
ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

IMPLANTACIÓN DE APLICACIONES WEB

ACTIVIDADES UNIDAD 2. Generación de documentos web utilizando lenguajes de script de servidor. SINTAXIS

Primeras Instrucciones

Sintaxis1

Escribe el siguiente código, guárdalo como sintaxis1.php en el directorio prácticas y comprueba el resultado.

```
<html>
<head>
<title>Aprendiendo PHP</title></head>
<body>
<?php
# aquí utilizamos solo unas comillas
echo "Este texto solo lleva las comillas de la instrucción<br>";
# aquí anidaremos comillas de distinto tipo
echo "La palabra 'comillas' aparecerá entrecomillada<br>";
# esta es otra posibilidad invirtiendo el orden de las comillas
echo 'La palabra "comillas" aparecerá entrecomillada<br>';
# una tercera posibilidad en la que utilizamos un mismo
# tipo de comillas. Para diferenciar unas de otras antepone
# la barra invertida, pero esta opción no podríamos utilizarla
# al revés.
# No podríamos poner \" en las comillas exteriores.
echo "La palabra \"comillas\" usando la barra invertida<br>";
?>
</body>
</html>
```

Sintaxis2

Escribe un script en el que se utilicen las funciones print, print() y echo, para presentar las siguientes cadenas ("Sensor de Temperatura", "Sensor de Humedad", "Sensor de Luminosidad"), añadiendo líneas de comentario que expliquen la sintaxis de cada una de las funciones.

Guárdalo como sintaxis2.php en el directorio prácticas y comprueba su funcionamiento.

CONSTANTES

Sintaxis3

Escribe el siguiente código en un fichero sintaxis3.php y observa el resultado.

```
<HTML><HEAD><TITLE>Constantes</TITLE></HEAD>
<BODY>
<?php
/* Definiremos la constante EurPta y le asignaremos el valor 166.386 */
define("EurPta",166.386);
/* Definiremos la constante PtaEur asignándole el valor 1/166.386
En este caso el valor de la constante es el resultado
de la operación aritmética dividir 1 entre 166.386*/
define("PtaEur",1/166.386);
/* Definimos la constante Cadenas y le asignamos el valor:
12Esta constante es una cadena*/
define("Cadena","12Esta constante es una cadena");
/* Definimos la constante Cadena2 y le asignamos el valor:
12.54Constante con punto decimal*/
define("Cadena2","12.54Constante con punto decimal");
/* Comprobemos los valores.
Observa la nueva forma en la que utilizamos echo
Lo hacemos enlazando varias cadenas separadas con
punto y/o coma, según se trate de echo o de print */
echo "Valor de la constante EurPta: ", EurPta, "<BR>";
echo "Valor de la constante PtaEur: ". PtaEur . "<BR>";
print "Valor de la constante Cadena: " . Cadena . "<BR>";
print "Valor de la constante Cadena x EurPta: " . Cadena*EurPta . "<br>";
print "Valor de la constante Cadena2 x EurPta: " . Cadena2*EurPta . "<br>";
echo "Con echo los números no necesitan ir entre comillas: " ,3,<br>";
print "En el caso de print si son necesarias: " . "7" . "<br>";
```

```
print ("incluso entre paréntesis necesitan las comillas: ". "45". "<br>");  
print "Solo hay una excepción en el caso de print. ";  
print "Si los números van en un print independiente no necesitan comillas ";  
print 23;  
# Pondremos la etiqueta de cierre del script y escribiremos  
# una línea de código HTML  
?>  
<br>Ahora veremos los mismos resultados usando la funcion prueba<br><br>
```

```
<?php  
# Estamos dentro de un nuevo script abierto por el <?php anterior  
/* Aunque aún no la hemos estudiado, escribiremos una función  
a la que (tenemos que ponerle siempre un nombre)  
Vamos a llamar prueba()  
Lo señalado en rojo es la forma de indicar el comienzo  
y el final de la función  
Lo marcado en azul son las instrucciones  
que deben ejecutarse cuando la función prueba() sea invocada */
```

```
function prueba(){  
echo "Valor de la constante EurPta: ". EurPta . "<BR>";  
print "Valor de la constante PtaEur: ". PtaEur. "<BR>";  
echo "Valor de la constante Cadena: ", Cadena , "<BR>";  
print ("Valor de la constante Cadena x EurPta: " .  
Cadena*EurPta . "<br>");  
print ("Valor de la constante Cadena2 x EurPta: " .  
Cadena2*EurPta . "<br>");  
}  
# Las funciones solo se ejecutan cuando son invocadas  
/* La función anterior no se ejecutará hasta que escribamos  
una línea –como esta de abajo– en la que ponemos  
únicamente el nombre de la función: prueba()  
*/  
?>
```

```
<?php  
prueba();  
?>  
</body>  
</HTML>
```

Sintaxis4

Escribe un script (guárdalo como sintaxis4.php) en el que se definan dos constantes, una numérica (TemperaturaMax) y otra de cadena (comentario) y en el que –mediante las funciones print y echo– aparezca en la página web resultante la siguiente información:

“Un ordenador no debe trabajar a más de 40° centígrados”.

La constante TemperaturaMax debe contener el valor 40.

La constante comentario debe contener el valor “Un ordenador no debe trabajar a más de”.

