Utilizare matrice - Array

Deseori este convenabila stocarea mai multor valori într-o variabila. O asemenea variabila se numeste matrice (Array sau "tablou"), iar valorile individuale se numesc elementele matricei. Aici, "matrice", "Array" sau "tablou" reprezinta acelasi lucru (in varianta engleza: Array), fapt pentru care vom folosi orcare dintre aceste denumiri.

Fiecare element al unei matrici are doua elemente importante: **cheia si valoarea elementului.**

Matricele pot fi create folosind doua medode principale:

1 - utilizand direct instructiunile de atribuire

2 - folosind sintaxa array() despre care am vorbit in Lectia 2 despre tipul variabilelor.

1. Crearea de matrice folosind functia de atribuire

Pentru a crea o matrice, atribuiti unui element al matricei o valoare si o cheie. De exemplu, urmatoarea instructiunea de atribuire:

$clasa[1] = "geometrie";

creeaza un tablou denumit **"$clasa"** si un element cu valoarea "geometrie" identificat prin cheia "1".

Pentru a stoca în matrice o a doua valoare, puteti folosi urmatoarea instructiune de atribuire:

$clasa[2] = "algebra";

Pentru a obtine acces la un element al matricei, specificati numele matricei si valoarea cheii. De exemplu, instructiunea de atribuire

$clasa_mate = $clasa[1];

Atribuie valoarea "geometrie" variabilei $clasa_mate.

Cheile folosite pentru identificarea elementelor unei matrice nu trebuie sa fie obligatoriu numere consecutive; nici macar nu trebuie sa fie numere. De exemplu, iata instructiuni de atribuire care creeaza o matrice ce stocheaza preferinte în materie de fructe:

$preferinte[Nelu] = "cirese";

$preferinte[Radu] = "mere";

$preferinte[Gabi] = "pere";

2. Crearea de matrice folosind sintaxa array()

Dincolo de utilizarea instructiunilor de atribuire, cealalta modalitate principala de creare a unui tablou PHP consta în utilizarea functiei array(). Iata un exemplu simplu, care creeaza un tablou având drept chei valori întregi consecutive:

$limbaje = array("Perl", "PHP", "Python");

Deoarece valoarea cheilor nu a fost specificata, acestea vor fi automat trecute de program ca numere intregi consecutive, incepand de la "0" (prima cheie va avea valoarea '0', a doua cheie va avea valoarea '1', ...)

Aceasta instructiune creeaza o matrice care contine urmatoarele asocieri:

0 => Perl

1 => PHP

2 => Python

Daca doriti sa asociati unei valori o anumita cheie, puteti folosi operatorul =>, astfel:

$limbaje = array(10=>"Perl", "PHP", "Python");

Aceasta instructiune creeaza urmatoarea matrice:

10 => Perl

11 => PHP

12 => Python

Ca si în cazul utilizarii unei instructiuni de atribuire pentru crearea unei matrice valorile cheilor nu trebuie sa fie consecutive si nici macar întregi:

$limbaje = array("PHP"=>"Ridicat", "Python"=>"Mediu", "Perl"=>"Redus");

Aceasta instructiune creeaza urmatoarea matrice:

PHP => Ridicat

Python => Mediu

Perl => Redus

3. Matrici multi-dimensionale

Este posibil ca un element al matricei sa fie de asemenea o matrice. In acest caz avem de-a face cu o matrice multi-dimensionala.

Sa luam un exemplu practic :

```
<?php

$multiDimArray[firstLine] = array(1=>10, 2=>20, "a"=>"alpha");

$multiDimArray["nextLine"] = array(1=>20, 2=>40, "b"=>"beta");


echo "<br />".$multiDimArray["firstLine"][1];

echo "<br />".$multiDimArray["nextLine"][1];

echo "<br />".$multiDimArray["firstLine"][2];

echo "<br />".$multiDimArray["firstLine"]["a"];

echo "<br />".$multiDimArray["nextLine"]["b"];

?>
```

Sau putem scrie acelasi script si astfel (rezultatul e acelasi) :

```
<?php

$multiDimArray = array("firstLine"=>array(1=>10, 2=>20, "a"=>"alpha"), "nextLine"=>array(1=>20, 2=>40, "b"=>"beta"));


echo "<br />".$multiDimArray["firstLine"][1];

echo "<br />".$multiDimArray["nextLine"][1];

echo "<br />".$multiDimArray["firstLine"][2];

echo "<br />".$multiDimArray["firstLine"]["a"];

echo "<br />".$multiDimArray["nextLine"]["b"];
```

?>

Rezultatul afisat va fi urmatorul :

10

20

20

alpha

beta

In exemplul de mai sus am declarat un array cu 2 linii si 3 coloane. Prima linie este identificata de array-ul cu numele "fisrtLine", iar ce-a de-a doua linie este identificata de "nextLine".

