

Entorns de Desenvolupament



12. Funció count i foreach

12. Funció count i foreach

Abans de poder **recórrer** un **array**, hem de saber quina és la seua **mida** per poder recórrer-lo. Suposem que un array té tres elements: hem de donar una instrucció perquè s'extrega el valor associat a cada un dels elements de l'array, que normalment serà una cosa semblant a "**per a cadascun dels tres elements de l'array, extreure el seu valor**".

No obstant això, si l'array té 20 elements, la instrucció serà del tipus "per a cada un dels vint elements de l'array, extreure el seu valor". Com veiem, moltes vegades ens cal **saber el nombre d'elements que hi ha a l'array** per poder recórrer-lo, i per això ens resultarà útil la funció **count**.

La funció **count** retorna la quantitat d'elements que hi ha a l'array. És a dir per a un array de 4 elements, la funció **count** tornarà el número 4. Recorda que si es tracta de valors numèrics d'índexs, els quatre valors numèrics seran normalment 0, 1, 2 i 3 en lloc d'1, 2, 3 i 4.

Exemple 01:

```
<?php
$estacions[0] = "Primavera"; // tots en ordre
$estacions[1] = "Estiu";
$estacions[2] = "Tardor";
$estacions[3] = "Hivern";
echo "Nombre d'elements en estacions: ".count($estacions)."<br>";
$alumnat[]= "Carla"; //amb índex no consecutius
$alumnat[7]= "Xavi";
$alumnat[15]= "Rubén";
echo "Nombre d'elements en alumnat: ".count($estacions)."<br>";
?>
```

El valor resultant de **count()** en els dos casos és **4** i **3** respectivament.

Recorregut d'arrays unidimensionals

Ara que ja coneixem què són els arrays, hem de conèixer com **recórrer** per **extreure** o **comprovar** els **valors** que conté cada un dels elements de l'array.

El arrays es poden recórrer de moltes maneres, tractarem de veure en principi les més habituals i senzilles.

Recorregut d'arrays mitjançant for o while:

Exemple 02:

```
<?php
$array[0] = "U"; $array[1] = "Dos";
$array[2] = "Tres"; $array[3] = "Quatre";
$array[4] = "Cinc"; $array[5] = "Sis";
$array[6] = "Set"; $array[7] = "Huit";
// recorreguem l'array amb un for.
// suposem que tots els índex són correlatius
//i comencen en 0.
for($i = 0; $i < count($array); $i++) {
    echo "Element ".$i." : ".$array[$i]. "<br>";
}
?>
```

```
$midaArray=count($array);
$i=0;
while($i < count($array)) {
    echo "Element ".$i." : ".$array[$i]. "<br>";
    $i++;
}
```

Eixida:

Element 0: U
 Element 1: Dos
 Element 2: Tres
 Element 3: Quatre
 Element 4: Cinc
 Element 5: Sis
 Element 6: Set
 Element 7: Huit

Veiem ara l'**exemple 03**, on els índex de l'arrayno comencen en 0:

```
<?php
```

```

$array[5] = "U";
$array[6] = "Dos";
$array[7] = "Tres";
$array[8] = "Quatre";
$array[9] = "Cinc";
$array[10] = "Sis";
$array[11] = "Set";
$array[12] = "Huit";

echo "Elements inicialitzats a la matriu: ".count($array)."<br>";
for ($i=0; $i<count($array); $i++) {
    echo "Element ".$i." : ".$array[$i]. "<br>";
}

echo "<br>Element 15 : ".$array[15]. "<br>";
echo "<br>Elements d'un altre array: ".count($altreArray[14][33]).
"<br>";
?>
```

El resultat obtingut serà semblant al següent:

| | |
|--|---|
| <p>Elements inicialitzats a la matriu: 8</p> <p>Notice: Undefined array key: 0</p> <p>Element 0 :</p> <p>Notice: Undefined array key: 1</p> <p>Element 1 :</p> <p>Notice: Undefined array key: 2</p> <p>Element 2 :</p> <p>Notice: Undefined array key: 3</p> <p>Element 3 :</p> <p>Notice: Undefined array key: 4</p> <p>Element 4 :</p> | <p>Element 5 : U</p> <p>Element 6 : Dos</p> <p>Element 7 : Tres</p> <p>Notice: Undefined array key: 15</p> <p>Element 15 :</p> <p>Notice: Undefined variable: \$altreArray</p> <p>Elements d'un altre array: 0</p> |
|--|---|

