Entorns de Desenvolupament

13. Més funcions de cadenes php

ma

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

IES DR. LLUÍS SIMARRO

13. Més funcions de cadenes php

FUNCIÓ str_replace

Aquesta funció s'utilitza per **reemplaçar caràcters dins d'una cadena de caràcters**. És a dir, ens permet definir una cadena que ha de ser reemplaçada amb una altra dins d'una frase o paraula. La funció **torna** la **frase original amb** totes les **aparicions** de la **cadena** a cercar **reemplaçades** amb una **cadena** de reemplaçament **especificada**.

La sintaxi bàsica habitual per a aquesta funció és:

```
str_replace ("cadena a buscar",
             "cadena de reemplaçament",
             $variableOFraseOriginal)
```

Exemple 01:

```
<?php
      $text = "on deia blanc ara diu negre.";
      echo str_replace("blanc" , "blau" , $text);
      echo "<br>" ;
      echo $text; //valor de la variable $text
?>
```

A la cadena de caràcters `$text`, hem substituït la paraula o conjunt de caràcters "blanc" per "blau" i hem tornat el resultat sense definir la variable d'entrada.

En el codi anterior **no hem reemplaçat el valor de la variable $text per un nou contingut**. Únicament **hem imprès per pantalla** el **resultat** que ens retorna la funció. Si haguérem escrit el següent si haguérem canviat el contingut de la variable:

```
$text = str_replace("Diego", "retallades", $text);
```

La funció permet passar opcionalment un paràmetre addicional que permet saber a posteriori el nombre de reemplaçaments que s'han realitzat:

```
str_replace ("cadena a buscar", "cadena de reemplaçament",
          $variableOFraseOriginal, $nombreReemplaçaments)
```

Exemple 02:

```
<?php
     $text="Ella va dir: està millor estudiant que no fent res";
     echo str_replace("es", "**", $text, $reemplaçaments);
     echo "S'han realitzat: ".$reemplaçaments." Reemplaçaments <br>";
     echo "Variable text: ".$text;
?>
```

El resultat esperat és:

Ella va dir: **tà millor **tudiant que no fent r"**

S'han realitzat: 3 Reemplaçaments

Variable text: Ella va dir: està millor estudiant que no fent res

Es reemplaçaria la cadena "es" indistintament segons estiguera separada per espais o dins d'una paraula.

FUNCIONS strtolower i strtoupper

Les funcions **strtolower** i **strtoupper** **transformen** una cadena de caràcters en la mateixa cadena en **minúscules** o **majúscules** respectivament.

Exemple 03:

```
<?php
	$cadena="Això és una cadena de caràcters" ;
	echo strtolower($cadena);
	echo "<br>" ;
	echo strtoupper($cadena);
?>
```

Com veiem el comportament de les funcions és senzill. Simplement tornen la cadena de caràcters passada com argument en minúscules o majúscules respectivament.

això és una cadena de caràcters
AIXò éS UNA CADENA DE CARàCTERS

Moltes vegades per **fer comparacions** o per **emmagatzemar dades** serà interessant **uniformitzar** la **informació** que s'emmagatzema. Per exemple, si es demana una ciutat, un usuari pot introduir El Genovés, un altre el Genovés i un altre EL GENOVÉS. Si utilitzem aquestes funcions, podem uniformitzar i fer que sempre es mostren o guarden d'una mateixa manera, la qual cosa facilitarà la feina posterior.

Si volem que també convertisca els caràcters que no formen part del codi ascii original (accents, dièresi, ç, ñ, etc.), hem d'utilitzar la funció:

```
mb_strtoupper($cadena,'utf-8')
```

Perquè realment es puguen utilitzar les funcions mb_, cal instal·lar un paquet en ubuntu (en aquest cas per a la versió 8.3 de php):

```
$ sudo apt install php8.3-mbstring
```

FUNCIÓ count_chars i substr_count

La funció **count_chars** serveix per comptar el nombre d'aparicions d'un caràcter en una cadena.

