

7

ARRAYS

7.1 Arrays met meer dimensies

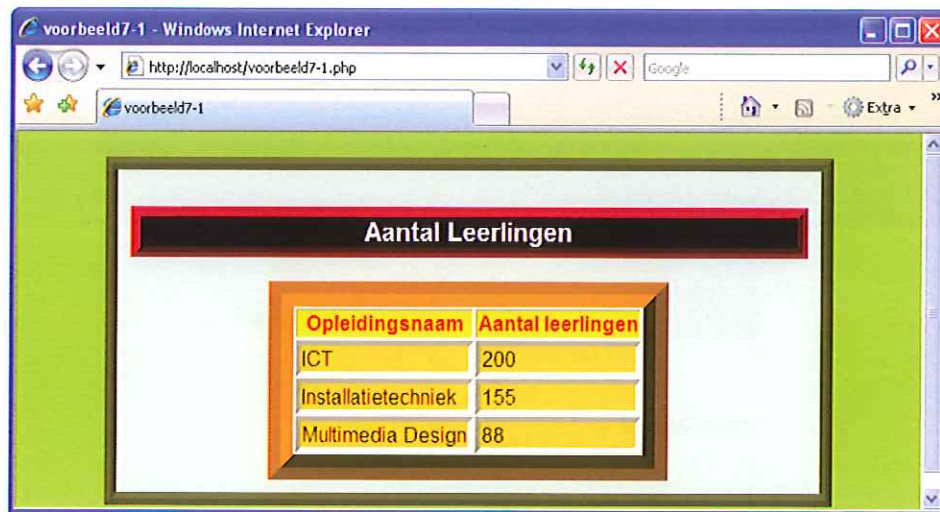
Voorbeeld 7-1

1. Open het programma Crimson Editor
2. Typ in.

```
<html>
<head>
<title>voorbeeld7-1</title>
<LINK rel=STYLESHEET href="opmaak.CSS" Type="text/css">
</head>
<body>
<div id= "tekstblok">
<?php
    $opleiding = array(); // nieuw array
    $opleiding[0][0]="ICT";
    $opleiding[2][0]="Multimedia Design";
    $opleiding[1][0]="Installatietechniek";

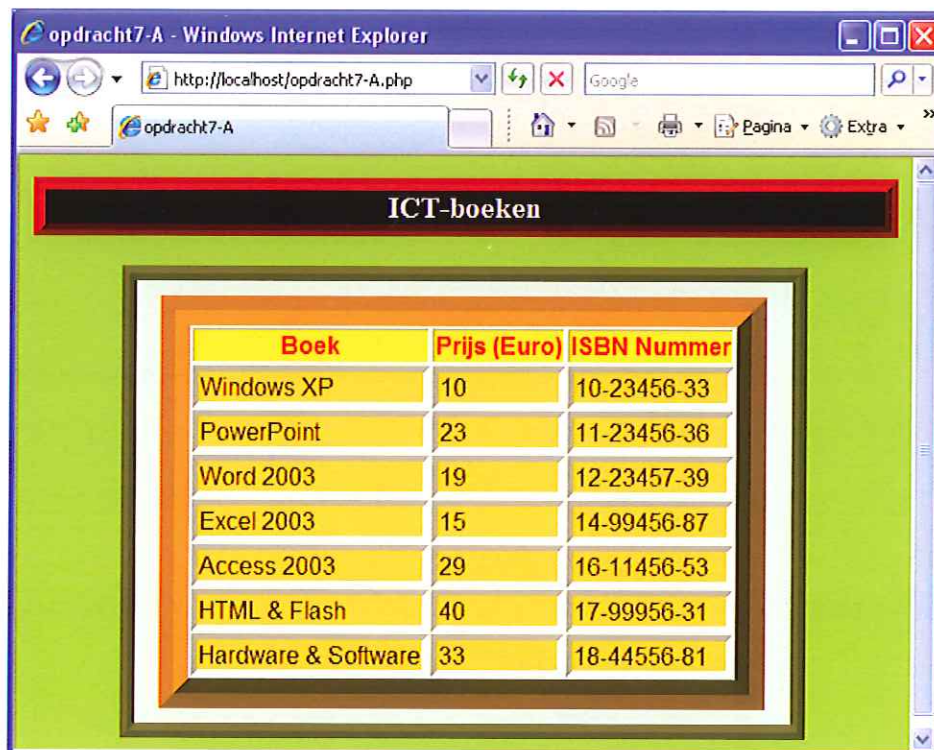
    $opleiding[0][1]=200;
    $opleiding[1][1]=155;
    $opleiding[2][1]=88;
    echo "<H2>Aantal Leerlingen</H2>";
    echo "<center><table border=2>";
    echo "<tr><th><b>Opleidingnaam</b></th><th><b>Aantal leerlingen</b></th></tr>";
    for($rij=0;$rij<3;$rij++){
        echo "<tr>";
        for($kolom=0;$kolom<2;$kolom++){
            echo "<td>";
            echo $opleiding[$rij][$kolom];
            echo "</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
?>
</body>
</html>
```

