

# Le funzioni native

## Funzioni per le stringhe

Mettiamo di voler sapere da quanti caratteri è composta la frase: "La mia casa ha due porte e due finestre".

Utilizzeremo la funzione `strlen`. Vediamola nella documentazione -> [www.php.net/strlen](http://www.php.net/strlen)

La descrizione è la seguente:

```
int strlen ( string $string )
```

Il formato è:

- Cosa ritorna la funzione
- Il nome della funzione
- Gli argomenti della funzione

Nel nostro caso `strlen` ritorna un INT (ovvero il numero di caratteri), e passa un unico argomento ovvero la stringa stessa.

Esempio di utilizzo:

```
<?php
```

```
$stringa = "La mia casa ha due porte e due finestre";
```

```
$num = strlen($stringa);
```

```
echo $num;
```

```
// oppure
```

```
echo strlen($stringa);
```

```
?>
```

## Funzioni per array

Vi siete fatti un po' di pratica con la guida? Guardate la funzione `count` (ormai dovrete sapere come trovarla). Con `count` contiamo gli elementi di un array. Ritorna un INT (il numero di elementi) e passa un array come argomento.

```
<?php
```

```
$frutta = array("mela", "pera", "banana", "pesca");
```

```
$num = count($frutta); // $num avrà valore 4
```

```
?>
```

E se volessimo ordinare il nostro array in ordine alfabetico?

Utilizzeremo la funzione `sort`

```
<?php
$frutta = array("mela", "pera", "banana", "pesca");
sort($frutta);
?>
```

Il nostro array sarà modificato ed avrà assunto ordine alfabetico.

### Funzioni matematiche

aprofitto di questo paragrafo per mostrarvi come le funzioni possano essere nidificate

```
<?php
$x = 3.57; // Vediamo come ricavare l'intero di un numero con la funzione
intval
echo intval($x); // stampa 3

// vediamo come calcolare la radice quadrata di un numero con sqrt
echo sqrt(16); // stampa 4

// mettiamo che desidero l'intero della radice quadrata di un numero //
posso fare così:
$x = 10;
$y = sqrt($x);
$z = intval($y); // $z vale 3

// oppure posso nidificare le funzioni
$z = intval(sqrt($x));
?>
```

Immagino che a questo punto vi rendiate conto che c'è una funzione per tutto, basta saperla cercare. L'unico insieme di funzioni per il quale ritengo di dover fare un breve approfondimento, sono le funzioni legate al tempo, in quanto necessitano di qualche spiegazione preliminare.

### Le funzioni di tempo

Il concetto fondamentale nella manipolazione del tempo in PHP (e non solo in PHP) è lo UNIX `TIMESTAMP` o semplicemente `timestamp`. Il `timestamp` rappresenta il numero di secondi trascorsi dal 1 gennaio 1970. Nel momento in cui sto scrivendo il `timestamp` è:

1225458927

Vi può sembrare bizzarro rappresentare il tempo in questo modo, in realtà è molto comodo in particolare quando si tratta di fare dei confronti o dei calcoli.

Per noi è molto semplice ad esempio stabilire se viene prima il 6 febbraio 2006, il 10.4.08 o il 7 ago 1997, ma è ovvio che in un codice è notevolmente più semplice confrontare i tre timestamp. Al contrario non possiamo mettere nel nostro guestbook una cosa del genere:

“questo messaggio è stato lasciato il 1225458927”.

Schematicamente: è meglio memorizzare una data in forma di timestamp in modo da poter fare calcoli e confronti, ma poi dovremo anche saper trasformare questo numero in un output “umano”.

Se vogliamo conoscere il timestamp attuale ci basterà utilizzare la funzione time

```
echo time();
```