

Ciclo de Grado Superior: **Desarrollo de Aplicaciones WEB**

MÓDULO 2ºDAW: **Desarrollo Web en entorno Servidor.**

Autor: Jorge López.



Tema 01: Sintaxis básica en PHP - Parte 4ª

11 Cadenas

- Introducción
- Operaciones con cadenas

11 Cadenas.

Introducción

Si pensamos en los tipos de datos que circulan por la red, llegaremos a la conclusión de que una gran porción la ocupan las imágenes, animaciones, videos o subprogramas escritos en Java. La otra gran porción mayoritaria, son los **textos o las cadenas de caracteres**.

Las cadenas de caracteres o string son secuencias de caracteres que pueden ser tratadas como una unidad, asignadas a variables, pasadas como parámetros a funciones o enviadas como salida al navegador. Un string se diferencia de otro tipo de dato en PHP 5-PHP 7, porque va encerrado entre comillas dobles (") o simples (').

"Cadena entre comillas dobles"

'Cadena entre comillas simples'

PHP **interpreta de distinta forma** las cadenas que van entre comillas dobles y las que van entre comillas simples. Los strings entre comillas dobles pueden sustituir ciertos símbolos por acciones, como por ejemplo la inclusión del valor de una variable. Las comillas simples, simplemente muestran todo el contenido, sin atender a caracteres especiales.

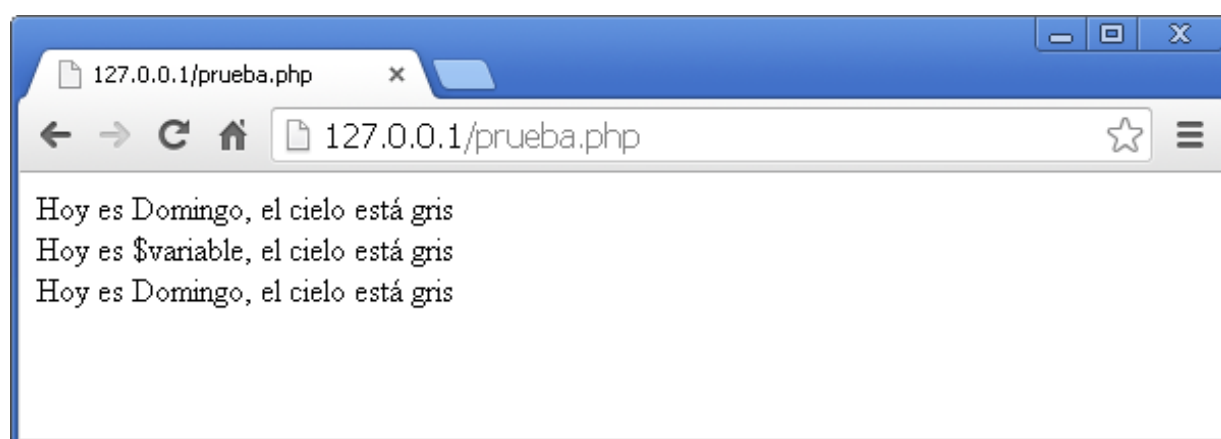
Podemos ver un ejemplo donde sucede esto:

Crea un fichero llamado prueba.php y prueba este código:

```
<?php
```

```
header('Content-Type: text/html; charset=UTF-8');  
# también se puede cambiar AddDefaultCharset en la configuración de Apache  
  
$variable = "Domingo";  
$frase_1 = "Hoy es $variable, el cielo está gris <br>";  
$frase_2 = 'Hoy es $variable, el cielo está gris <br>';  
$frase_3 = 'Hoy es '.$variable.', el cielo está gris <br>';  
echo $frase_1;  
echo $frase_2;  
echo $frase_3;
```

```
?>
```



Operaciones con cadenas

Índices de string:

