

## Tema 3. CADENES

El tractament de cadenes és de gran importància en l'ús de PHP, donat que, a la majoria de casos, l'objectiu final del processament d'un fitxer està relacionat amb la generació de documents (gairebé sempre documents HTML). Per això existeix un ampli conjunt de funcions per al manegament de cadenes agrupades al voltant del tipus de funció que realitzen.

### 3.1. DELIMITACIÓ DE CADENES

Una cadena està formada per zero o més caràcters tancats entre dobles cometes "" o entre cometes simples ' '.

Quan s'utilitzen cometes dobles, es poden incloure a dins de la cadena noms de variables o caràcters especials que seran avaluats (es substitueix pel seu valor) a l'hora de mostrar la informació, cosa que no succeeix si es tanca el nom de la variable dins d'una cadena tancada entre cometes senzilles.

```
<?php
```

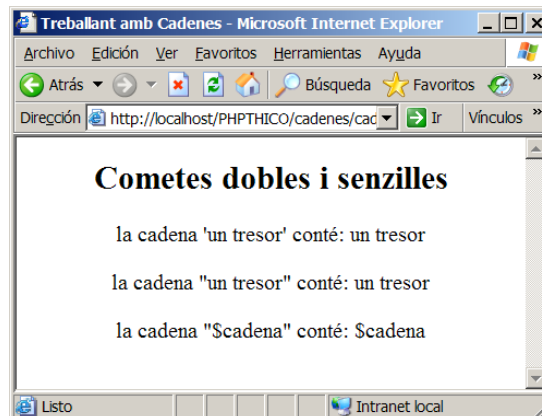
```
  $cadena = 'un tresor';
```

```
  echo "la cadena '$cadena' conté: $cadena<br><br>";
```

```
  echo "la cadena \"$cadena\" conté: $cadena<br><br>";
```

```
  echo 'la cadena "$cadena" conté: $cadena<br><br>';
```

```
?>
```

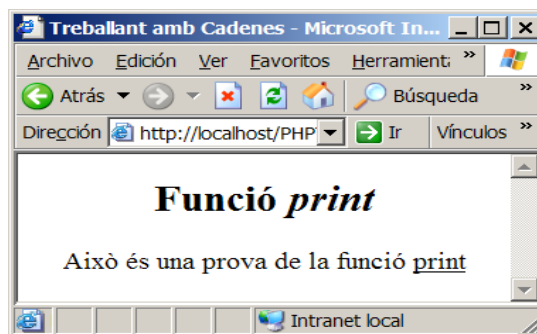


## 3.2. VISUALITZACIÓ DE CADENES

Començarem amb les funcions que ens permeten visualitzar i formatar cadenes de caràcters, permetent la gestió de la sortida de les dades per la pantalla:

- **echo(cadena), print(cadena)**: No són funcions pròpiament dites, sinó construccions del llenguatge. Ambdues mostren informació per la sortida estàndard; no suporten cap atribut de format de sortida i només admeten una única cadena com a argument en la seva crida. Quan es vol passar més d'un argument a qualsevol d'aquestes funcions, s'han d'utilitzar cometes.

```
<?php
$funcio = "print";
print "Això és una prova \n de la funció <U>$funcio</U>";
?>
```



- **printf(format [, arguments])**: Imprimeix informació per la sortida estàndard suportant diferents formats de sortida. Admet múltiples tipus d'arguments a visualitzar.

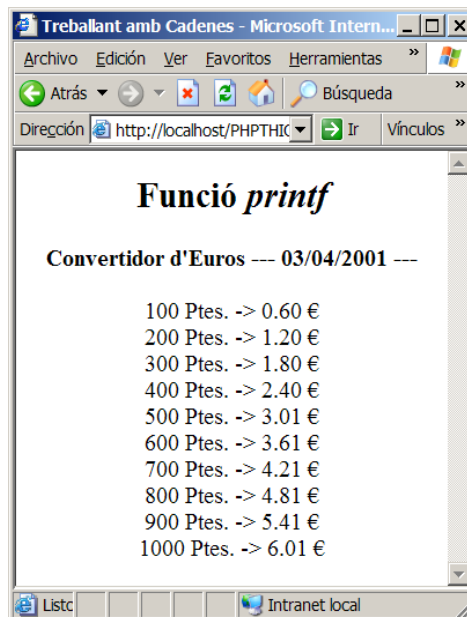
Imprimeix una cadena l'estructura de la qual depen del format descrit a l'argument *format*. Aquesta cadena està formada per un conjunt de caràcters, alguns dels quals es mostrarà directament, mentre que altres, els que van precedits pel caràcter % coneguts com a *especificacions de conversió*, són utilitzats per a formatar la informació.

