

Corso di PHP

4.2 – Array associativi

M. Malatesta 4.2-Array associativi-05

1
07/04/2015

Prerequisiti

- Variabili semplici e strutturate
- Concetto di array
- Strutture di controllo

M. Malatesta 4.2-Array associativi-05

2
07/04/2015

Introduzione

Talvolta il caricamento di un array richiede un indice non numerico oppure può capitare di caricare le sue componenti senza progressione dell'indice.

Quali strumenti offre PHP per trattare queste situazioni ?
Come si usano questi strumenti ?

In questa Unità vediamo gli **array associativi**, una evoluzione del concetto di array, che sono uno strumento di notevole potenza e flessibilità, presente in tutti i moderni linguaggi di programmazione.

M. Malatesta 4.2-Array associativi-05

3
07/04/2015

Informazioni generali

Gli **array associativi** intervengono in tutte quelle situazioni in cui non si ha progressione dell'indice, o quando l'indice sia una stringa (detta **chiave alfanumerica** o **associativa**) oppure in quei casi in cui gli elementi vengano caricati in punti diversi del programma.

In questa Unità vediamo la notevole potenza di questa struttura dati.

M. Malatesta 4.2-Array associativi-05

4
07/04/2015

Array associativi

Per gli array associativi vediamo:

- **caricamento**
- **utilizzo**

M. Malatesta 4.2-Array associativi-05

5
07/04/2015

Array associativi - **caricamento**

Il caricamento e la creazione di un array associativo si effettua nel seguente modo:

```
$utenti = array ("claudio" => "sistemista", "nicola" => "utente",  
                "antonio" => "programmatore");
```

"claudio", "nicola" e "antonio" sono le **chiavi**, mentre
"sistemista", "utente" e "programmatore" sono i rispettivi
valori.

M. Malatesta 4.2-Array associativi-05

6
07/04/2015

Array associativi

- list() ed each()

```
<!-- ArrayAssociativo.php -->
<?php
    $utenti = array ("claudio" => "sistemista",
                    "nicola" => "utente",
                    "antonio" => "programmatore");
    while (list($chiave,$valore) = each($utenti))
    {
        echo "$chiave : $valore<br>";
    }
?>
```

L'array può essere stampato come noto mediante un ciclo **while** come segue..

La funzione **each** estrae, una ad una, le componenti del vettore *\$utenti* e tramite la funzione **list** le pone in *\$chiave* e *\$valore*.

M. Malatesta 4.2-Array associativi-05

7
07/04/2015

Array associativi

- key(), current() e next()

```
<!-- key-current.php -->
<?php
$persona[1]="anna";
$persona[3]="luca";
$persona[6]="maria";
$persona[ ]="carlo";
for ($i=0; $i<count ($persona); $i++ )
{ echo "<BR>Pos = $i. " Chiave =", key ($persona).
    " Valore =<B>". current ($persona). "</B>";
  next ($persona);
}
?>
```

Il caricamento avviene senza un preciso ordine degli indici

La funzione **key()** estrae, uno ad uno, i valori di *\$chiave* mentre la **current()** estrae uno ad uno il *\$valore*.

La funzione **next()** serve a spostarsi sulla componente successiva, mancando l'indice nel ciclo.

M. Malatesta 4.2-Array associativi-05

8
07/04/2015

Array associativi

- in_array()

```
<?php
$elenco= array ("elena","maria","nicoletta","anna",
               "lucia","antonio","piera","giorgio");
$nome="mario";
if ( in_array ($nome,$elenco))
{   echo "$nome è presente"; }
else
{   echo "$nome non è presente"; }
?>
```

La ricerca viene svolta mediante la funzione predefinita **in_array()**

La funzione **in_array** (*valore, vettore*) restituisce true o false a seconda che *valore* sia presente o meno nel *vettore*.

M. Malatesta 4.2-Array associativi-05

9
07/04/2015

Array associativi

- array-merge()

Il **merge** di due array indica la fusione dei loro valori in un terzo array.
Esiste un funzione PHP che effettua questa operazione per il *merge* anche di più di due array.

```
<?php
$nord = array ("Rossi" => "Lombardia","Verdi" => "Piemonte");
$centro = array ("Bianchi" => "Lazio","Celestini" => "Abruzzo");
$sud = array ("Marroni" => "Campania","Viola" => "Puglia");
$rappresentanti = array_merge ($nord,$centro, $sud);
while (list ($chiave, $valore) = each ($rappresentanti))
{   echo "$chiave : $valore<BR> "; }
?>
```

Merge dei
due array

M. Malatesta 4.2-Array associativi-05

10
07/04/2015

Array associativi

- sizeof()

L'esistenza di un array può essere verificata mediante la funzione **sizeof()**.

```
<!--SizeofArray.php -->
<?php
$mio_array=array(1,2,3);
$mio_array1=null; // $mio_array1=array();
echo "sizeof(\$mio_array)=".sizeof($mio_array)."<BR>";
echo "sizeof(\$mio_array1)=".sizeof($mio_array1)."<BR>";
if (sizeof($mio_array1) < "1")
{ echo $_SERVER['PHP_SELF'], ": Errore: L'array è vuoto";
  exit;}
else
  while (list($c, $v)=each($mio_array))
    echo $c, " - ", $v, "<BR>";
?>
```