Analitzem el que està passant. La matriu **\$array** **no té definits** quins són els seus **elements** d'**índex 0, 1, 2, 3 i 4**. En canvi sí que **té definits** valors per als seus **elements** d'**índex 5, 6, 7, 8, 9, 10, 11 i 12**. Hi ha **8 elements amb valors definits**, però els seus **índexs no són 0, 1, 2, 3, 4, 5, 6, 7** com és l'habitual, sinó que **són 5, 6, 7, 8, 9, 10, 11 i 12**.

En executar el **for** començant amb un valor de la variable de control **\$i** igual a zero, s'executa:

```
echo "Element ".0." : ".$array[0]. "<br>";
```

En no tenir un valor assignat, **\$array[0]** retorna un missatge d'error. El mateix passa amb els índexs 1, 2, 3, 4 i només és en arribar a l'índex 5 quan es mostren valors per pantalla correctament.

No obstant això, només es **mostraran** els **elements** amb **índex 5 a 8 de l'array**, quedant la **resta d'elements sense** ser **mostrats**.

A més, comprovem que si invoquem **elements** de l'array per **damunt** de l'**índex** màxim definit, obtenim un **missatge d'error**, i si **invoquem** un nom **d'array** que **no** hem **declarat** ni **utilitzat**, amb **qualsevol índex**, només obtenim un **missatge d'error**.

En aquest exemple podríem mostrar tots els elements de l'array fent un "xicotet canvi":

```
for($i=5; $i<count($array)+5; $i++)
```

No obstant això aquesta solució no és satisfactòria perquè no la podem aplicar com a solució per recórrer qualsevol array. A manera de resum direm el següent:

- **count** ens retorna la **quantitat d'elements** inicialitzats l'array.
- Si els **índexs** de l'array **no** són **seqüencialment** 0, 1, 2, 3, etc. el **recorregut** amb un **for** tradicional pot resultar **no satisfactori**, per la qual cosa haurem de pensar en altres alternatives per recórrer l'array.

Count amb arrays multidimensionals

En el cas **d'arrays** de **més d'una dimensió**, la funció **count** retorna la **quantitat d'elements** que hi ha en un **nivell** de l'array **definit** i per a un **índex definit**. Anomenem **nivell** de l'array a cadascun dels **claudàtors** existents en l'array. Per exemple si hem definit:

```
$array[2][5][1][6][2]=33;
```

i

```
$array[2][5][3][1][9]=55;
```

Diem que:

- En el **primer nivell** de l'array (primer índex) hi ha un sol índex: el **2**.
 - En el **segon nivell** de l'array per índex **2** hi ha un sol índex: el **5**.
 - En el **tercer nivell** de l'array per índex **5** el **primer índex** és l'**1**.
 - En el **quart nivell** de l'array per índex **1** hi ha un sol índex: el **6**.
 - En el **cinquè nivell** de l'array per índex **6** hi ha un sol índex: el **2**.
 - En el **tercer nivell** de l'array per índex **5** el **segon índex** és el 3.
 - En el **quart nivell** de l'array per índex 3 hi ha un sol índex: l'**1**.
 - En el **cinquè nivell** de l'array per índex **1** hi ha un sol índex: el **9**.

Amb un exemple comprendrem millor el funcionament de **count** a una matriu multidimensional.