Cada *especificació de conversió* es compon dels següents elements:

1. Un *caràcter de farcit* – opcional – que s'utilitza per a farcir i ajustar el resultat fins a aconseguir el tamany de cadena correcte. Per exemple, l'espai en blanc per a caràcters (valor per defecte) o el zero per a valors numèrics.
2. Un *caràcter d'alineament* – opcional – per a indicar si el resultat ha d'alinear-se a l'esquerra (caràcter -) o a la dreta (valor per defecte).
3. Un *indicador de tamany* – opcional – que indica el tamany mínim de caràcters que ocuparà l'argument després de la conversió.
4. Un *indicador de precisió* – opcional – format per un punt seguit del nombre de díigits decimals que hauran de mostrar els nombres en punt flotant. No tenen efecte amb altres tipus de dades.
5. Finalment, un *identificador de tipus de dades*, que especifica com s'haurà de tractar la dada. Els tipus disponibles són:

Símbol	Tipus de dades	Tractament
%	caràcter %	caràcter %
d	decimals	com a enter decimal
b	binaris	com a enter binari
o	octals	com a enter
x	hexadecimal (lletres minúscules)	com a enter
X	hexadecimal (lletres majúscules)	com a enter
c	caràcters ASCII	com caràcter
f	punt flotant	com a <i>float</i> o decimal
e	punt flotant	com a <i>float</i> o decimal
s	cadenes	com a <i>string</i>

```
<?php
$euro=166.386;
printf ("%s--- %02d/%02d/%04d ---","<B>Convertidor d'Euros ", 3, 4, 2001);
echo "</B><BR><BR>";
for ($i=100;$i<1100;$i+=100)
    printf("%4d Ptes. -> %02.2f €%s",$i,$i/$euro,"<BR>");
?>
```



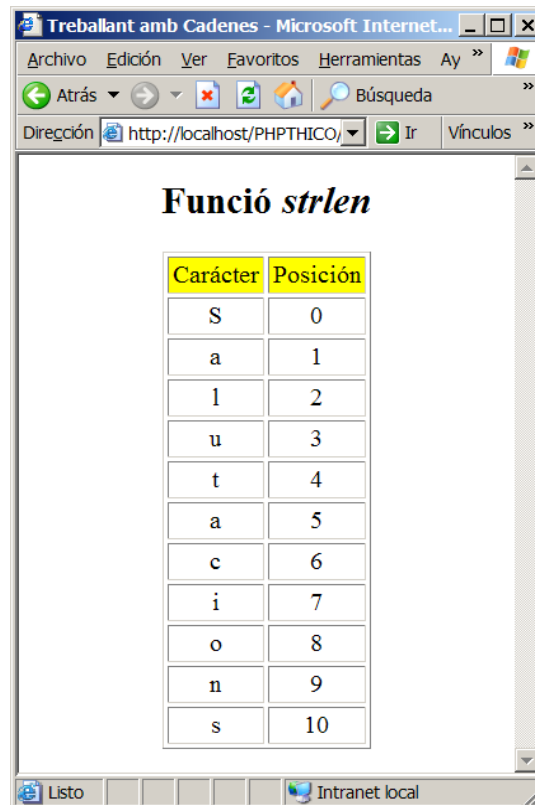
- **sprintf(format [, arguments]):** És molt semblant a **printf()**, però es diferencia en que retorna una cadena de caràcters.

### 3.3. ACCÉS AL CONTINGUT

Es pot accedir a cadascun dels caràcters que componen una cadena fent referència a la posició que ocupen dins de la cadena, d'igual forma a com es referencien els diferents components d'una *matriu* o *array*. Per això, es fa indispensable conèixer el tamany de la cadena a recórrer.