3. Sla de tekst op onder de naam **Voorbeeld7-1.php** (in de map C:\wamp\www)
4. Open het bestand Voorbeeld7-1.php. <http://localhost/Voorbeeld7-1.php>, dan zie je het volgende venster:



Opdracht7-A

1. Maak de onderstaande webpagina zo nauwkeurig mogelijk na.
2. Sla de pagina op onder de naam: **Opdracht7-A.php**



3. Wat moet je voor de opdracht inleveren:
- **Lever** deze opdracht in als Worddocument, want voor jouw docent zijn de codes in een Worddocument gemakkelijker na te kijken.
 - **Kopieer** de broncode van de opdracht naar dit Worddocument.
 - **Maak** een Print Screen van jouw opdracht en plak deze onder de broncode in het Worddocument.
 - **Vraag** jouw docent op welke manier (e-mail of print) je de opdracht moet inleveren.

7.2 Array's sorteren

We kunnen een array sorteren. Er bestaan veel verschillende sorteerfuncties en allemaal doen ze iets anders. De meest gebruikte zijn **"sort()"**, **"asort()"** en **"arsort()"**

"sort()": rangschikt de elementen, maar de indexen veranderen **niet mee**.

"asort()": verplaatst ook de indexen.

"arsort()": sorteert een array in omgekeerde volgorde en bewaart index associatie.

"reset()": verplaatst de interne pointer van array naar het eerste element en geeft de waarde ervan terug of FALSE wanneer de array leeg is.

"print_r()": Print leesbare informatie over een variabele uit. Als een string, **integer** of **float** wordt gegeven, zal de waarde ervan worden geprint. Als je een array meegeeft, zullen de waarden worden geprint in een opmaak met daarin sleutels en waarden. Voor objecten wordt eenzelfde notatie gebruikt.

Denk eraan dat **print_r()** de array pointer naar het einde van de array zal verplaatsen. Gebruik **reset()** om hem terug te brengen naar het begin.

"<PRE> tekst </PRE>": Met behulp van het **< PRE >** element kan tekst in een vaste opmaak (preformatted) worden opgenomen, zonder dat de browser daar iets aan mag veranderen. Bijvoorbeeld als je iets in tabelvorm moet opnemen, maar geen echte tabel wilt gebruiken.

Zie de volgende voorbeeld (voorbeeld7-2).

Voorbeeld7-2: sort(), Arsort(), asort(), reset, print_r en pre:

1. Open het programma Crimson Editor
2. Typ in.

```
<html>
<head>
<title>Voorbeeld7-2</title>
<LINK rel=STYLESHEET href="opmaak.css" Type="text/css">
</head>
<body>
<h2>Vakken sorteren: asort()</h2>
<?php
$Vak["a"] = "Wiskunde";
$Vak["d"] = "Scheikunde";
$Vak["c"] = "Natuurkunde";
$Vak["b"] = "Engels";
asort($Vak);
reset($Vak);
while(list($index, $naam)=each($Vak)){
echo "<center>$index = $naam <br></center>";
}
?>
<h2>Vakken sorteren: arsort()</h2>
<?php
arsort($Vak);
while(list($index, $naam)=each($Vak)){
echo "<center>$index = $naam <br></center>";
}
?>
<h2>steden sorteren: sort():Top 7 Inwoneraantal</h2>
<?php
$sort1 = array('Rotterdam', 'Tilburg', 'Groningen', 'Amsterdam', 'Den Haag', 'Utrecht',
'Eindhoven');
$sort2 = array(333, 12, 1, 35, 114, 222, 550);
sort($sort1);
sort($sort2, SORT_NUMERIC);
echo"<pre>";
print_r($sort1);
echo"<h2>Numeric sorteren: sort():Sort_numeric</h2>";
print_r($sort2);
echo "<pre>";
?>
</body>
</html>
```