Exemple 04:

```
<?php
$array1[6]=99;
$array1[22]=87;
echo "La quantitat d'elements d'array1 és: ".count($array1)."<br>";
echo "<br>";
$array2[0][0][0] = "prova1";  $array2[0][0][1] = "prova2";
$array2[0][0][2] = "prova3";
$array2[0][1][0] = "prova4";  $array2[0][1][1] = "prova5";
```

```

$array2[1][0][1] = "prova6";
$array2[1][1][2] = "prova7";
$array2[1][2][1] = "prova8";
$array2[2][0][0] = "prova9";  $array2[2][0][1] = "prova10";
$array2[5][0][1] = "prova11";

echo "Nombre d'índexs 1r nivell array2: " .count($array2)."<br>";
echo "Nombre d'índexs 2n nivell array2[0]: " .count($array2[0])."<br>";
echo "Nombre d'índexs 2n nivell array2[1]: " .count($array2[1])."<br>";
echo "Nombre d'índexs 2n nivell array2[2]: " .count($array2[2])."<br>";
echo "Nombre d'índexs 2n nivell array2[3]: " .count($array2[3])."<br>";
echo "Nombre d'índexs 2n nivell array2[5]: " .count($array2[5])."<br>";
echo "Nombre d'índexs 3r nivell array2[0][0]: " .count($array2[0][0])."<br>";
echo "Nombre d'índexs 3r nivell array2[0][1]: " .count($array2[0][1])."<br>";
echo "Nombre d'índexs 3r nivell array2[1][0]: " .count($array2[1][0])."<br>";
echo "Nombre d'índexs 3r nivell array2[1][1]: " .count($array2[1][1])."<br>";
echo "Nombre d'índexs 3r nivell array2[1][2]: " .count($array2[1][2])."<br>";
echo "Nombre d'índexs 3r nivell array2[2][0]: " .count($array2[2][0])."<br>";
echo "Nombre d'índexs 3r nivell array2[5][0]: " .count($array2[5][0])."<br>";

?>

```

El **resultat** obtingut serà el següent:

| | |
|--|---|
| La quantitat d'elements d'array1 és: 2 | Nombre d'índexs 2n nivell array2[5]: 1 |
| Nombre d'índexs 1r nivell: 4 | Nombre d'índexs 3r nivell array2[0][0]: 3 |
| Nombre d'índexs 2n nivell array2[0]: 2 | Nombre d'índexs 3r nivell array2[0][1]: 2 |
| Nombre d'índexs 2n nivell array2[1]: 3 | Nombre d'índexs 3r nivell array2[1][0]: 1 |
| Nombre d'índexs 2n nivell array2[2]: 1 | Nombre d'índexs 3r nivell array2[1][1]: 1 |
| Notice: Undefined offset: 3 | Nombre d'índexs 3r nivell array2[1][2]: 1 |
| Nombre d'índexs 2n nivell array2[3]: | Nombre d'índexs 3r nivell array2[2][0]: 2 |
| | Nombre d'índexs 3r nivell array2[5][0]: 1 |

Ens hem de fixar en què per exemple `count($array2[0][1])` ens retorna la quantitat d'elements existents de tipus `$array2[0][1][x]` on x és qualsevol índex, és a dir, la quantitat d'elements de tercer nivell de l'array per als índexs inicials 0 i 1.

Recorregut d'arrays multidimensionals

Per recórrer una matriu multidimensional, haurem d'anar **niant tantes estructures repetitives com dimensions** tinga **l'array**. Amb un exemple es veurà tot molt més clar.

Exemple 05:

```
<?php
$array3[0][0] = "U";
$array3[0][1] = "Dos";
$array3[1][0] = "Tres";
$array3[1][1] = "Quatre"; //tots els index són consecutius
for($i=0; $i<count($array3); $i++) { //per al primer nivell de l'array
    for($j=0; $j<count($array3[$i]); $j++) { //per al segon nivell
        echo "Valor [".$i."][".$j."]: ".$array3[$i][$j]."<br>";
    }
}
echo "<br>";
//per reinicialitzar l'array:
$array3=array();
$array3[0][0][0] = "Cinc";
$array3[0][0][1] = "Sis";
$array3[0][0][2] = "Set";
$array3[0][1][0] = "Huit";
$array3[0][1][1] = "Nou"; //tots els index són consecutius
for($i=0; $i<count($array3); $i++) { //per al primer nivell de l'array
    for($j=0; $j<count($array3[$i]); $j++) { //per al segon nivell
        for($k=0; $k<count($array3[$i][$j]); $k++) { //per al tercer nivell
            echo "Valor [".$i."][".$j."][".$k."]: ".$array3[$i][$j][$k]."<br>";
        }
    }
}
?>
```


Per poder obtenir el resultat desitjat els **elements** dels arrays han d'estar **definit**s usant **índexs** que **comencen** a **0** i **siguen progressivament 1, 2, 3, 4 ...** **Si no** fóra així, aquests bucles, en estar definits partint de 0, **no funcionarien**.