- **strlen(cadena)**: Retorna la longitud de la cadena passada com a argument.

```
<?php
$cadena="Salutacions";
echo "<TABLE BORDER='1'CELLPADDING='2' CELLSPACING='2'>\n";
echo "<TR BGCOLOR='yellow'><TD>Caràcter</TD><TD>Posició</TD></TR>\n";
for($i=0;$i<strlen($cadena);$i++)
    echo "<TR ALIGN='center'><TD>".$cadena[$i]."</TD><TD>".$i."</TD></TR>\n";
echo "</TABLE>\n";
?>
```



### 3.4. CERCA EN CADENES

Una de les operacions més habituals sobre les cadenes és la cerca de caràcters i subcadenes components, per al seu posterior tractament. Dins d'aquest tipus de funcions destaquen les següents:

- **strstr(cadena, cadCerca), strchr(cadena, cadCerca):** Cerca l'aparició d'una cadena dins de l'altra i retorna la subcadena que es troba entre la primera aparició de la cadena cercada (incloent-la) fins al final de la cadena on es realitza la cerca. En cas de no trobar la cadena, retorna una cadena buida. La cerca diferencia entre majúscules i minúscules.
- **strrchr(cadena, cadCerca):** Cerca l'aparició d'un caràcter (encara que s'utilitza una cadena de cerca només té en compte el primer caràcter) en una cadena i retorna la subcadena compresa entre la última aparició del caràcter (inclòs) fins al final de la cadena on es realitza la cerca. En cas de no trobar el caràcter de la cadena, es retorna una cadena buida. La cerca diferencia entre majúscules i minúscules. Si l'element a cercar no és una cadena, es converteix a enter i s'aplica com el valor ordinal d'un caràcter.

```
<?php
```

```
$cadena="Saludos para todos...";
```

```
$car1="do";
```

```
$car2="pc";
```

```
echo "<B>cadena:</B> $cadena <br>";
```

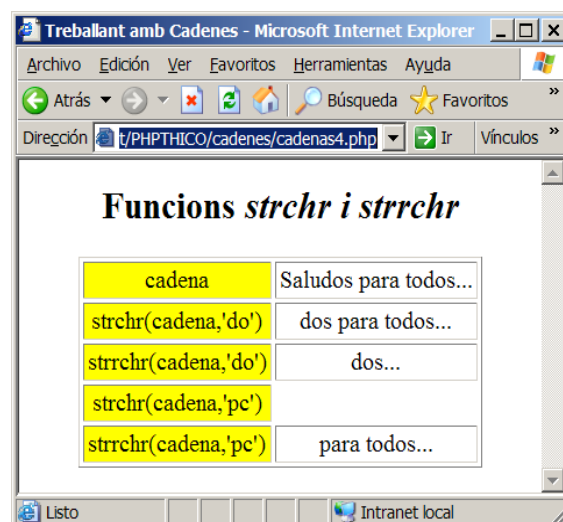
```
echo "<B>strchr(cadena,'$car1'):</B> ".strchr($cadena,$car1)."</BR>";
```

```
echo "<B>strrchr(cadena,'$car1'):</B> ".strrchr($cadena,$car1)."</BR>";
```

```
echo "<B>strchr(cadena,'$car2'):</B> ".strchr($cadena,$car2)."</BR>";
```

```
echo "<B>strrchr(cadena,'$car2'):</B> ".strrchr($cadena,$car2)."</BR>";
```

```
?>
```



- **strstr(cadena, cadCerca):** El comportament d'aquesta funció és igual al de la funció **strstr()**, excepte que aquesta no diferencia entre minúscules i majúscules.

```
<?php
```

```
$cadena="Saludos para todos...";
```

