


¿Qué son las constantes? ¿cómo declarar constantes en PHP?

- **Constante:** Espacio en la memoria del ordenador donde se almacenará un valor que no podrá cambiar durante la ejecución del programa



```
define("NOMBRE_CONSTANTE" , Valor);
```

A veces veremos la necesidad de almacenar valores y asegurarnos que no van a cambiar; EN este momento, declaramos una constante.

¿cómo declarar constantes?

En la imagen se ve la sintaxis de la función `define()` que nos sirve para declarar constantes.

Esta función admite tres parámetros: 2 obligatorios y uno opcional los dos obligatorios son el nombre de la constante, siempre en mayúsculas, y el valor: Si es numérico sin comillas si es de tipo texto con comillas.

El tercer argumento opcional permite ignorar el que vaya en mayúsculas, aunque normalmente van en mayúsculas. Es una buena forma de que a la hora de programar se distinga antes donde hay variables constantes.

Aspectos a tener en cuenta con las constantes:

- **Aspectos a tener en cuenta con las constantes:**
 - El nombre de las constantes ha de ir en mayúsculas (convenio).
 - El nombre de las constantes no deben llevar el símbolo \$.
 - Es obligatorio el uso de la función `define()` para definir constantes.
 - El ámbito de las constantes es global por defecto.
 - Las constantes no se pueden redefinir.
 - Las constantes solo pueden almacenar valores escalares.

Valor escalar es aquel que no se puede dividir en partes más pequeñas .

Ej de valores escalares: decimal, entero, string, booleano

Ej de valores no escalares: Los array

Conclusión una constante no puede almacenar un array porque un array si que puedes dividirlo en partes más pequeñas.

EJEMPLO

1. Creamos un archivo, llamado uso contantes
2. Creamos zona PHP
3. Definimos constante

```
<body>

<?php
    define( string $name , mixed $value [, bool $case_insensitive = false ] )
    define (

?>
```

El primer argumento es un string del nombre de la constante

El segundo es el valor de la constate

El tercero es el booleano true o false para especificar si queremos que sea sensible a las mayúsculas o no lo sea.

el primero y el segundo son obligatorios, el tercero es opcional

ej define("autor", "juan")

4. ¿Cómo imprimir una constante?
Echo Autor;
hasta aquí igual q con las variables
5. Si queremos introducir una constante en una cadena de texto.
ojo ¡! Hay matices diferentes de las constantes con las variables:
 - a. Una variable puedo incluirla dentro de un string, con las constantes
no. El nombre de la constante la trata como texto.

conclusión: Al combinar un string con el valor almacenad en una variable , habrá que concatenarlo con un punto.

- b. Para introducir el nombre de la constante en minúscula, habrá que introducir como 3º argumento true, para indicar que sea sensible a mayúsculas y minúsculas.

```
<?php

define("AUTOR", "Juan", true);
I
echo "El autor es : " . autor;

?>
```

- c. Cualquier intento de modificar valor de la contante va a dar error (de ahí su nombre)

Ejemplo curioso

```
<?php

define("AUTOR", "Juan", true);

define("AUTOR", "María");

echo "El autor es : " . autor;

?>
```

Partimos de que no se puede cambiar el valor de una constante si pongo la sintaxis del pantallazo

- si pongo true y el echo en minúscula, devuelve Juan
- si no, devuelve María porque sale en mayúsculas.

- d. Hasta aquí se trata de contantes propias, porque las defines tu. Pero en PHP disponemos de contantes predefinidas (Buscar en google “constantes predefinidas PHP” y sale PHP.net con las constantes predefinidas:

Constantes predefinidas

PHP ofrece un largo número de [constantes predefinidas](#) a cualquier script en ejecución. Muchas de estas constantes, sin embargo, son creadas por diferentes extensiones, y sólo estarán presentes si dichas extensiones están disponibles, bien por carga dinámica o porque han sido compiladas.

Hay ocho constantes predefinidas que cambian dependiendo de donde son usadas. Por ejemplo el valor de `__LINE__` depende en la línea que se use en el script. Estas constantes especiales son sensibles a mayúsculas y son las siguientes:

Varias constantes "mágicas" de PHP

Nombre	Descripción
<code>__LINE__</code>	El número de línea actual en el fichero.
<code>__FILE__</code>	Ruta completa y nombre del fichero con enlaces simbólicos resueltos. Si se usa dentro de un include, devolverá el nombre del fichero incluido.
<code>__DIR__</code>	Directorio del fichero. Si se utiliza dentro de un include, devolverá el directorio del fichero incluido. Esta constante es igual que <code>dirname(__FILE__)</code> . El nombre del directorio no lleva la barra inicial a no ser que esté en el directorio root.
<code>__FUNCTION__</code>	Nombre de la función.
<code>__CLASS__</code>	Nombre de la clase. El nombre de la clase incluye el namespace declarado en (p.e.j. <code>Foo\Bar</code>). Tenga en cuenta que a partir de PHP 5.4 <code>__CLASS__</code> también funciona con traits. Cuando es usado en un método trait, <code>__CLASS__</code> es el nombre de la clase del trait que está siendo utilizado.
<code>__TRAIT__</code>	El nombre de el trait. El nombre del trait incluye el namespace si alguno fué declarado en (p.e.j. <code>Foo\Bar</code>).
<code>__METHOD__</code>	Nombre del método de la clase.
<code>__NAMESPACE__</code>	Nombre del espacio de nombres actual.

