

Conversione dei tipi in PHP: il cast.

Vediamo un esempio semplice ma efficace:

```
<?php
echo '5' * '3';
?>
```

Abbiamo chiesto al PHP di mostrarci il risultato di una moltiplicazione tra due stringhe. Questa operazione non ha alcun senso eppure se lanciamo lo script otteniamo come risultato "15" ovvero il prodotto tra dei numeri contenuti all'interno della stringa. Per capire cosa sia successo vediamo i singoli passi che hanno portato a questo risultato:

- il PHP deve risolvere l'espressione '5' * '3';
- poiché si tratta di un operatore algebrico (la moltiplicazione *) il PHP si aspetta di trovare due numeri;
- entrambi i fattori sono però delle stringhe ('5' e '3') allora il PHP prima di procedere deve convertirle in numeri. Questa operazione di conversione di tipo (da tipo stringa a tipo numero) viene detta cast. La stringa '5' diventa il numero 5 e la stringa '3' diventa il numero 3;
- il PHP adesso può effettuare il prodotto tra i numeri 5 e 3;
- viene mostrato il numero 15.

Il PHP quindi effettua in maniera trasparente, ovvero senza una esplicita indicazione, delle conversioni di tipo ogni qualvolta deve effettuare una operazione su un tipo di dati diverso da quello atteso. Ovviamente questa conversione può portare a risultati privi di significato: "casa" * "3" sarà uguale a 0 poiché la stringa "casa" non sottintende un valore numerico e perciò sarà convertita nel numero 0.

Vediamo ora un esempio di conversione da numero a stringa:

```
<?php
echo 'un numero a caso: ' . 43;
?>
```

E' evidente che in questo caso il numero 43 verrà convertito in stringa prima di essere accodato attraverso l'operatore "." che lavora solo su stringhe.

Il caso più interessante di conversione di tipo però è quello che da come risultato il tipo booleano. Come accennato nella precedente lezione a proposito degli operatori logici, spesso capita che il PHP debba eseguire un'operazione e si aspetta che i dati su cui lavorare siano di tipo booleano (come nella condizione del costrutto if) ma invece gli vengono forniti dati di altro tipo (interi, stringhe, array).

PHP procederà quindi ad una conversione che seguirà queste regole:

- Tutti i numeri sono convertiti in true tranne lo 0
- Tutte le stringhe sono convertite in true tranne le stringhe vuote ("") e quelle che valgono "0"
- Tutti gli array sono convertiti in true tranne quelli vuoti (senza elementi)

Come sarà chiarito dal seguente esempio, l'uso di diversi tipi di dati e la loro implicita conversione in booleano permette di scrivere codice molto compatto ed efficace.

```
<?php
$array = array();

// conversione implicita da array a booleano
if($array)
{
    echo "\'array $array ha almeno un elemento <br>";
}
else
{
    echo "\'array $array è vuoto <br>";
}

$stringa = 'ciao';

// conversione implicita da stringa a booleano
if($stringa)
{
    echo 'la stringa $stringa vale ' , $stringa;
}
else
{
    echo 'la stringa $stringa è vuota';
}

$numero = 123;

// conversione implicita da numero a booleano
if($numero)
{
    echo 'la variabile $numero è diversa da 0';
}
else
{
```

```
    echo 'la variabile $numero è uguale a 0';  
}  
?>
```