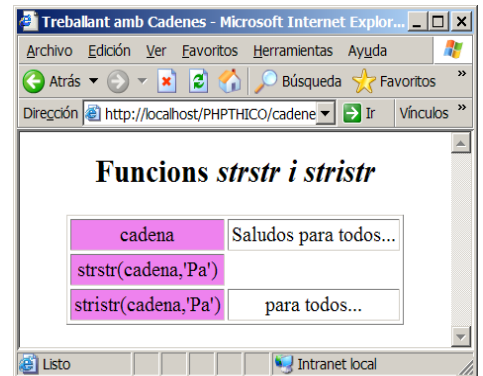
```
$car="Pa";
```

```
echo "<B>cadena:</B> $cadena</BR>";
```

```
echo "<B>strstr(cadena,'$car'):</B> ".strstr($cadena,$car)."</BR>";
```

```
echo "<B>strstr(cadena,'$car'):</B> ".strstr($cadena,$car)."</BR>";
```

```
?>
```



- **strpos(cad1, cad2 [, desplaçament]):** Troba la primera posició d'aparició d'una cadena a partir d'una certa posició (si no s'indica, el valor per defecte per al desplaçament és 0). La cerca diferencia entre majúscules i minúscules.
- **strrpos(cadena, caràcter):** Retorna la posició de la última aparició d'un caràcter determinat en una cadena. En cas de no trobar el caràcter a la cadena retorna *false*. La cerca diferencia entre majúscules i minúscules.

```
<?php
```

```
$cadena="Saludos para todos...";
```

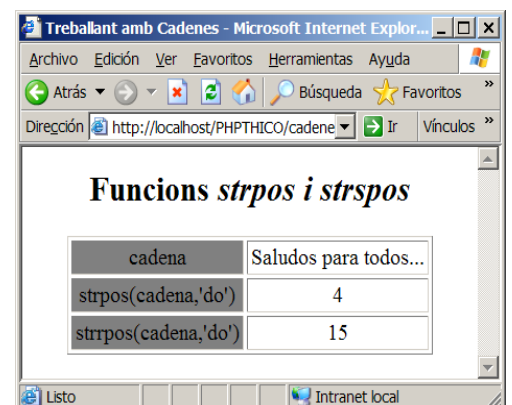
```
$car="do";
```

```
echo "<B>cadena:</B> $cadena</BR>";
```

```
echo "<B>strpos(cadena,'$car'):</B> ".strpos($cadena,$car)."</BR>";
```

```
echo "<B>strrpos(cadena,'$car'):</B> ".strrpos($cadena,$car)."</BR>";
```

```
?>
```



- **strspn(cadena, màscara)**: S'obté la longitud de la subcadena més llarga que està formada només per caràcters continguts en una màscara. Una vegada trobat un caràcter que no estigui a la màscara s'abandona la cerca. La cerca diferencia entre majúscules i minúscules.
- **strcspn(cadena, màscara)**: S'obté la longitud de la subcadena que està formada només per caràcters no continguts en una màscara (aquesta funció és la complementària a **strspn()**). Una vegada trobat un caràcter que estigui contingut a la màscara, s'abandona la cerca. La cerca diferencia entre majúscules i minúscules.

```
<?php
```

```
$cadena="Saludos para todos...";
```

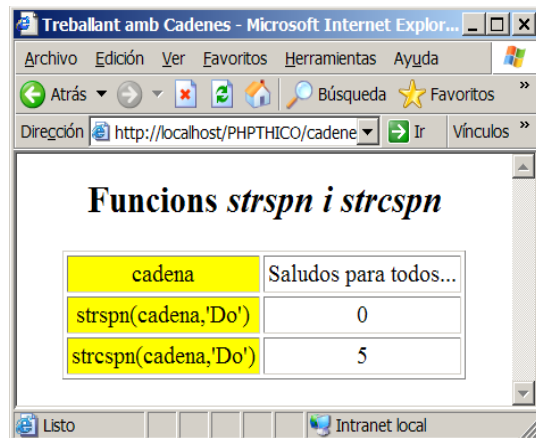
```
$car="Do";
```

```
echo "<B>cadena: </B>$cadena</BR>";
```

```
echo "<B>strspn(cadena,'$car'):</B> ".strspn($cadena,$car)."</BR>";
```

```
echo "<B>strcspn(cadena,'$car'):</B> ".strcspn($cadena,$car)."</BR>";
```

```
?>
```