Véase también [get_class\(\)](#), [get_object_vars\(\)](#), [file_exists\(\)](#) y [function_exists\(\)](#).

Historial de cambios

Versión	Descripción
---------	-------------

a sintaxis de las constantes predefinidas comienzan con dos guiones bajos y acaban con lo mismo.

Dependiendo de la versión PHP que tengas, algunos estarán disponibles y otros no:

- i. Donde la ponga me devuelve el número de línea del fichero
- ii. La segunda si la ponemos nos devuelve la ruta completa y el nombre del fichero

Probamos:

```
<?php

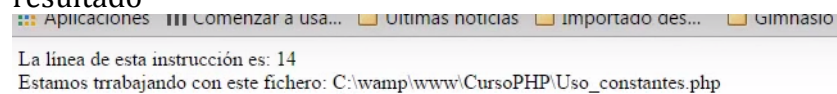
define("AUTOR", "Juan", true);

echo "La línea de esta instrucción es: " . __LINE__ . "<br>";

echo "Estamos trabajando con este fichero: " . __FILE__;

?>
```

resultado



La línea de esta instrucción es: 14
Estamos trabajando con este fichero: C:\wamp\www\CursoPHP\Uso_constantes.php



CÁLCULO MATEMÁTICO

1. OPERADORES MATEMÁTICOS

OPERADOR	NOMBRE
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo
++ +=	Incremento
-- -=	Decremento

Los habituales: + - * /, suma ,resta, multiplicación y división
Operador módulo: Devuelve el resto de una división:

$$\begin{array}{r} 10 \overline{) 3} \\ 1 \times 3 \end{array}$$

El resto es 1

Incremento ++: Incremento en 1

Decremento—

+= incrementa el valor de una variable el un número concreto

EJEMPLO

- Hacemos un formulario como el de la foto:

```
<body>

<p>&nbsp;</p>
<form name="form1" method="post" action="">
  <p>
    <label for="num1"></label>
    <input type="text" name="num1" id="num1">
    <label for="num2"></label>
    <input type="text" name="num2" id="num2">
    <label for="operacion"></label>
    <select name="operacion" id="operacion">
      <option>Suma</option>
      <option>Resta</option>
      <option>Multiplicación</option>
      <option>División</option>
      <option>Módulo</option>
    </select>
  </p>
  <p>
    <input type="submit" name="button" id="button" value="Enviar" onClick="prueba">
  </p>
</form>
<p>&nbsp;</p>

</body>
</html>
```

vista del ejemplo en el navegador

6 3 Multiplicación

Enviar

18

18

tiene :

- Dos cuadros para introducir dos números
- Un menú desplegable para escoger operación
- Botón enviar

Según expresa la imagen: esto se envía al servidor y una vez la información procesada nos la envía de vuelta para que aparezca donde le ha indicado el programador

Pasos de programación:

- Primero programamos donde va a salir el resultado.
abrimos bloque al final , después del formulario

```
<?php

    if(isset($_POST["button"])){

        $numero1=$_POST["num1"];
        $numero2=$_POST["num2"];
        $operacion=$_POST["operacion"];

    }

?>
```

hacemos un condicional con la función isset en cuyo contenido ponemos la variable superglobal para recibir información del formulario.

declaramos todas las variables y hacemos el condicional

```
<?php

    if(isset($_POST["button"])){

        $numero1=$_POST["num1"];
        $numero2=$_POST["num2"];
        $operacion=$_POST["operacion"];

        if(!strcmp("Suma",$operacion)){

            echo "El resultado es: " . ($numero1+$numero2);

        }

        if(!strcmp("Resta",$operacion)){

            echo "El resultado es: " . ($numero1-$numero2);

        }

        if(!strcmp("Multiplicación",$operacion)){

            echo "El resultado es: " . ($numero1*$numero2);

        }

        if(!strcmp("División",$operacion)){

            echo "El resultado es: " . ($numero1/$numero2);

        }

        if(!strcmp("Módulo",$operacion)){

            echo "El resultado es: " . ($numero1%$numero2);

        }

    }

?>
```

2. Si son iguales el string suma y lo que elige el usuario , entonces "l resultado es " .
3. Pongo el paréntesis de la suma porque quiero que lo concateno todo. Es decir primero suma el interior y luego lo concatene con el texto