Observati de asemenea foloasirea in functia "echo" a operatorului de concatenare "." impreuna cu "
" (care este un element din XHTML), pentru ca la afisarea in browser fiecare rezultat al functiei "echo" sa fie trecut pe o linie noua. In caz contrar rezultatele ar fi fost pe aceeasi linie : 10 20 20 alfa beta

4. Accesul la datele dintr-un array

Când ati stocat date într-un tablou, puteti obtine acces la valoarea unui element al tabloului sau îi puteti modifica valoarea prin intermediul cheii asociate elementului.

De exemplu, sa presupunem ca folositi urmatoarele instructiuni pentru a crea un tablou:

```
$x = array(1=>10, 2=>100, 3=>1000);
```

Puteti obtine acces la valoarea asociata cheii 2 prin intermediul unei instructiuni ca aceasta:

```
$y = 3*$x[2];
```

Similar, puteti modifica valoarea asociata cheii 3 prin intermediul unei instructiunu ca aceasta:

```
$x[3] = 101;
```

5. Parcurgere Array numeric

Un Array ale carui chei sunt valori întregi consecutive se numeste "Array numeric" (sau secvential).

În general, valoarea cea mai mica a unei chei dintr-un tablou numeric este zero; totusi, puteti crea array numeric incepand cu valoarea 1 sau orice alta valoare întreaga ca valoare minima a cheii.

În cazul în care cunoasteti valoarea minima a cheii unui tablou secvential, puteti parcurge tabloul folosind o bucla for. Pentru aceasta, initializati variabila de bucla la valoarea cea mai redusa a cheii. Folositi functia count() pentru a forma expresia de test (conditia) a buclei. Functia count() returneaza numarul elementelor dintr-un tablou.

Iata un exemplu simplu

```
<?php

$limbaje = array(0=>"Perl", 1=>"PHP", 2 =>"Python");

$limita = count($limbaje);

for ($i = 0; $i < $limita; $i++) {

    echo "<br />$i => $limbaje[$i]";

}

?>
```

Prima variabila $imbaje creeaza tabloul.

Cea de-a doua variabila $limita foloseste instructiunea count() pentru a obtine numarul elementelor din tablou.

Instructiunea for foloseste variabila bucla $i pentru a parcurge iterativ (element cu element) tabloul; corpul instructiunii include o instructiune echo care afiseaza cheia si valoarea fiecarui element din tablou.

Datele de iesire vor fi astfel:

```
0 => Perl

1 => PHP

2 => Python
```

Sa ne concentram asupra problemei de a determina daca un tablou contine o anumita valoare.

Iata un exemplu:

```
<?php

$limbaje = array(0=>"Perl", 1=>"PHP", 2 =>"Python");

$cauta = PHP;     // se cauta in tabloul $limbaje valoarea $cauta

$limita = count($limbaje);
```

```
for ($i = 0; $i < $limita; $i++) {
    echo "<br />Determinarea unei identitati cu $limbaje[$i]";
    if ($cauta == $limbaje[$i]) {
       echo "<br />$cauta este un limbaj excelent.";
    }
}
?>
```

Prima variabila $imbaje creeaza tabloul în care se va cauta.

Cea de-a doua instructiune atribuie valoarea "PHP" variabilei $cauta; (în exemplu, se cauta în tablou valoarea stocata în variabila $cauta).

Dupa comentariu (care incepe cu //), urmatoarea instructiune obtine numarul elementelor din tablou si stocheaza aceasta valoare în variabila $limita.

Instructiunea for functioneaza ca mai înainte; de data aceasta însa, corpul sau contine alte instructiuni si se executa o alta operatie. O instructiune echo afiseaza valoarea fiecarui element al tabloului pe masura ce parcurgerea tabloului avanseaza. Instructiunea if testeaza fiecare element si afiseaza un mesaj daca valoarea elementului este una si aceeasi cu valoarea variabilei $cauta.