### 3.5. COMPARACIÓ DE CADENES

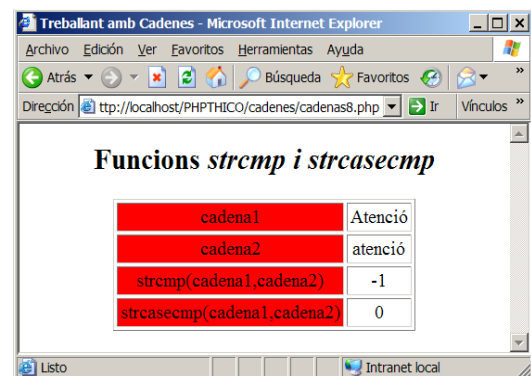
Una altra operació comú amb cadenes és comparar-les per saber quina és més gran. Dins d'aquests tipus de funció destaquen les següents:

- **strcmp(cad1, cad2)**: Compara dos cadenes i retorna un valor:
  - menor que 0 si la segona cadena és més gran que la primera
  - major que 0 en la segona cadena és més petita que la primera
  - igual a 0 si ambdues cadenes són iguals.

La comparació distingeix entre majúscules i minúscules.

- **strcasecmp(cad1, cad2)**: Es comporta igual que **strcmp()**, excepte que no diferencia entre majúscules i minúscules.

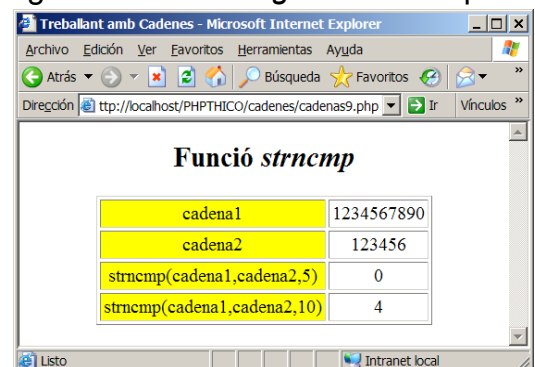
```
<?php
$cad1="Atenció";
$cad2="atenció";
echo "cadena1: $cad1<br>";
echo "cadena2: $cad2<br>";
echo "strcmp(cadena1,cadena2): ".strcmp($cad1,$cad2)."<br>";
echo "strcasecmp(cadena1,cadena2): ".strcasecmp($cad1,$cad2)."<br>";
?>
```



- **strncmp(cad1, cad2, long)**: Funciona com **strcmp()**, només que permet comparar els *long* primers caràcters de dos cadenes. Si alguna cadena és menor que el nombre de caràcters a comparar, s'utilitzarà la seva longitud com a *long* de la comparació. Diferencia entre majúscules i minúscules.

```
<?php
$cad1="1234567890";
$cad2="123456";

$long1=5;
$long2=10;
echo "<B>cadena1:</B> $cad1</BR>";
echo "<B>cadena2:</B> $cad2</BR>";
echo "<B>strncmp(cadena1,cadena2,$long1):</B> ".strncmp($cad1,$cad2,$long1).
"</BR>";
echo "<B>strncmp(cadena1,cadena2,$long2):</B> ".strncmp($cad1,$cad2,$long2).
"</BR>";
?>
```





- **strnatcmp**(cad1, cad2): Es comporta igual que **strcmp()**, excepte que utilitza una comparació "natural" de les cadenes alfanumèriques. Distingeix entre majúscules i minúscules.
- **strnatcasecmp**(cad1, cad2): Es comporta igual que **strnatcmp()** excepte en que no diferencia entre majúscules i minúscules.

```
<?php
```

```
$cad1="10000 Pts";
```

```
$cad2="2500 Pts";
```