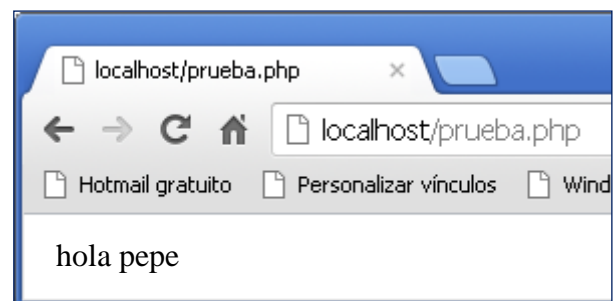
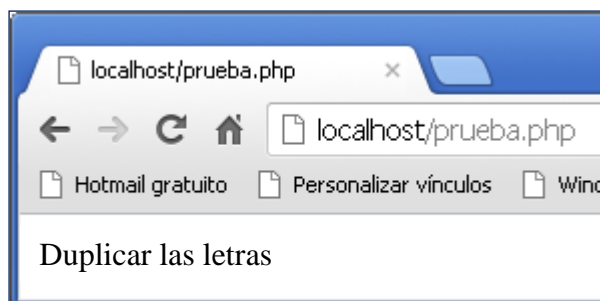
Si pensamos en las cadenas como una sucesión de caracteres en un orden determinado, podemos llegar a desear acceder libremente a parte de los caracteres. Esto es posible gracias a los símbolos de llave { } y un índice numérico que se corresponderá con la posición del carácter que busquemos.

El **ejemplo** muestra cómo crear una función que duplica las letras de una cadena aprovechando esta forma de acceder a los caracteres:

```
<?php
function duplicar_caracteres($cadena)
{
    $nc=strlen($cadena);
    $auxiliar="";
    for ($x=0;$x<$nc;$x++)
    {
        $auxiliar = $auxiliar.$cadena[$x];
    }
    return $auxiliar;
}
$cadena = "Duplicar las letras";
echo duplicar_caracteres($cadena);
?>
```

```
<?php
$cadena="hola pepe";
$longitud=strlen($cadena);

// el primer carácter está en la posición 0
// el último carácter está en la posición
// longitud-1
for ($x=0;$x<$longitud;$x++)
{
    echo $cadena[$x];
}
?>
```

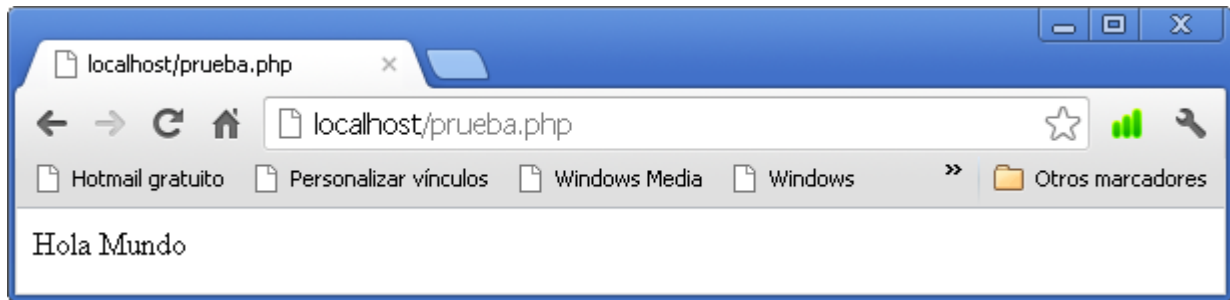


Operadores:

En este punto, aprovecharemos para hacer un breve repaso de los operadores de string que ya hemos comentado alguna vez. En otros lenguajes, como Java, se utiliza el operador suma (+) para unir dos cadenas. En PHP este mismo resultado se obtiene con el operador punto (.).

Así, podemos concatenar varias cadenas de la forma siguiente:

```
<?php
$cadena1="Hola";
$cadena2="Mundo";
$supercadena=$cadena1." ".$cadena2;
echo $supercadena;
?>
```

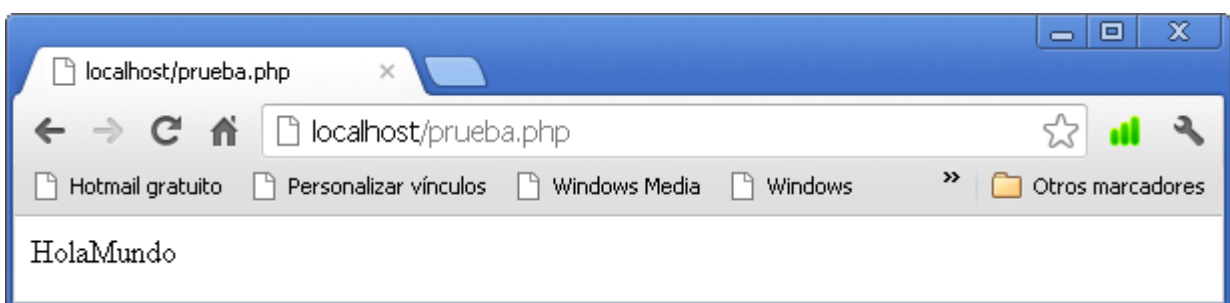


Es fácil intuir que el operador puede concatenar caracteres y variables de tipo string de forma conjunta. Es posible que desee ahora añadir texto a una cadena ya existente, esto se puede hacer de dos formas muy similares:

```
<?php
$cadena1="Hola";
$cadena2="Mundo";
$cadena1=$cadena1.$cadena2;
echo $cadena1;
?>
```

```
<?php
$cadena1="Hola";
$cadena2="Mundo";
$cadena1.=$cadena2;
echo $cadena1;
?>
```