Veim també en la lògica dels bucles, per exemple el primer bucle ho podríem llegir així:

per a cada element des de 0 fins la quantitat d'elements de primer nivell, i per a cada element des de 0 fins la quantitat d'elements de segon nivell amb l'índex extret prèviament, mostrar el contingut de l'array.

Si per exemple en el primer nivell tenim dos índexs que són 0 i 1, començarem amb l'índex 0 i es veurà per l'índex 0 quants índexs hi ha al segon nivell (són 2). D'aquesta manera el primer que es mostrarà són els índexs [0] [0] i [0] [1]. Una vegada completat el recorregut de l'índex 0 es passa a l'índex 1 i se segueix el mateix procés.

Recorregut d'arrays mitjançant foreach

PHP incorpora una **manera "còmoda"** per poder **recórrer** tots els elements d'un **array**. Aquesta forma es basa en l'ús de la instrucció **foreach**. Aquest tipus de recorreguts se sol usar quan obtenim dades d'una **base de dades**. El tipus **d'array** que ens sol **tornar** una **consulta** a la **base de dades** és similar al següent:

```
$persones = array(  
    array('nom'=>'Antonio', 'cognoms' => 'Gómez Gómez', 'telefon' => '675.832.145'),  
    array('nom'=>'Pedro', 'cognoms' => 'Guillén Gastón', 'telefon' => '674.562.178'),  
    array ('nom' => 'Lola', 'cognoms' => 'Pi Orts', 'telefon' => '689.765.432'),  
    array ('nom' => 'Rubén', 'cognoms' => 'Part Pla', 'telefon' => '654.213.896')  
);
```

L'anterior forma d'expressió és **equivalent** a aquesta altra:

```
$persones[0]['nom'] = 'Antonio';  
$persones[0]['cognoms'] = 'Gómez Gómez';  
$persones[0]['telefon'] = '675.832.145';  
$persones[1]['nom'] = 'Pedro';  
$persones[1]['cognoms'] = 'Guillén Gastón';  
$persones[1]['telefon'] = '674.562.178';  
$persones[2]['nom'] = 'Lola';  
$persones[2]['cognoms'] = 'Pi Orts';  
$persones[2]['telefon'] = '689.765.432';  
$persones[3]['nom'] = 'Rubén';  
$persones[3]['cognoms'] = 'Part Pla';  
$persones[3]['telefon'] = '654.213.896';
```

Els índexs de l'array en el segon nivell són cadenes (es tracta **d'arrays associatius**) mentre que en el primer nivell els índexs de l'array són nombres (**arrays tradicionals**). En el primer nivell els valors 0, 1 i 2 són assignats automàticament per php perquè no s'ha especificat una altra cosa. Per recórrer arrays de manera còmoda s'usa la instrucció **foreach**.

Exemple 06:

```
<?php  
$persones = array(  
    array('nom'=>'Antonio', 'cognoms' => 'Gómez Gómez', 'telefon' => '675.832.145'),  
    array('nom'=>'Pedro', 'cognoms' => 'Guillén Gastón', 'telefon' => '674.562.178'),  
    array('nom' => 'Lola', 'cognoms' => 'Pi Orts', 'telefon' => '689.765.432'),  
    array('nom' => 'Rubén', 'cognoms' => 'Part Pla', 'telefon' => '654.213.896'),  
);  
foreach($persones as $persona) { //recorreguem l'array $persones amb foreach  
    echo 'Nom: '.$persona['nom'].'<br>';  
}  
?>
```

Com podem observar, anirem recorrent totes les files i mostrant el nom.

\$persona és una **variable temporal** que **només existeix** durant l'execució de la

instrucció foreach, i aquesta variable va **prenent** en cada **repetició** o iteració el **valor** de **l'element següent**. En aquest cas, l'element que hi ha és un altre array. Al seu torn, indiquem que per a cada un dels arrays extrets en el recorregut, se'ns mostre l'ítem de l'array l'índex del qual és **'nom'**.

