

Entorns de Desenvolupament



8. Cadenes

8. Cadenes

Les **cadenes** de **caràcters** en PHP es defineixen, entre **cometes dobles o simples** (podem fer servir dobles o simples indistintament). És a dir, podem definir una variable com a cadena de caràcters de la següent manera:

```
$cadena = "IES Simarro";
```

o també

```
$cadena = 'IES Simarro';
```

Per al tractament d'aquestes variables o cadenes de caràcters ens serà útil conèixer les funcions bàsiques de manipulació. No entrarem a descriure cadascuna de les [funcions per a cadenes](#) de caràcters que PHP posseeix, que en són moltes, simplement veurem **les més bàsiques** i d'ús més comú.

echo en realitat no és una funció, és una construcció de PHP per poder incrustar codi php dins de codi HTML. Es pot utilitzar amb cometes dobles o amb cometes dobles i amb o sense parèntesi. Amb cometes simples, les variables i seqüències d'escapament per a caràcters especials no seran expandides en l'ordre **echo**:

El resultat d'aquesta declaració no serà l'esperada:

```
$cadena = 'IES Simarro';  
echo 'Valor de la cadena: $cadena';
```

Valor de la cadena: \$cadena

En canvi, si la cadena està tancada entre cometes dobles ("), PHP substituirà els valors de les variables i a més, interpretarà, entre altres, les següents seqüències d'escapament per a caràcters especials:

Seqüència	Significat
\n	nova línia (LF or 0x0A (10) en ASCII)
\r	retorn de carro (CR or 0x0D (13) en ASCII)
\t	tabulació horitzontal (HT or 0x09 (9) en ASCII)
\v	tabulació vertical (VT or 0x0B (11) in ASCII) (des de PHP 5.2.5)
\e	escapament (ESC or 0x1B (27) in ASCII) (des de PHP 5.4.4)
\f	avanç de pàgina (FF or 0x0C (12) in ASCII) (des de PHP 5.2.5)
\\	barra invertida
\\$	símbol de dòlar
\"	cometes dobles

PHP proporciona una gran quantitat de [funcions](#) per treballar amb cadenes, en veurem a continuació només unes quantes:

Funció **strlen**: `strlen(string $cadena): int`

La funció **strlen** obté la **longitud de la cadena que se li passa com a paràmetre** i retorna un nombre **sencer**. És a dir, obtenim el nombre de caràcters que formen la cadena completa.

Exemple01:

```
<?php
    $cad="cadena";
    $n=strlen($cad);
    echo "La longitud de la cadena és: $n";
?>
```

Si la cadena tinguera algun caràcter que no fora de l'[ASCII](#) original, el comptaria com a doble. Per exemple el valor de `strlen("adéu")` seria **5**, no 4 com esperaríem.

Si volem comptar els caràcters segons el joc de caràcters que estem utilitzant en la pàgina, és millor utilitzar la funció **mb_strlen**.

`mb_strlen(string $cadena, ?string $codificacio = null): int`

```
<?php
```

```
    $cad="adéu";
```

```
    $n=mb_strlen ($cad);
```

```
    echo "La longitud de la cadena és: $n"; //mostra 4
```

```
?>
```

Perquè realment es puguin utilitzar les funcions mb_, cal instal·lar prèviament un paquet en Ubuntu (en aquest cas per a la versió 8.1 de PHP):

```
$ sudo apt install php8.1-mbstring
```

Funció substr:

```
substr(string $string, int $offset, ?int $length = null): string
```

La funció **substr** retorna una **subcadena de l'original**, és a dir, una part de l'original. Aquesta funció té 3 paràmetres:

- La **variable** que conté la **cadena**.
- Nombre que representa la **posició** (inclusivament) en la qual **començarà** a ser extreta la **subcadena**. **Posició d'inici de la subcadena dins de la cadena**.
- Nombre que indica la **quantitat de caràcters** que seran **extrets**. Aquest argument és **opcional**. Si s'omet s'extraurà la cadena des de la posició d'inici indicada fins al final.

