

VII. Funzioni Matematiche BCMath a precisione arbitraria

Introduzione

Per la matematica a precisione arbitraria PHP offre il Binary Calculator che supporta numeri di qualsiasi dimensione e precisione, rappresentati da stringhe;

Requisiti

Dalla versione 4.0.4 del PHP, libbcmath è inclusa nella distribuzione. Non c'è bisogno di altre librerie esterne per questa estensione.

Installazione

Queste funzioni sono disponibili solo se PHP è stato configurato con `--enable-bcmath`. Nel PHP 3, queste funzioni sono disponibili solo se PHP *NON* è stato configurato con `--disable-bcmath`.

La versione per Windows di PHP ha già compilato il supporto per questo modulo. Non occorre caricare alcun modulo addizionale per potere utilizzare queste funzioni.

Configurazione di Runtime

Il comportamento di queste funzioni è influenzato dalle impostazioni di `php.ini`.

Tabella 1. Opzioni di configurazione di BC math

Nome	Default	Modificabile in
<code>bcmath.scale</code>	0	PHP_INI_ALL

Per ulteriori dettagli e definizioni delle costanti `PHP_INI_*` vedere [ini_set\(\)](#).

Breve descrizione dei parametri di configurazione.

`bcmath.scale` [integer](#)

Numero di cifre decimali per tutte le funzioni bcmath. Vedere anche [bcscale\(\)](#).

Tipi di risorse

Questa estensione non definisce alcun tipo di risorsa.

Costanti predefinite

Questa estensione non definisce alcuna costante.

Sommario

[bcadd](#) -- Somma due numeri a precisione arbitraria

[bccomp](#) -- Confronta due numeri a precisione arbitraria

[bcdiv](#) -- Divide due numeri a precisione arbitraria

[bcmod](#) -- Ricava il modulo di un numero a precisione arbitraria

[bcmul](#) -- Moltiplica due numeri a precisione arbitraria

[bcpow](#) -- Effettua l'elevamento a potenza

[bcpowmod](#) -- Effettua l'elevamento a potenza, applicando quindi il modulo

[bcscale](#) -- Imposta il valore di precisione di default per tutte le funzioni matematiche BCMath

[bcsqrt](#) -- Ottiene la radice quadrata di un numero a precisione arbitraria

[bcsub](#) -- Sottrae un numero a precisione arbitraria da un altro

bcadd

(PHP 3, PHP 4, PHP 5)

bcadd -- Somma due numeri a precisione arbitraria

Descrizione

string **bcadd** (string primo operando, string secondo operando [, int precisione])

Somma il *primo operando* con il *secondo operando* e restituisce la somma sotto forma di stringa. Il parametro opzionale *precisione* è utilizzato per impostare il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. Esempio di bcadd()

```
<?php
$a = 1.234;
$b = 5;
```

```
echo bcadd($a, $b); // 6
echo bcadd($a, $b, 4); // 6.2340
?>
```

bccomp

(PHP 3, PHP 4, PHP 5)

`bccomp` -- Confronta due numeri a precisione arbitraria

Descrizione

int **bccomp** (string primo_operando, string secondo_operando [, int precisione])

Confronta il *primo_operando* e il *secondo_operando* e restituisce il risultato sotto forma di intero. Il parametro opzionale *precisione* è utilizzato per impostare il numero di cifre dopo il punto decimale che verranno usate nel confronto. Il valore restituito è 0 se i due operandi sono uguali. Se il *primo_operando* è più grande del *secondo_operando* il valore restituito è +1 e se il *primo_operando* è minore del *secondo_operando* il valore restituito è -1.

Esempi

Esempio 1. esempio di bccomp()

```
<?php
echo bccomp('1', '2') . "\n"; // -1
echo bccomp('1.00001', '1', 3); // 0
echo bccomp('1.00001', '1', 5); // 1
?>
```

bcdiv

(PHP 3, PHP 4, PHP 5)

`bcdiv` -- Divide due numeri a precisione arbitraria

Descrizione

string **bcdiv** (string primo operando, string secondo operando [, int precisione])

Divide il *primo operando* per il *secondo operando* e restituisce il risultato. Il parametro opzionale *precisione* imposta il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. esempio di bcddiv()

```
<?php
echo bcddiv(105, 6.55957, 3); // 16.007
?>
```

bcmmod

(PHP 3, PHP 4, PHP 5)

bcmmod -- Ricava il modulo di un numero a precisione arbitraria

Descrizione

string **bcmmod** (string operando, string modulo)

Ricava il modulo di *operando* usando *modulo*.