3. Sla de tekst op onder de naam **Voorbeeld7-2.php** (in de map C:\wamp\www)
4. Open het bestand Voorbeeld7-2.php. <http://localhost/Voorbeeld7-2.php>, dan zie je het volgende venster:

```

Vakken sorteren: asort()

b = Engels
c = Natuurkunde
d = Scheikunde
a = Wiskunde

Vakken sorteren: arsort()

a = Wiskunde
d = Scheikunde
c = Natuurkunde
b = Engels

steden sorteren: sort():Top 7 Inwoneraantal

Array
(
    [0] => Amsterdam
    [1] => Den Haag
    [2] => Eindhoven
    [3] => Groningen
    [4] => Rotterdam
    [5] => Tilburg
    [6] => Utrecht
)

Numeric sorteren: sort():Sort_numeric

Array
(
    [0] => 1
    [1] => 12
    [2] => 35
    [3] => 114
    [4] => 222
    [5] => 333
    [6] => 550
)

```

7.3 Tellen in array's

count(), min() en max()

Met behulp van de functie **count()** kunnen we bijvoorbeeld onderzoeken hoeveel groepen ICTboeken er zijn.

Met behulp van de functie **min()** kunnen we bijvoorbeeld de minimale aantal ICTboeken onderzoeken.

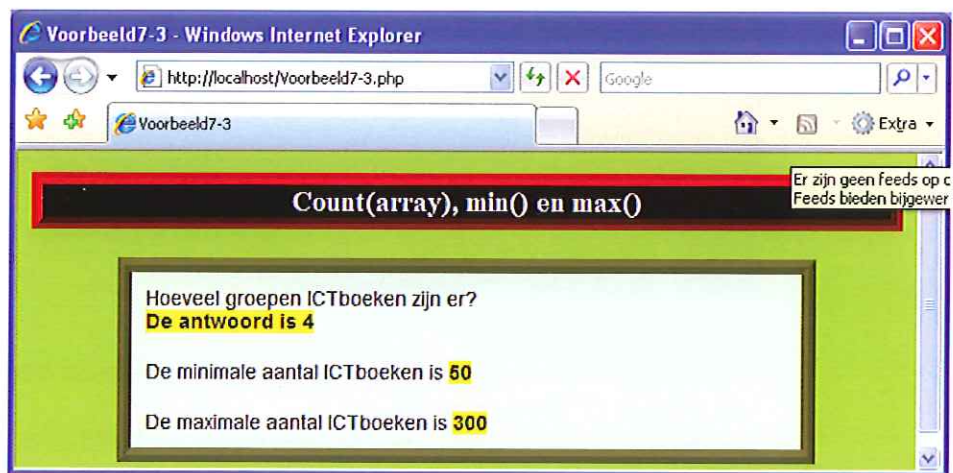
Met behulp van de functie **max()** kunnen we bijvoorbeeld de maximale aantal ICTboeken onderzoeken

Voorbeeld7-3: count() , min() en max()

1. Open het programma Crimson Editor
2. Typ in.

```
<html>
<head>
<title>Voorbeeld7-3</title>
<LINK rel=STYLESHEET href="opmaak.css" Type="text/css">
</head>
<body>
<h2>Count(array), min() en max()</h2>
<div id="tekstblok">
<?php
$ICTboek[0] = 50;
$ICTboek[1] = 60;
$ICTboek[2] = 100;
$ICTboek[3] = 300;
$antwoord= count($ICTboek);
$min= min($ICTboek);
$max=max($ICTboek);
echo "Hoeveel groepen ICTboeken zijn er?<br>";
echo "<b>De antwoord is $antwoord</b><p>";
echo "De minimale aantal ICTboeken is <b>$min</b><p>";
echo "De maximale aantal ICTboeken is <b>$max</b> ";
?>
</body>
</html>
```

3. Sla de tekst op onder de naam **Voorbeeld7-3.php** (in de map C:\wamp\www)
4. Open het bestand Voorbeeld7-3.php. <http://localhost/Voorbeeld7-3.php> , dan zie je het volgende venster:



7.4 Array's aflopen

We kunnen met behulp van een lus een bepaalde array aflopen, maar dan mogen we **geen strings gebruiken als index**.