La sintaxi a emprar és la següent:

```
count_chars ($cadena, $opcMode)
```

$OpcMode és un enter opcional. Si no s'especifica val 0 per defecte. Els seus valors admesos són:

- **0**: es retornarà un **array** amb el valor numèric **ascii** com a **índex** i la **freqüència** de cada **caràcter** ascii com a valor.
- **1**: es retornarà un **array** amb el valor numèric **ascii** com a **índex** i la **freqüència** de cada **caràcter** que aparega a **mínim una vegada** com a valor.
- **2**: es retornarà un **array** de **caràcters** que **no apareixen** a la **cadena**, amb el valor numèric ascii com a índex i la freqüència de cada caràcter ascii que no apareix com a valor.
- **3**: retorna una **cadena** que conté tots els **caràcters únics**.
- **4**: retorna una **cadena** que conté tots els **caràcters no utilitzats**.

Els codis numèrics ascii van de 0 a 255 i no tots són visibles per pantalla.

Exemple 04:

```
<?php
	$cadena='hui es dijous';
	$meuArray=count_chars($cadena,1);
	foreach($meuArray as $indexNum => $vegades) {
		echo 'Lletra: '.chr($indexNum).', Trobada '. $vegades.' vegades. <br> ';
	}
?>
```

El resultat esperat és:

Lletra: , Trobada 2 vegades.

Lletra: d, Trobada 1 vegades.

Lletra: e, Trobada 1 vegades.

Lletra: h, Trobada 1 vegades.

Lletra: i, Trobada 2 vegades.

Lletra: j, Trobada 1 vegades.

Lletra: o, Trobada 1 vegades.

Lletra: s, Trobada 2 vegades.

Lletra: u, Trobada 2 vegades.

Una altra funció útil és `substr_count($cadena,$subcadena)`. Aquesta funció ens retorna el nombre de vegades que apareix la subcadena dins de la cadena. Per exemple:

```
<?php
      $cadena1='Pere Joan Lluís Marc Lluís Pere Joan Lluís Pere';
      $cadena2 = 'Lluís';
      echo 'Lluís apareix: '.substr_count($cadena1, $cadena2). ' vegades
<br>';
?>
```

El resultat esperat és :

Lluís apareix 3 vegades

ALTRES FUNCIONS PER CADENES DE TEXT

PHP defineix nombroses funcions natives per al maneig de cadenes de text. Anem a veure algunes d'elles:

- **strlen($cadena)**

Retorna la **longitud** o **nombre** de **caràcters** de la **cadena**. Exemple:

```
$cadena="hui és divendres"; //caràcters amb accent compten x2
echo strlen($cadena);
```

Eixida:

17

- **substr($cadena, $inici, $opcNumCar)**

Si no s'especifica **$opcNomCar retorna** la **subcadena entre** la posició $**inici** i la **fi** de **cadena** sent la posició inicial la zero. Si **s'especifica $opcNomCar extreu $opcNomCar caràcters des de** la **posició $inici** (inclosos).

- **ucfirst($cadena);**

Retorna la **cadena** amb la **primera lletra** en **majúscules**. Exemple:

```
$salutacio="hui és divendres";
echo ucfirst($salutacio);
```

Eixida:

Hui és divendres

- `ucwords($cadena);`

Retorna la **cadena** amb **cadascuna** de les seues **paraules** amb la **primera lletra** en **majúscules**. Exemple:

```
$salutacio="hui és divendres";
echo ucwords($salutació);
```

Eixida:

Hui És Divendres

- `strpos($cadena, $subcadena, $opcPosInici)`

Retorna la **posició** en què **comença** la **subcadena** a **partir** de la **posició inicial** (zero) si **no s'especifica** `$opcPosInici`, o a partir de la posició `$opcPosInici` **si s'especifica**. Si no es troba la subcadena retorna false. Exemple:

```
$salutacio="hui és divendres";
echo strpos($salutacio, "és");
```

Eixida:

4

- `trim($cadena, $opcCaracters)`

Si no s'especifica `$opcCaracters`, retorna la mateixa cadena de la variable `$cadena` **eliminant espais** en blanc, **tabuladors**, **salts** de **línia** i **retorns** de **carro** del **principi** i **final** de la **cadena**. Si s'especifica `$opcCaracters`, s'eliminen els **caràcters especificats**. Exemple:

```
$salutacio="*** hui és divendres ***";
echo trim($salutacio, "*");
```

Eixida:

hui és divendres

Nota: pot no funcionar com s'espera a causa de joc de caràcters, configuració local i configuració del servidor.

