígitos.
s	segundos, siempre con dos dígitos.
u	microsegundos.
a	am o pm, en minúsculas.
A	AM o PM, en mayúsculas.
r	fecha entera con formato RFC 2822.

Ejemplos

```
<?php
```

```
date_default_timezone_set('UTC'); // Establece la zona horaria predeterminada a usar.
```

```
echo date("l"); // Imprime el día: Monday, Tuesday, etc...
```

```
echo "<br/>";
```

```
echo date('l jS \of F Y h:i:s A'); // Imprime : Monday 8th of August 2005 03:12:46 PM
```

```
// S → sufijo inglés
```

```
echo "<br/>";
```

```
echo date(DATE_RFC2822); // Imprime : Wed, 25 Sep 2013 15:28:57 -0700
```

```
echo "<br/>";
```

```
echo date(DATE_ATOM, mktime(0, 0, 0, 7, 1, 2000)); // Imprime : 2000-07-01T00:00:00+00:00
```

```
?>
```

También existe **getdate**, una función que devuelve un array con información sobre la fecha y hora actuales.

Clave	Descripción	Ejemplo de valores devueltos
"seconds"	Representacion numérica de los segundos	0 a 59
"minutes"	Representacion numérica de los minutos	0 a 59
"hours"	Representacion numérica de las horas	0 a 23
"mday"	Representacion numérica del día del mes	1 a 31
"wday"	Representacion numérica del día de la semana	0 (para Domingo) hasta 6 (para Sábado)
"mon"	Representacion numérica de un mes	1 hasta 12
"year"	Una representacion numérica completa de una año, 4 dígitos	Ejemplos: 1999 o 2003
"yday"	Representacion numérica del día del año	0 hasta 365
"weekday"	Una representación textual completa del día de la semana	Sunday hasta Saturday
"month"	Una representación textual completa de un mes, como January o March	January hasta December
0	Los segundos desde la Época Unix, similar a los valores devueltos por time() y usados por date() .	Dependiente del Sistema, típicamente -2147483648 hasta 2147483647.

Características básicas del lenguaje

Funciones para gestionar fechas y horas:

<http://php.net/manual/es/ref.datetime.php>

Características básicas del lenguaje

Variables especiales de PHP (superglobales)

`$_SERVER`

Contiene información sobre el entorno del servidor web y de ejecución.

Valor	Contenido
<code>\$_SERVER['PHP_SELF']</code>	guión que se está ejecutando actualmente.
<code>\$_SERVER['SERVER_ADDR']</code>	dirección IP del servidor web.
<code>\$_SERVER['SERVER_NAME']</code>	nombre del servidor web.
<code>\$_SERVER['DOCUMENT_ROOT']</code>	directorio raíz bajo el que se ejecuta el guión actual.
<code>\$_SERVER['REMOTE_ADDR']</code>	dirección IP desde la que el usuario está viendo la página.
<code>\$_SERVER['REQUEST_METHOD']</code>	método utilizado para acceder a la página ('GET', 'HEAD', 'POST' o 'PUT')

<http://es.php.net/manual/es/reserved.variables.server.php>

Características básicas del lenguaje

Otras variables especiales de PHP son:

- `$_GET`, `$_POST` y `$_COOKIE` contienen las variables que se han pasado al guion actual utilizando los métodos GET, POST y Cookies.
- `$_REQUEST` junta el contenido de los tres arrays anteriores, `$_GET`, `$_POST` y `$_COOKIE`, es un solo array.
- `$_ENV` contiene las variables que se puedan haber pasado a PHP desde el entorno en que se ejecuta.
- `$_FILES` contiene los ficheros que se puedan haber subido al servidor utilizando el método POST.
- `$_SESSION` contiene las variables de sesión disponibles para el guion actual.

<http://es.php.net/manual/es/language.variables.superglobals.php>

Las veremos en profundidad más adelante.

EJERCICIO



- ❖ Crea una página llamada ***ejer03.php*** cuyo título sea *Ejercicio03*

En ella aparecerá lo siguiente:

```
Espero sacar un 10 en DWES  
12/09/2019 - 06:30
```

Ten en cuenta los requisitos:

- La nota 10 es una **constante** llamada NOTA, y “DWES” es la variable **\$asignatura**
- La segunda línea muestra la fecha y hora actuales, con el formato que ahí se ve. Se puede usar **date** (o getdate). *Entre sus parámetros deberás usar las letras de la tabla vista anteriormente.*