

Un ejemplo con constantes

¿Qué es una constante?

Una **constante** es un valor –un número o una *cadena*– que **no va a ser modificado** a lo largo del proceso de ejecución de los *scripts* que contiene un documento.

Para mayor comodidad, a cada uno de esos valores se le *asigna un nombre*, de modo que cuando vaya a ser *utilizado* **baste** con escribir su nombre.

Cuando *ponemos nombre* a una *constante* se dice que **definimos** esa constante.

¿Cómo definir constantes?

En **PHP** las constantes se definen mediante la siguiente instrucción:

```
define("Nombre","Valor")
```

Los valores asignados a las constantes se mantienen en todo el documento, *incluso* cuando son invocadas desde *una función*.

No es necesario escribir *entre comillas* los valores de las constantes cuando se trata de constantes **numéricas**.

Si se realizan operaciones aritméticas con constantes tipo **cadena**, y su valor **comienza por una letra**, PHP les asigna **valor cero**.

Si una *cadena* **empieza** por **uno o varios caracteres numéricos**, al tratar de operarla aritméticamente PHP considerará **únicamente** el valor de los **dígitos anteriores a la primera letra o carácter no numérico**.

El **punto** entre caracteres numéricos es considerado como **separador de parte decimal**.

Tal como puedes ver en el *código fuente* del ejemplo que tienes al margen, es posible **definir constantes** a las que se **asigne** como **valor** el *resultado de una operación aritmética*.

Ampliando echo

Mediante *una sola* instrucción **echo** se pueden *presentar* (en la ventana del navegador del cliente) *de forma simultánea* varias *cadenas de caracteres* y/o *constantes* y

```
<HTML><HEAD><TITLE>Constantes</TITLE></HEAD>
<BODY>
<?
/* 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 function prueba<br><br>
<?

# Estamos dentro de un nuevo script abierto por el <? 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
```

variables. Basta con **ponerlas** una a continuación de otra utilizando **una coma** como *separador* entre cada una de ellas.

La forma anterior no es la única –ni la más habitual– de enlazar elementos mediante la instrucción **echo**. Si en vez de utilizar la **coma** usáramos un **punto** (el *concatenador* de cadenas) conseguiríamos el mismo resultado.

Cuando *enlacemos* elementos distintos –cadenas, constantes y/o números– hemos de tener muy en cuenta lo siguiente:

– Cada una de las sucesivas **cadenas** debe ir encerrada entre sus **propias** comillas.

– Los **nombres** de constantes **nunca** van entre comillas.

Ampliando *print*

Las instrucciones **print** también permiten concatenar cadenas en una misma instrucción.

En este caso **solo es posible** usar el **punto** como elemento de unión. Si pusiéramos *comas* –como hacíamos con **echo**– PHP nos daría un **error**.

```
/* 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()
*/

?>

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

[Ver ejemplo8.php](#)

Ejercicio nº 4

Escribe un script (guárdalo como **ejercicio4.php**) en el que se definan dos constantes, una numérica y otra de cadena y en el que –mediante la las opciones **print** y **echo**– aparezca en la página web resultante un comentario sobre el tipo de cada una de ellas seguido de su valor. Intenta conseguir que los elementos concatenados aparezcan en la presentación separados por un espacio. ¡Observa el código fuente!

¡Cuidado!

Presta mucha atención a la sintaxis. Olvidar los «;» o no poner unas comillas suelen ser la causa de muchos mensajes de error.

[Anterior](#)

[Índice](#)

[Siguiente](#)