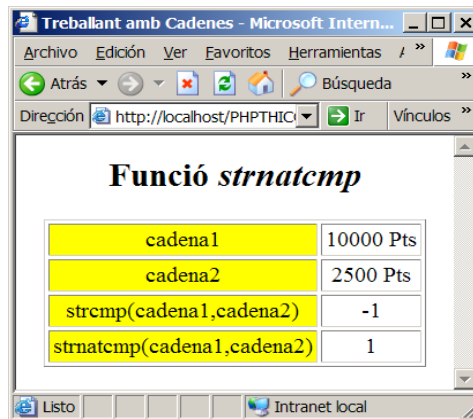
```
echo "<B>cadena1:</B> $cad1</BR>";
```

```
echo "<B>cadena2:</B> $cad2</BR>";
```

```
echo "<B>strcmp(cadena1,cadena2):</B> ".strcmp($cad1,$cad2)."</BR>";
```

```
echo "<B>strnatcmp(cadena1,cadena2):</B> ".strnatcmp($cad1,$cad2)."</BR>";
```

```
?>
```



### 3.6. OPERAR AMB SUBCADENES

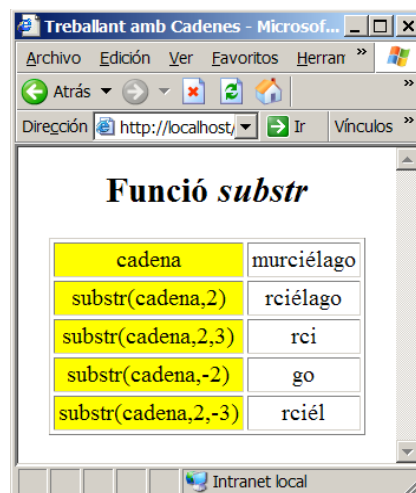
No sempre interessen tots els caràcters que componen una cadena, sinó que només interessa treballar amb un conjunt. Per a això existeix un conjunt de funcions preparades per a extreure subcadena d'una cadena original. Entre les més utilitzades estan:

- **substr(cadena, inici [, tamany]):** Retorna una subcadena que es troba a partir d'una posició donada i arriba fins al final de la cadena original. Es pot decidir, de forma opcional, el tamany de la subcadena que es vol recuperar.

Els arguments enters poden ser nombres negatius de manera que:

- Una posició d'inici amb valor negatiu significa que s'ha de començar des del final de la cadena.
- Un tamany amb valor negatiu indica quants caràcters del final de la cadena no es tindran en compte.

```
<?php
$cadena="murciélago";
$ini1 = 2;
$ini2 = -2;
$tam1 = 3;
$tam2 = -3;
echo "<B>cadena:</B> $cadena</BR>";
echo "<B>substr(cadena,$ini1):</B> ".substr($cadena,$ini1)."</BR>";
echo "<B>substr(cadena,$ini1,$tam1):</B> ".substr($cadena,$ini1,$tam1)."</BR>";
echo "<B>substr(cadena,$ini2):</B> ".substr($cadena,$ini2)."</BR>";
echo "<B>substr(cadena,$ini1,$tam2):</B> ".substr($cadena,$ini1,$tam2)."</BR>";
?>
```



- **substr\_replace(cad1, cad2, inici [, tamany]):** Retorna una cadena que és resultat de la substitució de part del contingut d'una cadena original (indicat per una posició d'inici i opcionalment per un tamany) pel contingut de l'altra cadena. La cadena original no pateix cap modificació. Distingeix entre majúscules i minúscules.

Els arguments enters poden ser nombres negatius de manera que:

- Una posició d'inici amb valor negatiu significa que s'ha de començar des del final de la cadena.
- Un tamany amb valor negatiu indica quants caràcters del final de la cadena no es tindran en compte.

```
<?php
```

```
$cad1="murciélago";
```

```
$cad2="123";
```

```
$ini1 = 2;
```

```
$ini2 = -2;
```

```
$tam1 = 3;
```

```
$tam2 = -3;
```

```
echo "<B>cadena1:</B> $cad1</BR>";
```

```
echo "<B>cadena2:</B> $cad2</BR>";
```

```
echo "<B>substr_replace(cadena1,cadena2,$ini1):</B> "
```

```
.substr_replace($cad1,$cad2,$ini1)."</BR>";
```

```
echo "<B>substr_replace(cadena1,cadena2,$ini1,$tam1):</B> "
```

```
.substr_replace($cad1,$cad2,$ini1,$tam1)."</BR>";
```

```
echo "<B>substr_replace(cadena1,cadena2,$ini2):</B> "
```

```
.substr_replace($cad1,$cad2,$ini2)."</BR>";
```

