

## Tipi di dato base

Per i dettagli si veda il manuale: <http://www.php.net/manual/en/language.types.php>

### Boolean

Ammette i due valori vero e falso espressi con le costanti TRUE e FALSE, che possono essere scritti indifferentemente in maiuscole e minuscole essendo case insensitive.

### Integer

Ammette i valori interi.

Le costanti intere possono essere scritte nei formati decimali, ottali, esadecimali:

```
decimal      : [1-9][0-9]*  
              | 0  
  
hexadecimal  : 0[xX][0-9a-fA-F]+  
  
octal        : 0[0-7]+  
  
integer      : [+]?decimal  
              | [+]?hexadecimal  
              | [+]?octal
```

### Float

Ammette i valori con parte decimale.

Le costanti float possono essere espresse nel formato xx.xxx o in forma mantisse ed esponente:

```
LNUM          [0-9]+  
DNUM          ([0-9]*[\.]{LNUM}) | ({LNUM}[\.][0-9]*)  
EXPONENT_DNUM [+]?(({LNUM} | {DNUM}) [eE][+]? {LNUM})
```

### String

Ammette valori definiti come sequenze di caratteri.

Le costanti string possono essere rappresentate nei seguenti formati:

#### Virgolette semplici

Una sequenza di caratteri racchiusa tra una coppia di apici (virgolette semplici – single quote) rappresenta il valore string che **riporta esattamente quei caratteri qualunque essi siano**:

Esempi:

'stringa', 'stringa più lunga', 'string con caratteri non alfabetici : , ^ , & , \$ , ecc ecc'

Sono previste due eccezioni:

- se devo inserire un apostrofo all'interno di una stringa devo “proteggerlo” con il

backslash in modo che non sia interpretato come terminatore di stringa.

Es: 'gettate \\l'ancora'

- se devo inserire il carattere backslash prima di un apice (ad es prima come ultimo carattere di una stringa) va raddoppiato, negli altri casi non è necessario, ma un doppio backslash \\ viene interpretato come se fosse uno solo.

## Virgolette doppie

Una sequenza di caratteri racchiusa tra una coppia di doppio apice (virgolette doppie – double quote) rappresenta il valore string ottenuto interpretandone il contenuto, cioè sostituendo il valore delle variabili eventualmente presenti e degli eventuali caratteri speciali (il più diffuso è \n per il carattere 'a capo'.

Esempio:

```
<?php
class Prova {
    public $attributo;
}
$numero=3;
$parola='esempio';
$oggetto=new Prova();
$oggetto->attributo=3.5;

echo 'Questa è una stringa semplice';
echo "Questa è una stringa semplice";
echo 'Questa e\' una stringa semplice';
echo "Questa e' una stringa semplice";
echo 'Attenzione alle " (virgolette) !!!';
echo "Attenzione alle \" (virgolette) !!!";
echo 'Casi interessanti: $numero, $parola, {$oggetto->attributo}.\n';
echo "Casi interessanti: $numero, $parola, {$oggetto->attributo}.\n";
echo "Fine degli esempi";
?>
```

Una volta eseguito produce:

```
Questa è una stringa sempliceQuesta è una stringa sempliceQuesta
e' una stringa sempliceQuesta e' una stringa sempliceAttenzione
alle " (virgolette) !!!Attenzione alle " (virgolette) !!!Casi
interessanti: $numero, $parola, {$oggetto->attributo}.\nCasi
interessanti: 3, esempio, 3.5.
Fine degli esempi
```

## Heredoc e Nowdoc

Sono altre due modalità per rappresentare stringhe, per i dettagli rimando al manuale.

## Variabili e tipo di dato

Come già detto il linguaggio php non vincola le variabili a un tipo di dato, ma si adatta al contesto, si veda l'esempio:

```
<?php
$foo = "0"; // $foo is string (ASCII 48)
$foo += 2; // $foo is now an integer (2)
$foo = $foo + 1.3; // $foo is now a float (3.3)
$foo = 5 + "10 Little Piggies"; // $foo is integer (15)
$foo = 5 + "10 Small Pigs"; // $foo is integer (15)
?>
```

Quando si desidera forzare una variabile su uno specifico tipo di dato occorre esplicitare il cast.

## ***Type casting***

Le conversioni di tipo (cast) sono esprimibili con:

- \* (int), (integer) - cast verso integer
- \* (bool), (boolean) - cast verso boolean
- \* (float), (double), (real) - cast verso float
- \* (string) - cast verso string
- \* (array) - cast verso array
- \* (object) - cast verso object
- \* (unset) - cast verso NULL

## **Conversione a boolean**

When converting to boolean, the following values are considered FALSE:

- \* the boolean FALSE itself
- \* the integer 0 (zero)
- \* the float 0.0 (zero)
- \* the empty string, and the string "0"
- \* an array with zero elements
- \* an object with zero member variables (PHP 4 only)
- \* the special type NULL (including unset variables)
- \* SimpleXML objects created from empty tags

Every other value is considered TRUE (including any resource).

## **Conversione a integer**

Sono ammesse conversioni a intero dai tipi:

- da boolean: false->0, true->1
- da float: viene arrotondato all'intero
- da string: viene considerata la parte iniziale della stringa, se risponde a una corretta sintassi per numerico (int o float che sia) allora viene assunto il valore corrispondente negli altri casi viene assunto 0

Non sono ammessi conversioni a intero da altri tipi (risultato imprevedibile)

## Conversione a float

Per cast da string a float si applica la regola citata per integer, negli altri casi viene implicitamente effettuato prima il cast a integer e poi a float.

## Conversione a string

Da boolean: true->'1' false ->""(stringa vuota)

Da numerico: la stringa corrispondente

Esempio:

```
<?php
$intero=3; //int
$parola='esempio'; //string
$float=7.65; //float
$logico='b'>'a'; //bool
echo true.' vero ' . $logico . " ancora vero " . $parola . ' string ' .
$float . ' un float ' . $intero . " un intero\n";
echo "Ripasso: stringa virgolette doppie $parola $logico $intero,
$float \n";
echo 'Ripasso: stringa virgolette semplici $esempio $logico
$intero, $float \n';
?>
```

produce:

```
1 vero 1 ancora vero esempio string 7.65 un float 3 un intero
Ripasso: stringa virgolette doppie esempio 1 3, 7.65
Ripasso: stringa virgolette semplici $esempio $logico $intero, $float \n
```

cc

---

cc Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/3.0/it/> o spedisci una lettera a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.  
Giovanni Ragno – ITIS Belluzzi Bologna 2011