Esempi

Esempio 1. esempio di bcmmod()

```
<?php
echo bcmmod(4, 2); // 0
echo bcmmod(2, 4); // 2
?>
```

bcmul

(PHP 3, PHP 4, PHP 5)

bcmul -- Moltiplica due numeri a precisione arbitraria

Descrizione

string **bcmul** (string primo operando, string secondo operando [, int precisione])

Moltiplica il *primo operando* per il *secondo operando* e restituisce il risultato. Il parametro opzionale *precisione* imposta il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. esempio di bcmul()

```
<?php
echo bcmul(1.34747474747, 35, 3); // 47.162
echo bcmul(2, 4); // 8
?>
```

bcpow

(PHP 3, PHP 4, PHP 5)

bcpow -- Effettua l'elevamento a potenza

Descrizione

string **bcpow** (string x, string y [, int precisione])

Eleva *x* alla potenza *y*. Il parametro opzionale *precisione* può essere usato per impostare il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. bcpow() example

```
<?php
echo bcpow(4.2, 3, 2); // 74.08
?>
```

bcpowmod

(PHP 5)

`bcpowmod` -- Effettua l'elevamento a potenza, applicando quindi il modulo

Descrizione

string **bcpowmod** (string *x*, string *y*, string modulo [, int *precisione*])

Utilizza il metodo di esponenziazione veloce per elevare *x* alla potenza *y* rispetto al modulo *modulo*. Il parametro opzionale *precisione* può essere utilizzato per impostare il numero di cifre dopo il punto decimale.

Nota: Dal momento che questo metodo utilizza l'operatore modulo, numeri non naturali possono dare risultati inattesi. Un numero naturale è un qualsiasi numero positivo intero diverso da zero.

Esempi

Le seguenti istruzioni sono funzionalmente identiche. La versione **bcpowmod()**, comunque, esegue in meno tempo e può accettare parametri più grandi.

```
<?php
$a = bcpowmod($x, $y, $mod);

$b = bcmath(bcpow($x, $y), $mod);

// $a e $b sono uguali.

?>
```

bcscale

(PHP 3, PHP 4, PHP 5)

`bcscale` -- Imposta il valore di precisione di default per tutte le funzioni matematiche BCMath

Descrizione

bool **bcscale** (int *precisione*)

Questa funzione imposta il valore di default del parametro *precisione* per tutte le funzioni BCMath susseguenti, che non specificano esplicitamente un parametro di precisione numerica. Restituisce **TRUE** in caso di successo, **FALSE** in caso di fallimento.

Esempi

Esempio 1. esempio di bcscale()

```
<?php
// precisione di default : 3
bcscale(3);
echo bcddiv(105, 6.55957); // 16.007

// la stessa cosa, senza utilizzare bcscale()
echo bcddiv(105, 6.55957, 3); // 16.007

?>
```

bcsqrt

(PHP 3, PHP 4, PHP 5)

bcsqrt -- Ottiene la radice quadrata di un numero a precisione arbitraria

Descrizione

string **bcsqrt** (string operando [, int precisione])

Restituisce la radice quadrata di *operando*. Il parametro opzionale *precisione* imposta il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. esempio di bcsqrt()

```
<?php
echo bcsqrt(2, 3); // 1.414

?>
```

bcsb

(PHP 3, PHP 4, PHP 5)

bcsb -- Sottrae un numero a precisione arbitraria da un altro

Descrizione

string **bcsub** (string primo operando, string secondo operando [, int precisione])

Sottrae il *primo operando* dal *secondo operando* e restituisce il risultato in una stringa. Il parametro opzionale *scale* è usato per impostare il numero di cifre dopo il punto decimale nel risultato.

Esempi

Esempio 1. esempio di bcsub()

```
<?php
$a = 1.234;
$b = 5;

echo bcsub($a, $b); // -3
echo bcsub($a, $b, 4); // -3.7660

?>
```