



PHP 09

Més funcions de cadenes

13) Més funcions de cadenes PHP

str_replace

- 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.

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

```
str_replace ($agulla,  
            $reemplaçament,  
            $paller)
```

- 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
```

```
?>
```

Variable **\$text** manté el seu valor

- Eixida:
On deia blau ara diu negre

On deia blanc ara diu negre

- **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);
```
- **Opcionalment** es pot passar un paràmetre addicional:
 - 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;
```

```
?>
```

- Eixida:

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

- **strtolower i strtoupper** : retornen una cadena de caràcters transformada en **minúscules** o **majúscules** respectivament.

- **Exemple 03:**

```
<?php
```

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

```
?>
```

- **Eixida:**
això és una cadena de caràcters
AIXÒ ÉS UNA CADENA DE CARÀCTERS

- Es poden utilitzar:
 - Per fer **comparacions**.
 - Per **emmagatzemar** dades de manera **uniforme**.
- Per exemple, si es demana una ciutat, diferents usuaris poden introduir:
 - El Genovés, el Genovés o EL GENOVÉS.
- Amb **strtolower** i **strtoupper** podem uniformitzar:
 - Fer que sempre es **mostren o guarden d'una mateixa manera**:
 - Facilitant 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 puguin utilitzar les funcions **mb_**:
 - Cal **instal·lar** un paquet en Ubuntu (per exemple per a la versió 8.3 de php):

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

- **count_chars**: compta 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 els següents:

- **0**: es retornarà un **array amb el valor numèric ascii com a índex** i la **frequència** de cada caràcter ascii com a **valor**.
- **1**: es retornarà un **array amb el valor numèric ascii com a índex** i la **frequència** 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 **frequència** 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> ';
    }
?>
```

- **Eixida:**

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.

- **substr_count(\$cadena,\$subcadena)** : retorna el **nombre de vegades** que apareix la **\$subcadena** dins de la **\$cadena**.

- **Exemple:**

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

- **Eixida:**

Lluís apareix 3 vegades

- **strlen(\$cadena):** Retorna la **longitud o nombre de caràcters** de la cadena.
- **Exemple:**

```
$cadena="hui és divendres"; //accents: x2  
echo strlen($cadena);
```

• **Eixida:**
17

- **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** (0) si no s'especifica **\$opcPosInici** (opcional).
 - 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, "*");
```

Manté els espais
a dreta i esquerra

- **Eixida:**
hui és divendres

- **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,"*");
```

Manté espai
a l'esquerra

- **Eixida:**
hui és divendres ***

- **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, "*");
```

Manté espai
a la dreta

- **Eixida:**
***** hui és divendres**

- **str_repeat(\$cadena, \$nVegades)**: Retorna la **cadena repetida** tantes **vegades** com indique **\$nVegades** sent aquest un nombre sencer.
- **Exemple:**

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

- **Eixida:**
holaholahola

- **strstr(\$cadena, \$des, \$opcBoolean):** Retorna la **subcadena** des de la **primera aparició** de **\$des** (**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** (**exclusa**).
- **Exemple:**

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

• **Exemple:**
és divendres

- **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:
 - **Recomanable** fer ús de les **funcions natives** del llenguatge.
 - Si comparem utilitzant l'**operador ==** es poden obtenir **resultats estranys**.
- Utilitzant l'**operador ===** pot donar **millors resultats**.
- En general tractarem d'usar funcions com **strcmp** previstes específicament per a això.



- `strcmp($cadena1,$cadena2)` : realitza la **comparació segura** de Strings i **retorna** un **valor numèric**.
 - És **sensible a majúscules i minúscules**: dimarts no és igual a Dimarts.
`if (strcmp ($cadena1, $cadena2) == 0) {...}`
- La funció **retorna** un **valor numèric** que pot ser:
 - **0**: quan les dues **cadenes són iguals**. En cas contrari el valor retornat és diferent de zero
 - Un valor numèric **menor que 0** si `$cadena1` és «menor» que `$cadena2`.
 - Un valor numèric **major que 0** si `$cadena1` és «més gran» que `$cadena2`.

- Si volem saber si **dues cadenes són diferents** podem fer servir:

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

- 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**.

- La **comparació** es fa en funció dels **codis numèrics** ASCII de cada caràcter.
- Per **exemple**:
 - La lletra **A** té codi numèric **65**.
 - La lletra **a** codi numèric **97**.
- Els codis numèrics **no permeten una ordenació alfabètica precisa** :
 - **No ordenen correctament** majúscules i minúscules, paraules amb accent, caràcters com la ñ, etc.
- Aquest **no és un bon mitjà per ordenar alfabèticament paraules**:
 - Només té una **utilitat limitada**.

```
<?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>';
    }
?>
```

- **Eixida:**
Segons == les dues cadenes són iguals

```
<?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>';
    }
?>
```

- **Eixida:**

Segons === les dues cadenes NO són iguals

```
<?php
    $cadena1='1e3'; $cadena2='1000';
    if(strcmp($cadena1,$cadena2) == 0) {
        echo 'Segons strcmp les dues cadenes són iguals <br>';
    } else {
        echo 'Segons strcmp les dues cadenes NO són iguals <br>';
    }
?>
```

- **Eixida:**

Segons strcmp les dues cadenes NO són iguals.

- **strcasecmp(\$cadena1, \$cadena2)**: Retorna com a resultat:
 - **0** si les dues cadenes **són iguals**.
 - Un valor numèric **menor o major de zero** en **cas contrari**.
- No diferencia entre majúscules i minúscules.

- **Exemple:**

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

- **Eixida: 0**

FARCIT DE CADENES AMB STR_PAD

- **str_pad**: 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 la següent:

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


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'; //longitud 14  
echo "La cadena és: $cadena <br>";  
echo '<p>'.str_pad($cadena, 28, '*').'</p>';  
echo '<p>'.str_pad($cadena, 48, ' Xàtiva').'</p>';
```

```
?>
```

- **Eixida:**

```
La cadena és: iessimarro.com  
iessimarro.com*****  
iessimarro.com Xàtiva Xàtiva Xàtiva Xàtiva X
```

- **Exemple:**

```
<?php
```

```
$cadena = 'iessimarro.com'; //longitud 14  
echo '<p>'.str_pad($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>';
```

```
?>
```

- **Eixida:**

-----iessimarro.com

iessimarro.com*****

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

Quantitat imparella,
en fica **1 més per la dreta**