- **ltrim($cadena, $opcCaracters)**

Si no s'especifica `$opcCaracters`, retorna la mateixa cadena indicada en la variable $cadena **eliminant espais** en **blanc**, **tabuladors**, **salts** de **línia** i **retorns** de **carro** del **principi** de la **cadena**. Si s'especifica `$opcCaracters`, s'eliminen els **caràcters especificats**. Exemple:

```
$salutacio="*** hui és divendres ***";
echo ltrim($salutacio,"*");
```

Eixida:

hui és divendres *

Nota: pot no funcionar com s'espera a causa de joc de caràcters, configuració local i configuració del servidor.

- **rtrim($cadena, $opcCaracters)**

Si no s'especifica `$opcCaracters`, retorna la mateixa cadena de la variable `$cadena`, però **eliminant espais** en **blanc**, **tabuladors**, **salts** de **línia** i **retorns** de **carro**

del final de la cadena. Si s'especifica $opcCaracters, s'eliminen els caràcters especificats. Exemple:

```
$salutacio="*** hui és divendres ***";
echo rtrim($salutacio, "*");
```

Eixida:

*** hui és divendres

Nota: pot no funcionar com s'espera a causa de joc de caràcters, configuració local i configuració del servidor.

- **str_repeat($cadena, $nVegades)**

Retorna la cadena repetida tantes vegades com indique $nVegades sent aquest un nombre sencer. Exemple:

```
$salutacio='hola';
str_repeat($salutacio,3);
```

Eixida:

holaholahola

- **strstr($cadena, $des, $opcBoolean)**

Retorna la subcadena des de la primera aparició de $des de (inclosa) fins al final si no s'especifica $opcBoolean o aquest és false. Si $opcBoolean és true retorna la subcadena des de l'inici fins a la primera aparició de $des de (exclosa). Exemple:

```
$salutacio="hui és divendres";
echo strstr($salutacio,"és", false);
```

Eixida:

```
és divendres
```

- `strchr`

  Igual que `strstr`

- `chr($ascii)`

Retorna el **caràcter** corresponent al **codi** numèric enter `$ascii` segons el codi **ascii**. Exemple:

```
echo chr(65);
```

Eixida:

```
A
```

- `ord($cadena)`

Retorna el **valor ascii** corresponent al primer caràcter de la variable `$cadena` que conté un string que se li passa com a paràmetre. Exemple:

```
$cadena="a";
echo ord($cadena);
```

Eixida:

```
97
```

COMPARACIÓ DE CADENES PHP

Quan comparem cadenes en PHP, es **recomana** fer ús de les **funcions natives** del llenguatge previstes per a això. Si es tracten de fer comparacions utilitzant l'operador

== es poden obtenir resultats estranys. Utilitzar l'operador === pot donar millors resultats, però en general tractarem d'usar funcions com strcmp previstes específicament per a això.

FUNCIÓ strcmp

Aquesta funció realitza la **comparació segura** de Strings i retorna un valor numèric. La seua sintaxi habitual és la següent:

```
if (strcmp ($cadena1, $cadena2) == 0) {...}
```

strcmp és **sensible** a **majúscules** i **minúscules**, és a dir, no considera igual dimarts que Dimarts.

La funció retorna un valor numèric que pot ser:

- 0: quan les dues **cadenes** són **iguals**. En cas contrari el valor retornat és diferent de zero, per la qual cosa si volem saber **si** dues cadenes **són diferents** podem fer servir:

```
if (strcmp ($cadena1, $cadena2)! == 0) {...}
```

- Un **valor numèric menor que 0** si **$cadena1** és **menor** que **$cadena2**
- Un **valor numèric més gran que zero** si **$cadena1** és **més gran** que **$cadena2**.

Què vol dir que una cadena siga més gran que una altra? La **comparació** es fa en **funció** dels **codis numèrics** equivalents de **cada caràcter**. Per exemple la lletra A té codi numèric 65 i la lletra a codi numèric 97. A causa que els **codis numèrics no** permeten una **ordenació alfabètica precisa** (en no ordenar correctament majúscules i minúscules, paraules amb accent, caràcters com la ñ, etc.), aquest **no** és un **bon mitjà** per **ordenar alfabèticament paraules** i només té una utilitat limitada.

