

1 Curso POO PHP Cadeas de texto

1.1 Cadeas de texto

En PHP as cadeas de texto poden usar tanto comiñas simples coma comiñas dobres. Non obstante hai unha diferenza importante entre usar unhas ou outras. Cando se pon unha variable dentro dunhas **comiñas dobres**, procésase e substitúese polo seu valor. Ademais, para que PHP distinga correctamente o texto que forma a cadea do nome da variable, ás veces é necesario rodeala entre chaves.

```
<?php
echo "O código da cor azul é {$cor['azul']}";
?>
```

Cando se usan **comiñas simples**, só se realizan dúas substitucións dentro da cadea: cando se encontra a secuencia de caracteres `\'`, móstrase na saída unha comiña simple; e cando se encontra a secuencia `\\`, móstrase na saída unha barra invertida.

Estas secuencias coñécense como secuencias de escape. Nas cadeas que usan comiñas dobres, ademais da secuencia `\\`, pódense usar algunhas máis, pero non a secuencia `\'`. Nesta táboa podes ver as secuencias de escape que se poden utilizar, e cal é o seu resultado.

Secuencia	Resultado
<code>\\</code>	amosa unha barra invertida
<code>\'</code>	amosa unha comiña simple
<code>\"</code>	amosa unha comiña dobre
<code>\n</code>	amosa un avance de liña (LF o 0x0A (10) en ASCII)
<code>\r</code>	amosa un retorno de carro (CR o 0x0D (13) en ASCII)
<code>\t</code>	amosa un tabulador horizontal (HT o 0x09 (9) en ASCII)
<code>\v</code>	amosa un tabulador vertical (VT o 0x0B (11) en ASCII)
<code>\f</code>	amosa un avance de páxina (FF o 0x0C (12) en ASCII)
<code>\\$</code>	amosa o símbolo do dólar

En PHP tes outra alternativa para crear cadeas: **a sintaxe heredoc**. Consiste en poñer o operador `<<<` seguido dun identificador da túa elección, e a continuación e empezando na liña seguinte a cadea de texto, sen utilizar comiñas. A cadea finaliza cando escribes ese mesmo identificador nunha nova liña. Esta liña de peche non debe levar máis caracteres, nin sequera espazos ou sangría, salvo quizais un punto e coma despois do identificador.

```
echo <<<DOC
Este é un exemplo
da sintaxe "heredoc".
O valor da variable \$a é $a.
DOC;
```

O texto procésase de igual forma que se fose unha cadea entre comiñas dobres, substituindo variables e secuencias de escape. Se non quixeses que se realizase ningunha substitución, debes poñer o identificador de apertura entre comiñas simples. Neste caso tamén se lle chama sintaxe **nowdoc**.

```
echo <<<'DOC'
Este é un exemplo
da sintaxe "heredoc".
Non se amosa o valor da variable $a.
DOC;
```

Para manexar cadeas de texto en PHP podemos empregar o **conxunto de funcións** dispoñibles no núcleo, moitas das cales manteñen o mesmo nome e sintaxe que as da librería estándar da linguaxe C.

1.1.1 Codificación e escapado

As cadeas de texto que empregamos dentro de páxinas web deben axustarse a unhas regras. O mesmo acontece cos textos que empregamos nas URLs ou nas consultas a bases de datos. PHP contén funcións axudarnos a converter estas cadeas según o fin ao que estean destinadas.

A función **htmlspecialchars** devolve o mesmo texto que se lle pasa como parámetro, cambiando os seus caracteres (excepto o espazo) polas entidades HTML equivalentes (por exemplo, o carácter "`<`" por `<`"). O seu inverso é **htmlspecialchars_decode**. Outra función semellante é **htmlspecialchars_decode**, que soamente converte os mínimos caracteres necesarios para obter texto HTML válido. O seu inverso é **htmlspecialchars_decode**.

Outra función útil para traballar con textos que conteñen etiquetas HTML é **strip_tags**, que elimina as etiquetas HTML do texto.

Á hora de traballar co URLs temos a función **rawurlencode**, que adapta a cadea que lle pasamos para que poida ser empregada como parte dunha URL, e o seu inverso, **rawurldecode**.

E para sentenzas SQL, a función **addslashes** engade unha barra invertida diante de certos caracteres como comiñas (dobre e simples) ou outras barras invertidas, de xeito que a cadea resultante poda ser empregada como parte dunha consulta. A súa función inversa é **stripslashes**.

--Víctor Lourido 14:42 25 jun 2013 (CEST)