En cualquier caso el resultado será el mismo:



Sintaxis para múltiples líneas:

Existe en **PHP** una forma más de introducir cadenas, aparte de las comillas, muy recomendable para largos textos o un considerable conjunto de etiquetas HTML. La sintaxis es muy sencilla y comienza con el operador (**<<<**) seguido de una etiqueta que indica el principio del texto. Después de esto podemos escribir un conjunto de caracteres muy numerosos y para finalizar, y en una línea aparte, pondremos el nombre del identificador y un punto y coma:

```
$var= <<< identificador
    contenido de la cadena...
    puede ir ....
    en varias líneas...
identificador;
```

Veamos un [ejemplo](#) de esto:

```
<?php
```

```
$cadena1="Esto es una cadena de texto<br>;
```

```
$cadena2= <<<Pepa
```

```
Esta es otra cadena con el nombre Pepa  
que puedo escribir Pepa cuantas veces quiera  
porque el PHP no interpretará el final de  
documento incrustado hasta que no la escriba  
en una sola línea y seguida del punto y coma<br><br>
```

```
Si quiero reflejar los saltos de línea tendré  
que poner el símbolo br
```

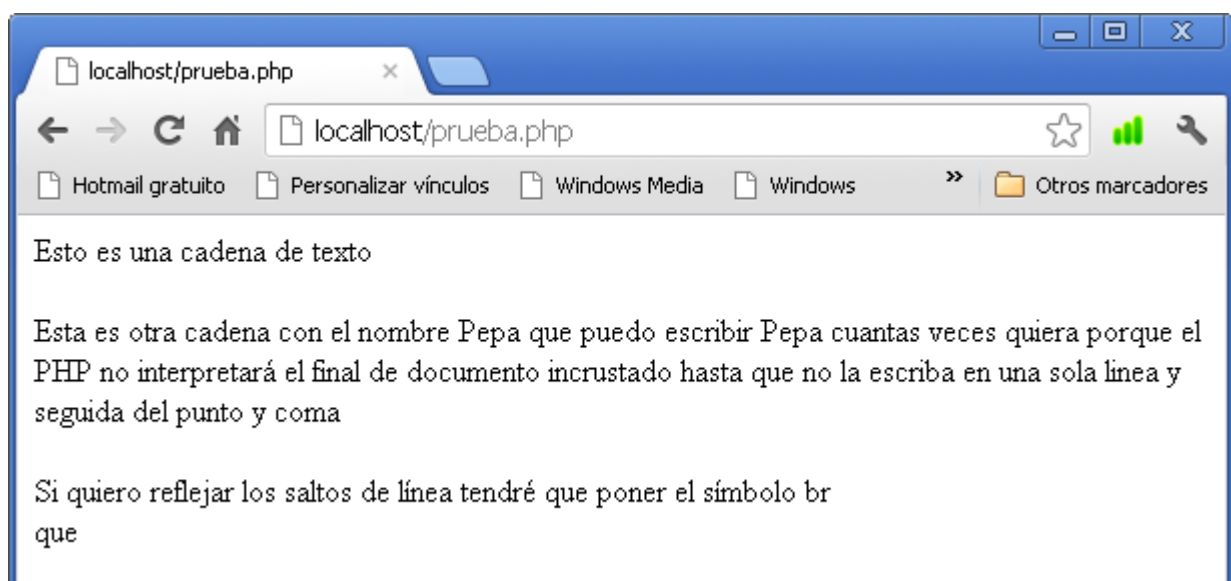
```
Pepa;
```

```
echo $cadena1,"<BR>";
```

```
echo $cadena2,"<BR>";
```

```
echo $cadena2[40].$cadena2[41].$cadena2[42];
```

```
?>
```



Funciones de cadena:

Existen numerosas funciones predefinidas relacionadas con cadenas.
Ya citamos algunas en el tema anterior.