És important tenir en compte que la **numeració** de la **posició** dels caràcters **comença** per **zero**. Per exemple per la cadena 'GPS', que té 3 caràcters:

- El caràcter en **posició 0** és **G**
- La **posició 1** és **P**
- La **posició 2** és **S**

No hi ha la posició 3 a causa de que es comença a comptar pel zero.

Exemple02:

<?php

```
$cad="cadena";  
$sub1=substr($cad, 2); // des del caràcter 2 fins al final  
  
$sub2=substr($cad, 2, 2); // des del caràcter 2, 2 caràcters  
echo "La subcadena número u és la següent: $sub1 <br/>";  
echo "La subcadena número dos és la següent: $sub2";  
?>
```

El programa mostraria com a resultat:

La subcadena número u és la següent: dena

La subcadena número dos és la següent: de

Exemple03:

<?php

```
$cad="IES Lluís Simarro" ;  
$sub1=substr($cad,0 ); // La cadena completa  
$sub2=substr($cad,0,3 );// La subcadena  
$sub3=substr($cad,4,5);// La subcadena  
$sub4=substr($cad,10); // La subcadena  
echo "La subcadena número u és la següent: $sub1<br/>";  
echo "La subcadena número dos és la següent: $sub2<br/>";  
echo "La subcadena número tres és la següent: $sub3<br/>";  
echo "La subcadena número quatre és la següent: $sub4";  
?>
```

Cal fixar-se en les següents qüestions:

- a) La **numeració** de caràcters **comença** en **zero** i no en u. Així en aquest últim exemple el caràcter 0 és la I, el caràcter 1 la E, el 2 la S, el 3 l'espai en blanc i així

successivament.

- b) La **posició** en què **comença l'extracció** va **inclosa** en la **subcadena** extreta. Això vol dir que **substr(\$cad,0,1)** només extreu el primer caràcter de la cadena.

Operador de concatenació: .

La concatenació és el procés **d'unir diverses porcions** de **text** en **una sola cadena**. Aquest operador s'expressa com un **punt '.'**

Exemple04:

```
<?php
    $cad1="Carpe" ;
    $cad2="Diem" ;
    $cadt=$cad1.$cad2;
    echo "La cadena 1 concatenada amb la cadena 2 és: $cadt <br/>" ;
    // concatenem en aquest ordre: Exemple: amb $cad2 i $cad1
    echo "Exemple: ".$cad2.$cad1;
?>
```

En aquest exemple veiem com les cadenes es poden **concatenar** en l'**ordre** que nosaltres **desitgem**.

funció str_word_count:

```
str_word_count(string $cadena, int $format = 0, ?string $characters = null): array|int
```

La funció **str_word_count()** **compta la quantitat de paraules** que hi ha en una cadena que se li passa com a paràmetre. Per defecte el separador entre paraules és el caràcter d'espai. Per **exemple**:

```
<?php
```

```
echo str_word_count("Hello world!");  
echo str_word_count("A E I O U")
```

```
?>
```

Dóna com a eixida: **2** i **5** respectivament.

Funció **strrev**: `strrev(string $cadena): string`

La funció **strrev()** **inverteix el contingut** de la cadena que se li passa com a paràmetre. Per **exemple**:

```
<?php  
echo strrev("Hello world!");  
?>
```

Dóna com a eixida: **!dlrow olleH**

Funció **strpos**:

`strpos(string $paller, string $agulla, int $offset = 0): int|false`

La funció **strpos()** **busca un text específic dins d'una cadena**. Si es troba una coincidència, la funció retorna la posició de la primera coincidència. Si no es troba cap coincidència, retorna false. Cal anar amb compte amb els **comparacions**, perquè la **posició 0** (la subcadena **s'ha trobat** i està al principi de la cadena) pot **assimilar-se al valor false** (**no s'ha trobat** la cadena). La funció té dos paràmetres:

- El **primer**: Cadena en la que anem a buscar.
- El **segon**: text que volem buscar dins de la cadena del paràmetre anterior.