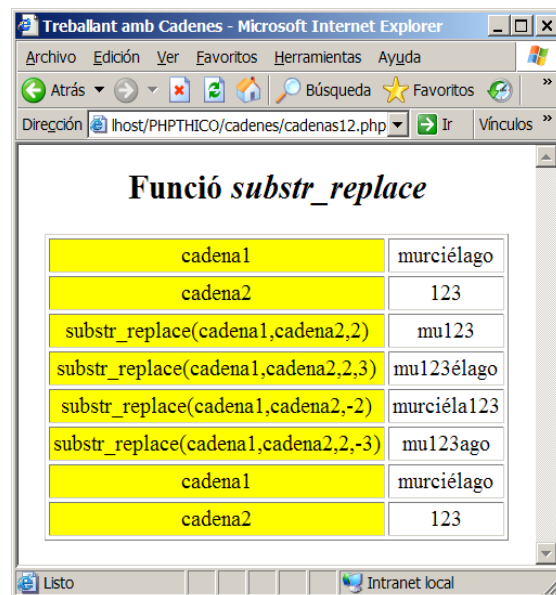
```
echo "<B>substr_replace(cadena1,cadena2,$ini1,$tam2):</B> "
```

```
.substr_replace($cad1,$cad2,$ini1,$tam2)."</BR>";
```

```
echo "<B>cadena1:</B> $cad1</BR>";
```

```
echo "<B>cadena2:</B> $cad2</BR>";
```

```
?>
```



## 3.7. MODIFICACIÓ DEL CONTINGUT

### 3.7.1. NETEJA DE CADENES

- **chop(cadena):** Retorna la cadena sense els espais en blanc i caràcters de fi de línia que es troben al final de la cadena.
- **rtrim(cadena):** Es comporta igual que **chop()**.
- **ltrim(cadena):** Elimina espais en blanc que es troben al principi d'una cadena de text. Rep com a paràmetre una cadena de caràcters i retorna la mateixa cadena però sense els espais inicials.
- **trim(cadena):** Elimina els espais en blanc que es troben al principi i al final d'una cadena de text. Rep com a argument una cadena de caràcters i retorna com a resultat aquesta cadena sense els espais inicials i finals.

```
<?php
```

```
// 6 espais per davant i 4 per darrera
```

```
$cadena="  abcdefghi  ";
```

```
echo "<B>cadena:</B> $cadena ".strlen($cadena)."<BR>";
```

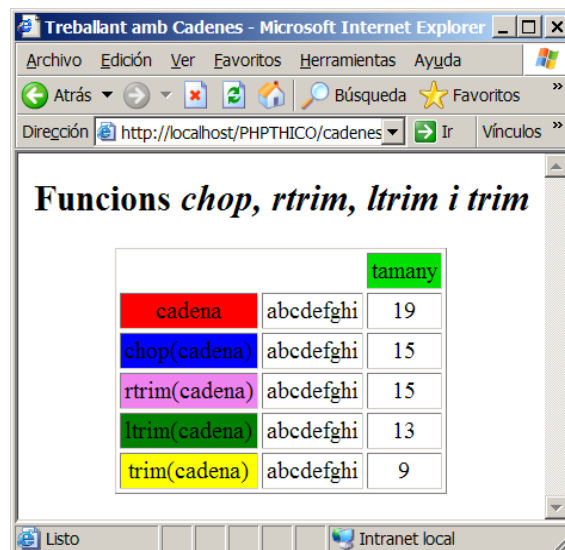
```
echo "<B>chop(cadena):</B>".chop($cadena)." ".strlen(chop($cadena))."<BR>";
```

```
echo "<B>rtrim(cadena):</B>".rtrim($cadena)." ".strlen(rtrim($cadena))."<BR>";
```

```
echo "<B>ltrim(cadena):</B>".ltrim($cadena)." ".strlen(ltrim($cadena))."<BR>";
```

```
echo "<B>trim(cadena):</B>".trim($cadena)." ".strlen(trim($cadena))."<BR>";
```

```
?>
```