Alternativa per creare
un array vuoto

Il valore "-1"
indica che
l'array è vuoto

M. Malatesta 4.2-Array associativi-05

11
07/04/2015

Array associativi

- array_push() e array_slice()

```
<!-- ArrayPushSlice.php -->
<?php
$pila = array();
for ($i=1; $i<6; $i++)
  array_push ($pila, $i);
foreach ($pila as $valore)
  echo $valore, "<BR>";
echo "<BR>Elimino i primi due elementi<BR>";
$out = array_slice($pila, 2);
foreach ($out as $valore)
  echo $valore, "<BR>";
?>
```

array_push() inserisce
elementi nell'array

Notare l'uso del
costrutto **foreach**

array_slice() elimina un
numero di elementi
dall'array

M. Malatesta 4.2-Array associativi-05

12
07/04/2015

Array associativi

- array_rand() e array_reverse()

```
<!-- ArrayOperations.php -->
<?php
$pila = array();
srand ((float) microtime () * 10000000);
for ($i=1; $i<6; $i++) array_push ($pila, $i);
foreach ($pila as $valore) echo $valore, "<BR>";
echo "Estraggo casualmente un valore: ", array_rand ($pila, 1), "<BR>";
$estratti = array_rand($pila, 2);
echo "Estraggo casualmente 2 valori: ", $pila[$estratti[0]], " - ",
    $pila[$estratti[1]];
echo "<BR>Inverto gli elementi<BR>";
$rev = array_reverse($pila, TRUE);
foreach ($rev as $valore)
    echo $valore, "<BR>";
?>
```

Forza la generazione
dei numeri casuali

array_rand() estrae
a caso da un array

array_reverse() inverte
gli elementi di un array

M. Malatesta 4.2-Array associativi-05

13
07/04/2015

Array associativi

- sort() e rsort()

```
<!-- Ordinamento.php -->
<?php
$persona[1]="anna";
$persona[3]="luca";
$persona[6]="maria";
$persona[ ]="carlo";
sort ($persona);
for ($i=0; $i<count($persona); $i++ )
{ echo "<BR>Pos = ".$i." Valore =<B>".current ($persona)."</B>";
  next ($persona);
}
?>
```

La funzione **sort()** dà
l'ordinamento dei
valori del vettore

La funzione **rsort()** serve a ordinare in senso decrescente
gli elementi del vettore.

M. Malatesta 4.2-Array associativi-05

14
07/04/2015

Array associativi

- sort() e rsort()

OSSERVAZIONI:

Le funzioni **sort()** e **rsort()**:

- eseguono un ordinamento sullo stesso vettore di input;
- ridefiniscono gli indici ed eliminano le chiavi alfanumeriche
- eliminano tutte le posizioni vuote nell'array

Per mantenere le chiavi associative, si usano le funzioni:

- **asort()**
- **arsort()**

Array associativi

- explode()

```
<!-- explode.php -->
<?php
$dati ="Brontolo,Eolo,Cucciolo,Mammolo,Pisolo,Dotto,Gongolo";
$nani =explode ("",$dati);
while (list($chiave,$valore) = each($nani))
{ echo "Chiave: ".$chiave." Valore: <b>".$valore."</b><br>";
}
?>
```

La funzione **explode** serve a creare da una stringa (o un file di dati) un array utilizzando un dato carattere di separazione, che nell'esempio è la virgola “,”.

Array associativi

- implode()

```
<!-- implode.php -->
<?
$persona[1]="anna";
$persona[3]="luca";
$persona[6]="maria";
$persona[ ]="carlo";
$utenti = implode ("-", $persona);
echo "Utenti: ".$utenti;
?>
```

La funzione **implode** è simmetrica alla **explode** e crea una stringa (o un file di dati) partendo da un array e utilizzando un dato carattere di separazione, che nell'esempio è il trattino "-".

M. Malatesta 4.2-Array associativi-05

17
07/04/2015

Array associativi

- reset(), next(), prev(), end()

```
<!-- reset-end.php -->
<?php
$persona[1]="anna";
$persona[3]="luca";
$persona[6]="maria";
$persona[ ]="carlo";
reset($persona);
echo key($persona), " - ", current($persona), "<BR>";
next($persona);
echo key($persona), " - ", current($persona), "<BR>";
end($persona);
echo key($persona), " - ", current($persona), "<BR>";
prev($persona);
echo key($persona), " - ", current($persona), "<BR>";?>
```

reset() si porta sul primo elemento,
end() si porta sull'ultimo elemento.

next() si porta sul successivo elemento,
prev() si porta sul precedente elemento.

M. Malatesta 4.2-Array associativi-05

18
07/04/2015

Argomenti

- Array associativi
 - caricamento
 - **list()** ed **each()**
 - **key()**, **current()** e **next()**
 - **in_array()**
 - **array_merge()**
 - **sizeof()**
 - **array_push()** e **array_slice()**
 - **array_rand()** e **array_reverse()**
 - **sort()** e **rsort()**
 - **explode()**
 - **implode()**
 - **reset()**, **next()**, **prev()**, **end()**

M. Malatesta 4.2-Array associativi-05

19
07/04/2015

Altre fonti di informazione

- P.Camagni, R.Nicolassy – PHP ed. HOEPLI Informatica

M. Malatesta 4.2-Array associativi-05

20
07/04/2015