Els paràmetres aportats a la funció han de ser Strings. En cas que algun **paràmetre no siga** una **cadena de text**, el **resultat** que retorna la funció pot ser **impredictible**.

Escrivim aquest codi i visualitzem els resultats que produeix:

```
<?php
	$cadena1='1e3'; $cadena2='1000';
	if($cadena1==$cadena2) {
		echo 'Segons == les dues cadenes són iguals <br>';
	} else {
		echo 'Segons == les dues cadenes NO són iguals <br>';
	}

	if($cadena1 === $cadena2) {
		echo 'Segons === les dues cadenes són iguals <br>';
	} else {
	echo 'Segons === les dues cadenes NO són iguals <br>';
	}

	if(strcmp($cadena1,$cadena2) == 0) {
		echo 'Segons strcmp les dues cadenes són iguals <br>';
	} else {
		echo 'Segons strcmp les dues cadenes NO són iguals <br>';
	}
?>
```

El resultat serà similar a aquest:

Segons == les dues cadenes són iguals

Segons === les dues cadenes NO són iguals

Segons strcmp les dues cadenes NO són iguals.

Recordeu que per fer comparacions segures de cadenes de text farem servir `strcmp` o una altra funció PHP prevista per a això en lloc dels operadors `==` o `===`.

ALTRES FUNCIONS PER COMPARAR CADENES AMB PHP

- `strcasecmp($cadena1, $cadena2)`

Retorna com a **resultat 0** si les dues **cadenes** són **iguals**, o un **valor numèric menor** o **major** de **zero** en **cas contrari**. **No diferencia** entre **majúscules** i **minúscules**.

```
$cadena1='dimarts';
$cadena2='Dimarts';
echo strcasecmp($cadena1, $cadena2);
```

Retorna **0**, són iguals

FARCIT DE CADENES AMB STR_PAD

Aquesta funció realitza un **farciment** de **cadenes** ampliant la cadena fins a una **longitud especificada** i **omplint** amb el **caràcter** o caràcters **especificats** fins aquesta longitud. La sintaxi és:

```
str_pad ($cadena,$novaLongitud $opcCarFarcit,$opcTipusDeFarcit)
```

- `opcCarFarcit` és opcional i indica el **caràcter** o **caràcters** de **farciment** que s'empraran. **Si no** s'especifica, es prendrà **l'espai** en **blanc** com a **caràcter** de **farciment**.

- `opcTipusDeFarcit` és opcional i indica **com s'omplirà** fins a aconseguir la nova longitud amb els següents valors:
  - per la dreta: `STR_PAD_RIGHT`
  - per l'esquerra: `STR_PAD_LEFT`
  - per ambdós costats:`STR_PAD_BOTH`

  Si no s'especifica per defecte s'omplirà utilitzant `STR_PAD_RIGHT` (per la dreta).

Exemple:

```
<?php
	$cadena = 'iessimarro.com';
	echo "La cadena és: $cadena <br>";
	echo '<p>'.str_pad($cadena, 28, '*').'</p>';
	echo '<p>'.str_pad($cadena, 48, ' Xàtiva').'</ p>';
	echo '<p>'.str_pad($cadena, 28, '-', STR_PAD_LEFT).'</p>';
	echo '<p>'.str_pad($cadena, 28, '*', STR_PAD_RIGHT).'</p>';
	echo '<p>'.str_pad($cadena, 29, '@', STR_PAD_BOTH).'</p>';
?>
```

El resultat esperat és:

La cadena és: iessimarro.com

iessimarro.com*************

iessimarro.com Xàtiva Xàtiva Xàtiva Xàtiva X

----------iessimarro.com

iessimarro.com*************

@@@@@@@iessimarro.com@@@@@@@@

En aquest darrer cas la longitud de la cadena és de 14 caràcters. Si li posem que la nova longitud és 29 i que òmpliga a banda i banda comença col·locant un caràcter a dreta i un altre a esquerra, però després de col·locar 7 només li queda 1 per col·locar i aquest caràcter queda a la dreta.