VARIABLES

Sintaxis5

Escribe un script (guárdalo como sintaxis5.php) que calcule el coste de un ordenador completo utilizando las siguientes variables:

```
$cpu  
$monitor  
$teclado  
$raton  
$impresora  
$iva  
$total
```

De imprimir en pantalla el precio de cada componente, el iva aplicado y el precio total del ordenador.

Sintaxis6

Escribe un script (guárdalo como sintaxis6.php) que construya una pequeña página web en la que aparezca el texto **"Estás utilizando la versión: 5.5 de PHP"** (incluidas las comillas y extrayendo el número de versión de la constante predefinida correspondiente). Trata de que todas las etiquetas HTML que utilices estén recogidas en variables PHP, de manera que no exista ninguna línea en el código fuente de la página que esté fuera de las etiquetas `<?php ... ?>`

Sintaxis7

- A) Analiza el contenido del siguiente script e investiga cual sería la salida en pantalla anotándola en un papel.

```
<?php
# Observa que hemos prescindido de los encabezados HTML.
# No son imprescindibles para la ejecución de los scripts
/* Escribamos una función y llamémosla sinEstaticas
Definamos en ella dos variables sin ninguna otra especificación
e insertemos las instrucciones para que al ejecutarse
se escriban los valores de esas variables */

function sinEstaticas(){
# Pongamos aquí sus valores iniciales
$a=0;
$b=0;
# Imprimamos estos valores iniciales
echo "Valor inicial de $a: ",$a,"<br>";
echo "Valor inicial de $b: ",$b,"<br>";
/* Modifiquemos esos valores sumando 5 al valor de $a
y restando 7 al valor de $b.
$a +=5 y $b -=7 serán quienes haga esas
nuevas asignaciones de valor
ya lo iremos viendo, no te preocupes */
$a +=5;
$b -=7;
# Visualicemos los nuevos valores de las variables
echo "Nuevo valor de $a: ",$a,"<br>";
echo "Nuevo valor de $b: ",$b,"<br>";
}

# Escribamos ahora la misma función con una modificación que será
# asignar la condición de estática a la variable $b
# Llamemos a esa función: conEstaticas

function conEstaticas(){
# Definimos $b como estática
$a=0;
static $b=0;
echo "Valor inicial de $a: ",$a,"<br>";
echo "Valor inicial de $b: ",$b,"<br>";
$a +=5;
$b -=7;
echo "Nuevo valor de $a: ",$a,"<br>";
echo "Nuevo valor de $b: ",$b,"<br>";
}

# Insertemos un texto que nos ayude en el momento de la ejecución
print ("Esta es la primera llamada a sinEstaticas()<br>");
# Invoquemos la función sinEstaticas;
sinEstaticas();
# Añadamos un nuevo comentario a la salida
print ("Esta es la segunda llamada sinEstaticas()<br>");
print ("Debe dar el mismo resultado que la llamada anterior<br>");
# Invoquemos por segunda vez sinEstaticas;
sinEstaticas();
# Hagamos ahora lo mismo con la función conEstaticas
print ("Esta es la primera llamada a conEstaticas()<br>");
conEstaticas();
print ("Esta es la segunda llamada a conEstaticas()<br>");
print ("El resultado es distinto a la llamada anterior<br>");
conEstaticas();
?>
```

- B) Escribe el script (guárdalo como sintaxis7). Observa el resultado mostrado y compáralo con lo que anotaste, estudias las diferencias si las hubiera.

Sintaxis 8

Copia el siguiente script (guárdalo como sintaxis8) y ábrelo en el navegador. Estudia el resultado obtenido

```
<?php
# Definamos una variable y asignémosle un valor
$color="rojo";
# Definamos ahora una nueva variable de nombre variable
# usando para ello la variable anterior
$$color=" es mi color preferido";
# Veamos impresos los contenidos de esas variables
print ( "El color ".$color. "$$color."<br>");
#o también
print ( "El color ".$color. "${color}."<br>");
# o también
print ( "El color ".$color. $rojo."<br>");
# advertamos lo que va a ocurrir al visualizar la página
print ("Las tres líneas anteriores deben decir lo mismo<br>");
print ("Hemos invocado la misma variable de tres formas diferentes<br>");
# cambiemos ahora el nombre del color
$color="magenta";
/* La variable $rojo seguirá existiendo.
El hecho de cambiar el valor a $color
no significa que vayan a modificarse
las variables creadas con su color anterior
ni que se creen automáticamente variables
que tengan por nombre el nuevo valor de $color */
# Pongamos un mensaje de advertencia para que sea visualizado en la salida
print ("Ahora la variable $color ha cambiado a magenta<br>");
print ("pero como no hemos creado ninguna variable con ese color<br>");
print ("en las líneas siguientes no aparecerá nada <br>");
print ("detrás de la palabra magenta <br>");
# Escribimos los print advertidos
print (" El color ".$color.$$color."<br>");
print (" El color ".$color.${color}."<br>");
# Comprobemos que la variable $rojo creada como variable de variable
# cuando $color="rojo" aún existe y mantiene aquel valor
print ("Pese a que $color vale ahora ".$color."<br>");
print ("la vieja variable $rojo sigue existiendo <br>");
print ("y conserva su valor. Es este: ".$rojo);
?>
```