Met behulp van de functie **count()** kunnen we onderzoeken hoeveel indexen er reeds in gebruik zijn.

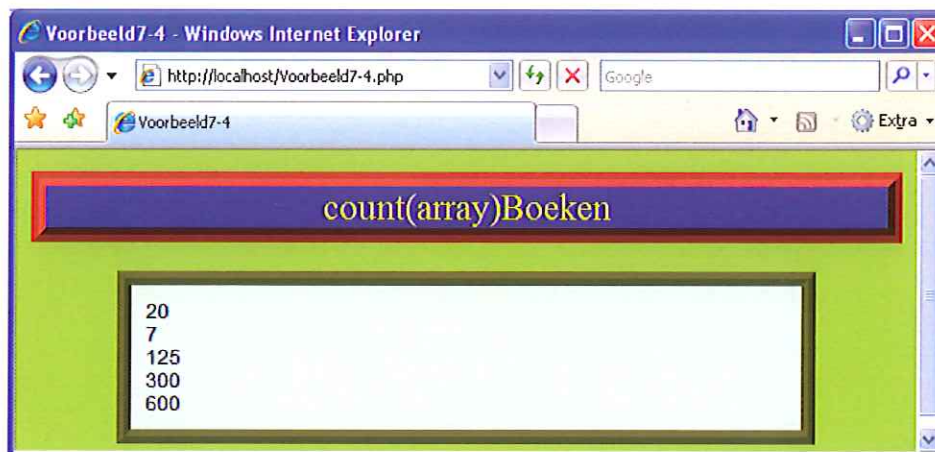
We kunnen dan enkel maar hopen dat deze indexen sequentieel ingevuld zijn.

Voorbeeld7-4: count()

1. Open het programma Crimson Editor
2. Typ in.

```
<html>
<head>
<title>Voorbeeld7-4</title>
<LINK rel=STYLESHEET href= "opmaak.css" Type= "text/css">
</head>
<body>
<h1>count(array)Boeken</h1>
<div id= "tekstblok">
<?php
$Boek[] = 20;
$Boek[] = 7;
$Boek[] = 125;
$Boek[] = 300;
$Boek[] = 600;
for($i=0; $i < count($Boek); $i++)
{
echo $Boek[$i] . "<BR>";
}
?>
</body>
</html>
```

3. Sla de tekst op onder de naam **Voorbeeld7-4.php** (in de map C:\wamp\www)
4. Open het bestand Voorbeeld7-4.php. <http://localhost/Voorbeeld7-4.php> , dan zie je het volgende venster:



7.5 List: "List()"="each()"

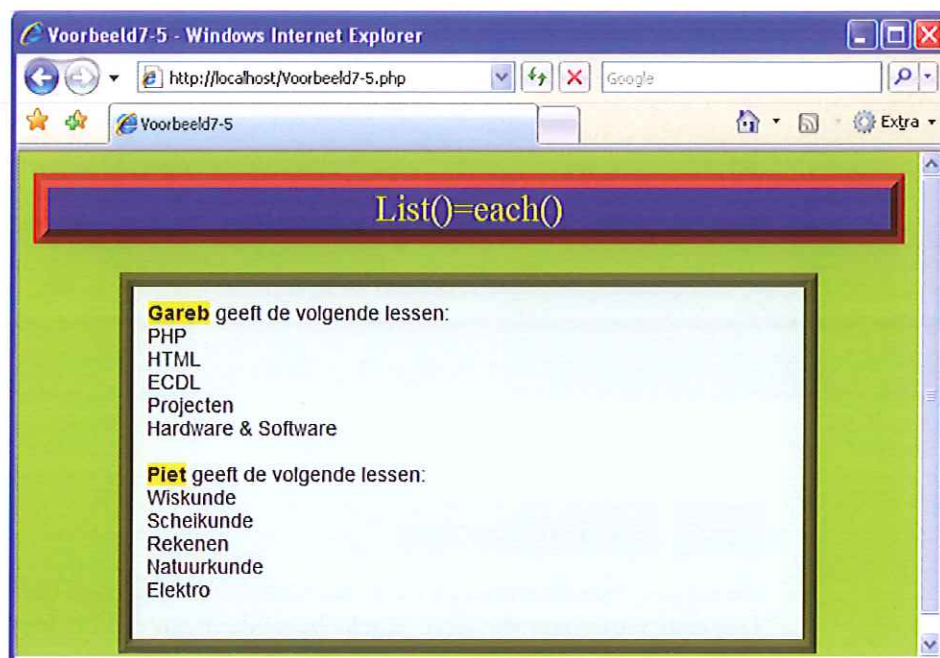
Wanneer we echter strings gebruikt hebben, is deze constructie niet te gebruiken. Hiervoor moeten we de **list()=each()** constructie gebruiken. Deze constructie gaat de array overlopen en telkens de index en de waarde teruggeven. Deze methode kunnen we ook genest gebruiken om meerdimensionele arrays te overlopen. Het voorbeeld zal volstaan om de eenvoudige syntax te begrijpen.