Iata rezultatul rularii exemplului:

Determinarea unei identitati cu Perl

Determinarea unei identitati cu PHP

PHP este un limbaj excelent.

Determinarea unei identitati cu Python

• Incepand cu versiunea PHP 5.4 a fost introdus un nou mod de a defini variabile array, folosind o sintaxa scurta.

Ex.:

```
$arr = [];          // array gol, in loc de array();

$arr = [1, 2, 3, 4];    // array numeric
```

```
$arr = ['eng'=>'courseweb.net', 'ro'=>'marplo.net', 'num'=>8];       // array asociativ
```

- Array-ul creat cu aceasta sintaxa poate fi accesat, modificat si parcurs ca orice array.

Instructiunea break

Observati ca parcurgerea continua chiar si dupa gasirea valorii cautate.

Când se cauta într-un tablou, executia cautarii poate fi oprita dupa gasirea elementului dorit; continuarea parcurgerii in Array nu face decât sa iroseasca resursele calculatorului, fara a afecta rezultatele operatiei.

Pentru a opri executia unei parcurgeri, puteti folosi instructiunea break, care determina încheierea imediata a buclei care o contine.

Iata cum se poate revizui exemplul anterior, astfel încât sa includa o instructiune break:

```
<?php

$limbaje = array(0=>"Perl", 1=>"PHP", 2 =>"Python");

$cauta = PHP;     // se cauta in tabloul $limbaje valoarea $cauta

$limita = count($limbaje);

for ($i = 0; $i < $limita; $i++) {

    echo "<br />Determinarea unei identitati cu $limbaje[$i]";

    if ($cauta == $limbaje[$i]) {

       echo "<br />$cauta este un limbaj excelent.";

       break;

    }

}

?>
```

Acum, dupa stabilirea unei identitati, instructiunea break provoaca sistarea buclei for.

Iata datele de iesire rezultate, care acum omit examinarea inutila a elementului tabloului asociat cu limbajul "Python":

```
Determinarea unei identitati cu Perl
```

Determinarea unei identitati cu PHP

PHP este un limbaj excelent.

Instructiunea continue

O instructiune corelata cu instructiunea break este continue.

Instructiunea continue opreste secventa curenta a buclei, determinând evaluarea imediata a expresiilor de incrementare si de test.

Ca un exemplu, sa presupunem ca doriti sa cautati în tabloul $limbaje pentru a determina numarul limbajelor care au nume scurte, adica nume alcatuite din maximum 4 caractere. Iata un exemplu care executa aceasta prelucrare a datelor:

```
<?php

$limbaje = array(0=>"Perl", 1=>"PHP", 2 =>"Python");

$scurt = 0;

$limita = count($limbaje);

for ($i = 0; $i < $limita; $i++) {

    $n = strlen($limbaje[$i]);

    echo "<br />$limbaje[$i] are $n caractere lungime";

    if ($n > 4) continue;

       $scurt++;

}

echo "<br />Au fost gasite $scurt limbaje cu nume scurte.";

?>
```

O instructiune de atribuire stabileste valoarea initiala a variabilei $scurt la zero, folosita pentru a numara numele scurte gasite.

Instructiunea for se aseamana celor folosite anterior. Corpul acestei instructiuni difera, desigur, de cele folosite anterior. Valoarea variabilei $n este stabilita ca fiind egala cu numarul caracterelor care compun numele limbajului, folosind functia strlen(), care calculeaza lungimea unui sir.

Daca instructiunea if stabileste ca elementul curent al tabloului face referire la un limbaj cu nume lung, se executa instructiunea continue.

Instructiunea continue determina trecerea parcurgerii la urmatorul element din tablou, fara a mai executa expresia "$scurt++"; daca nu au mai ramas elemente în tablou, bucla for îsi încheie executia.

La finalizarea parcurgerii, o instructiune echo afiseaza numarul numelor scurte de limbaje gasite în tablou. Iata rezultatul:

Perl are 4 caractere lungime

PHP are 3 caractere lungime

Python are 6 caractere lungime

Au fost gasite 2 limbaje cu nume scurte.