En el següent **exemple** es busca el text "world" en la cadena "Hello world!":

```
<?php  
echo strpos("Hello world!", "world");  
?>
```

El resultat de l'operació és 6. Recordant que el primer caràcter d'una cadena està en la posició 0, no en la 1.

Possibles confusions:

```
<?php
    if (strpos("Hello world!", "Hola")==false){
        echo "La cadena no s'ha trobat";
    }
    if (strpos("Hello world!", "Hello")==0){
        echo "La cadena s'ha trobat en la posició 0";
    }
    if (strpos("Hello world!", "Hola")==0){
        echo "La cadena no s'ha trobat";
    } else{
        echo "La cadena s'ha trobat";
    }
?>
```

Funció str_replace

`str_replace(array|string $busca, array|string $reemplaça,
string|array $cadena, int &$count = null): string|array`

La funció `str_replace()` **reemplaça alguns caràcters amb alguns altres caràcters** dins d'una cadena. Té quatre paràmetres:

- El **primer**: `$busca` : cadena a substituir.
- El **segon**: `$reemplaça` : cadena que substituirà a l'anterior.
- El **tercer**: `$cadena` : cadena sobre la que es van a realitzar les operacions.
- El **quart**: `$count` : és opcional. S'hi ficarà la quantitat de reemplaçaments fets.

Es pot utilitzar també amb arrays en lloc de cadenes. En el següent **exemple** es reemplaça el text "world" amb "Dolly":

```
<?php
```



```
$cad = str_replace("world", "Dolly", "Hello world!");  
echo $cad;
```

```
?>
```

El resultat de l'operació serà: **Hello Dolly!**

```
<?php
```

```
$cadena = str_replace("a", "", "Sargantana"); //elimina les a  
echo $cadena;
```

```
?>
```

El resultat de l'operació serà: **Srgntn**

Funció explode

`explode(string $separador, string $cadena, int $limit = PHP_INT_MAX): array`

La funció **explode()** **divideix una cadena en funció d'una altra cadena** creant com a resultat un array de cadenes. Té dos paràmetres:

- El **primer**: **\$separador** : el delimitador en funció del qual dividirem la cadena.
- El **segon**: **\$cadena** : la cadena que volem dividir.
- El **tercer**: **\$limit** : és opcional.
 - **\$limit** > 0 : l'array retornat tindrà com a màxim \$limit elements. L'últim element és la resta de la cadena.
 - **\$limit** < 0 : es tornen tots els components excepte els darrers -**\$limit**.
 - **\$limit** = 0 : es tracta com si fora 1.

En el següent exemple, dividim una cadena pels espais que continga ' ', el resultat és un array que conté les diferents cadenes en què s'ha dividit la cadena inicial.

```
<?php
```

```
$cadena = "blau roig verd groc taronja negre";  
$colors = explode(" ", $cadena);  
echo $colors[0];
```

```
echo $colors[3];
```

```
?>
```

A l'eixida tindríem els valors **blau** i **groc**.

En aquest altre exemple dividim una cadena pel caràcter o (que es lleva de les cadenes resultants):

```
<?php
```

```
    $cadena = "blau roig verd groc taronja negre";
```

```
    $colors = explode("o", $cadena);
```

```
    echo $colors[0];
```

```
    echo $colors[3];
```

```
?>
```

A l'eixida tindríem els valors **blau r** i **nja negre**.

Funció strtolower: strtolower(string \$string): string

La funció **strtolower()** converteix a **minúscules el contingut** de la cadena que se li passa com a paràmetre. Per **exemple**:

```
<?php
```

```
    $cadena = strtolower("IES Lluís Simarro");
```

```
    echo $cadena;
```

```
?>
```

Dóna com a eixida: **ies lluis simarro**

Funció strtoupper: strtoupper(string \$string): string

La funció **strtoupper()** converteix a **majúscule el contingut** de la cadena que

se li passa com a paràmetre. Per **exemple**:

```
<?php
```

```
    $cadena = strtoupper("IES Lluís Simarro");  
    echo $cadena;
```

```
?>
```

Dóna com a eixida: **IES LLUÍS SIMARRO**