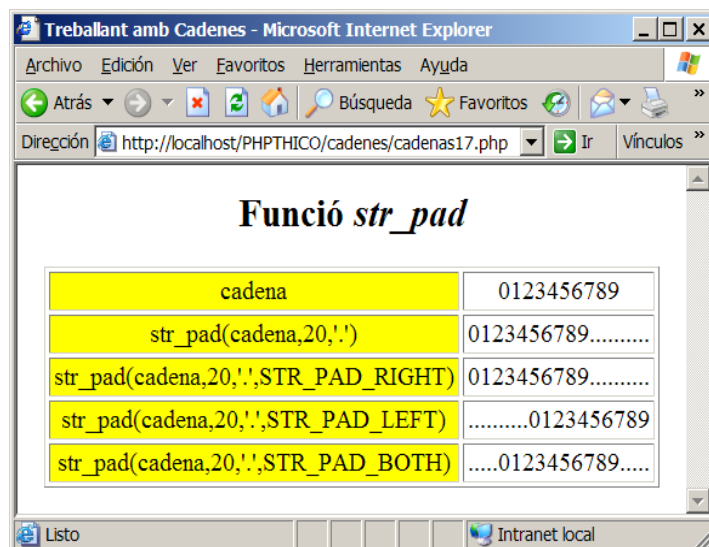


### 3.7.2. OMPLIR CADENES

- **str\_pad(cadena, long)**: Omple una cadena amb un caràcter de farcit (per defecte, és l'espai en blanc) fins que la cadena resultant tingui la longitud desitjada. Opcionalment es pot indicar el mode d'ompliment amb els següents valors:
  - x **STR\_PAD\_RIGHT**: Omple per la dreta (opció per defecte).
  - x **STR\_PAD\_LEFT**: Omple per l'esquerra.
  - x **STR\_PAD\_BOTH**: Intenta omplir amb el mateix nombre de caràcters per la dreta i per l'esquerra.

```
<?php
$cadena="0123456789";
$long = 20;
$car = ".";
echo "<B>cadena:</B> $cadena<BR>";
echo "<B>str_pad(cadena,$long,$car):</B> ".str_pad($cadena,$long,$car).
" <BR>";

echo "<B>str_pad(cadena,$long,$car,STR_PAD_RIGHT):</B>";
echo str_pad($cadena,$long,$car,STR_PAD_RIGHT)." <BR>";
echo "<B>str_pad(cadena,$long,$car,STR_PAD_LEFT):</B>";
echo str_pad($cadena,$long,$car,STR_PAD_LEFT)." <BR>";
echo "<B>str_pad(cadena,$long,$car,STR_PAD_BOTH):</B>";
echo str_pad($cadena,$long,$car,STR_PAD_BOTH)." <BR>";
?>
```



### 3.7.3. CONVERSIÓ ENTRE MAJÚSCULES I MINÚSCULES

- **strtolower(cadena):** Converteix una cadena de caràcters a minúscules. Rep com a argument una cadena i retorna la mateixa cadena amb els caràcters alfabètics en minúscules.
- **strtoupper(cadena):** Funciona de forma contrària a la funció anterior. Converteix una cadena de caràcters a majúscules.
- **ucfirst(cadena):** S'encarrega de convertir a majúscules el primer caràcter d'una cadena de caràcters, sempre que aquest sigui alfabètic.
- **ucwords(cadena):** Es comporta igual que la funció anterior, només que aquesta converteix a majúscules el primer caràcter de cada paraula que compona la cadena, sempre i quan siguin caràcters alfanumèrics.

```
<?php
```

```
$frase="saludos PARA todos...";
echo "<B>frase:</B> $frase<BR>";
echo "<B>strtoupper(frase):</B> ".strtoupper($frase)."<BR>";
echo "<B>strtolower(frase):</B> ".strtolower($frase)."<BR>";
echo "<B>ucfirst(frase):</B> ".ucfirst($frase)."<BR>";
echo "<B>ucwords(frase):</B> ".ucwords($frase)."<BR>";
```

```
?>
```