En aquest cas podríem obtenir el mateix resultat usant aquest codi:

```
echo 'Nom: '.$persones[0]['nom'].'<br>';  
echo 'Nom: '.$persones[1]['nom'].'<br>';  
echo 'Nom: '.$persones[2]['nom'].'<br>';  
echo 'Nom: '.$persones[3]['nom'].'<br>';
```

Però per recórrer un array de diversos centenars d'elements no resultarà pràctic escriure'ls un a un. Si volguérem obtenir la resta de valors, podríem fer-ho així:

```
foreach($persones as $persona) {  
    echo 'Nom: '.$persona['nom'].' - '  
    echo 'Cognoms: '.$persona['cognoms'].' - '  
    echo 'Telèfon: '.$persona['telefon'].'<br>';  
}
```

I obtindríem com a resultat:

```
Nom: Antonio - Cognoms: Gómez Gómez - Telèfon: 675.832.145  
Nom: Pedro - Cognoms: Guillén Gastón - Telèfon: 674.562.178  
Nom: Lola - Cognoms: Pi Orts - Telèfon: 689.765.432  
Nom: Rubén - Cognoms: Part Pla - Telèfon: 654.213.896
```

Si volguérem recórrer també automàticament l'array que conté la informació de les persones, podríem tornar a utilitzar un altre **foreach**;

```
foreach($persones as $persona) { //recorreríem els elements 0,1,2,3  
    foreach($persona as $valorCamp) { //recorreríem nom,cognoms,telefon  
        echo 'Element: '.$valorCamp.'<br>';  
    }  
}
```

I si a més volguérem veure el nom de l'índex associatiu, ho podríem fer així:

```
foreach($persones as $persona) { //recorreríem els elements 0,1,2,3
    foreach($valor as $clau => $valorCamp){ //recorreríem nom,cognoms,telefon
        echo 'Valor['.$clau.']: '.$valorCamp.'<br>';
    }
}
```

Obtenint en aquest cas com a eixida per pantalla:

Valor[nom]: Antonio

Valor[cognoms]: Gómez Gómez

Valor[telefon]: 675.832.145

Valor[nom]: Pedro

Valor[cognoms]: Guillén Gastón

Valor[telefon]: 674.562.178

Valor[nom]: Lola

Valor[cognoms]: Pi Orts

Valor[telefon]: 689.765.432

Valor[nom]: Rubén

Valor[cognoms]: Part Pla

Valor[telefon]: 654.213.896

Altre exemple més senzill:

```
<?php
    $estacions = array("Primavera","Estiu","Tardor","Hivern");
    foreach($estacions as $estacio) {
        echo "Estació: ". $estacio. '<br>';
    }
?>
```

El resultat que obtenim és similar al que es veu a continuació. Com es pot comprovar, la variable **\$estacio** (variable **transitòria**) va prenent en cada repetició del bucle el contingut dels elements o ítems de l'array que estem recorrent.

Estació: Primavera

Estació: Estiu

Estació: Tardor

Estació: Hivern

Si el que volem és canviar el valor dels elements de l'array utilitzant **foreach**, no podem utilitzar la variable transitòria, ja que el valor original no canviaria. Per exemple:

```
<?php
$estacions = array("Primavera","Estiu","Tardor","Hivern");
$comptador=1;
foreach($estacions as $valor) {
    $valor = $comptador." ".$valor;
    $comptador++;
}
?>
```

Si ara mostrem el resultat, obtindrem els valors originals, ja que utilitzant **\$valor**, no els hem modificat dins de l'array **\$estacions**.

```
<?php
foreach($estacions as $valor) {
    echo $valor.'<br>';
}
?>
```

L'eixida obtinguda no és l'esperada, no s'ha afegit el comptador a l'array:

Primavera

Estiu

Tardor

Hivern

Per modificar l'array efectivament, és necessari utilitzar el paràmetre que ens proporciona la clau i accedir-hi utilitzant l'array original, per exemple:

```
<?php
$estacions = array("Primavera","Estiu","Tardor","Hivern");
$comptador=1;
foreach($estacions as $clau => $estacio) {
    $estacions[$clau] = $comptador." ".$estacio;
    $comptador++;
}
?>
```

Si ara mostrem els valors de l'array obtindrem els valors modificats, ja que hem accedit directament a l'array, no els hem modificat dins de l'array **\$estacions**.

```
<?php
foreach($estacions as $valor) {
    echo $valor.'<br>';
}
?>
```

El **resultat** és:

1 Primavera

2 Estiu

3 Tardor

4 Hivern