Voorbeeld7-5: List()=each()

1. Open het programma Crimson Editor
2. Typ in.

```
<html>
<head>
<title>Voorbeeld7-5</title>
<LINK rel=STYLESHEET href="opmaak.css" Type="text/css">
</head>
<body>
<h1>List()=each()</h1>
<div id="tekstblok">
<?php
$les["Gareb"][] = "PHP";
$les["Gareb"][] = "HTML";
$les["Gareb"][] = "ECDL";
$les["Gareb"][] = "Projecten";
$les["Gareb"][] = "Hardware & Software";
$les["Piet"][] = "Wiskunde";
$les["Piet"][] = "Scheikunde";
$les["Piet"][] = "Rekenen";
$les["Piet"][] = "Natuurkunde";
$les["Piet"][] = "Elektro";
while(list($naam,$lessen) = each($les)){
echo "<b>$naam</b> geeft de volgende lessen:<br>";
while(list($index,$lesnaam) = each($les[$naam]))
echo $lesnaam."<br>";
echo "<br>";
}
?>
</body>
</html>
```

3. Sla de tekst op onder de naam **Voorbeeld7-5.php** (in de map C:\wamp\www)
4. Open het bestand Voorbeeld7-5.php. <http://localhost/Voorbeeld7-5.php>, dan zie je het volgende venster:



Opdracht7-B:

1. Bedenk een opdracht. Gebruik de mogelijkheden binnen "Array"
2. Sla de webpagina onder de naam **Opdracht7-B.php** op.
3. Wat moet je voor de opdracht inleveren:
 - **Lever** deze opdracht in als Worddocument, want voor jouw docent zijn de codes in een Worddocument gemakkelijker na te kijken.
 - **Kopieer** de broncode van de opdracht naar dit Worddocument.
 - **Maak** een Print Screen van jouw opdracht en plak deze onder de broncode in het Worddocument.
 - **Vraag** jouw docent op welke manier (e-mail of print) je de opdracht moet inleveren.

8

BESTANDEN EN MAPPEN
BEHEREN

8.1 Bestand openen

De **fopen()** functie wordt gebruikt om bestanden te openen in PHP. De eerste parameter van deze functie bevat de naam van het bestand dat geopend moet worden. De tweede parameter specificeert in welke modus het bestand geopend moet worden:

```
<?php
$file=fopen("leerlingen.txt","r");
?>
```

Het bestand kan in één van de onderstaande manieren worden geopend:

Modes	Betekenis
r	Alleen lezen. Start aan het begin van het bestand.
r+	Lezen/Schrijven. Start aan het begin van het bestand
w	Alleen schrijven. Opent en wist de inhoud van het bestand; of maakt nieuw bestand als het niet bestaat.
w+	Lezen/Schrijven. Opent en wist de inhoud van het bestand; of maakt nieuw bestand als het niet bestaat.
a	Toevoegen. Opent en schrijft aan het einde van het bestand, of maakt nieuw bestand als het niet bestaat.
a+	Lezen/Toevoegen. Bewaard inhoud door aan het einde van het bestand te schrijven.
x	Alleen Schrijven. Maakt een nieuw bestand. Geeft FALSE terug en een error als het bestand al bestaat.
x+	Read/Write. Creates a new file. Returns FALSE and an error if file already exists

Als de **fopen()** functie een bepaald bestand niet kan openen, zal er **0 (false)** te zien zijn (terug worden gegeven).