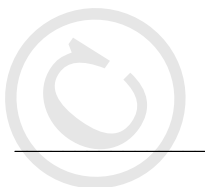


Student-notities
PHP en MySQL
Voorbeeld-hoofdstuk

02. Control structures en arrays

 **at computing**
The Linux/UNIXperts

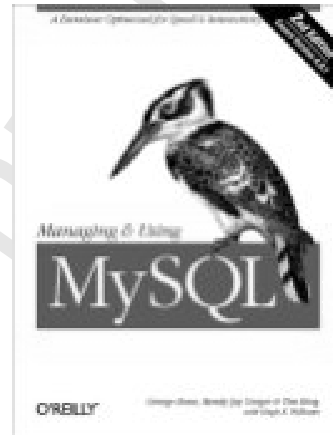
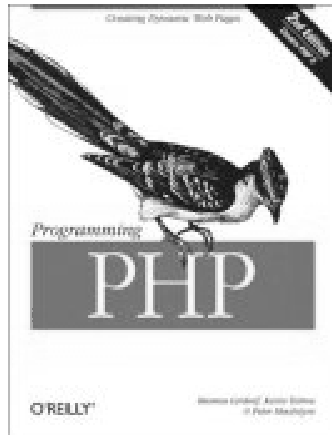
Nijmegen



Copyright © AT Computing 2004
Versie: 1e

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



If-then-else

```
<?php
    if (expressie) {
        hier php-statement(s);
    } else {
        hier php-statement(s);
    }
?>
```

```
<?php if (expressie) { ?>
```

```
    hier HTML-code
```

```
<?php } else { ?>
```

```
    hier HTML-code
```

```
<?php } ?>
```

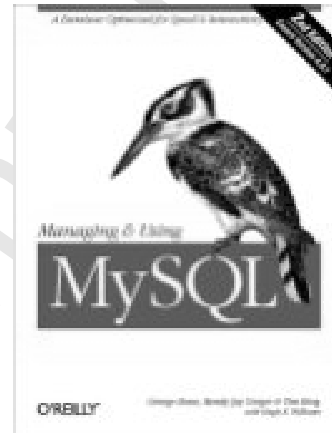
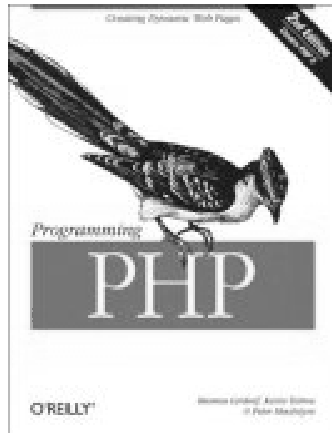
```
<?php if (expressie): ?>
    hier HTML-code
<?php else: ?>
    hier HTML-code
<?php endif; ?>
```

- **else**-tak mag altijd wegblijven.
- Bij enkelvoudig statement accolades niet nodig
Advies: *altijd* accolades (of : vorm) gebruiken!!
- Bij de : vorm altijd **endif** afsluiting nodig.
- Er bestaat ook een **elseif**

Figuur 1.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



Switch - case

```
<?php
  switch ($variabele) {

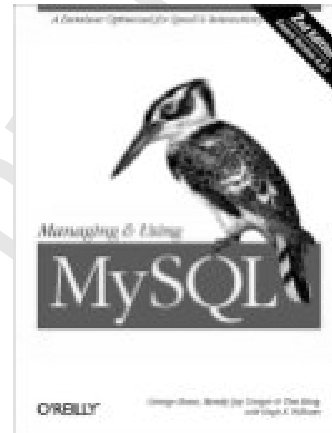
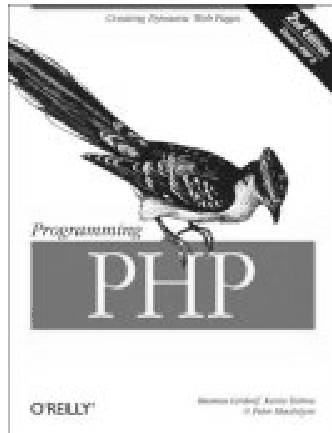
    case constanteA:
      hier php-statement(s);
      break;
    case constanteB:
      hier php-statement(s);
      break;
    case constanteC: case constanteD:
      hier php-statement(s);
      break;
    case constanteE:
      hier php-statement(s); // hier fall through
    case constanteF:
      hier php-statement(s);
      break;
    default: // default niet verplicht
      hier php-statement(s);
      break; // laatste break niet verplicht
  }
?>
```

- Waarde van **\$variabele** bij binnenkomst bepaalt de keuze van **case**-ingang
- Geen **break** betekent "fall through"
Dit is geen bug; het is een feature!
Maakt meerdere cases op 'n hoop mogelijk.
- Er is ook weer een versie met **:** i.p.v. **{** en **endswitch;** als afsluiter (vergeet de **;** niet).
- HTML-code buiten **<? .. ?>** mag natuurlijk ook

Figuur 2.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



Foreach-loop

```
<?php
  foreach ($array as $kladvalue) {
    hier php-statement(s) waarin de
    kladvariabele mag worden gebruikt
  }
?>
```

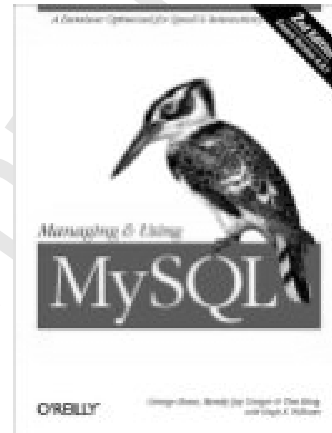
- Inhoud van **\$array** bij binnenkomst bepaalt aantal slagen door de loop
- **kladvalue** krijgt (automatisch) steeds de waarde van een ander array-element als inhoud
- Elk array-element komt eenmaal aan de beurt
- Volgorde is onbepaald (als array-inhoud verandert, verandert de volgorde ook).
- Inhoud van **\$array** wijzigen binnen de loop-body is niet slim (loop werkt met een kopie!)
- Versie met **:** en met **endforeach;** als afsluiter

```
<?php
  foreach ($array as
           $kladkey => $kladvalue) {
    .....
  }
?>
```

Figuur 3.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



While-loop

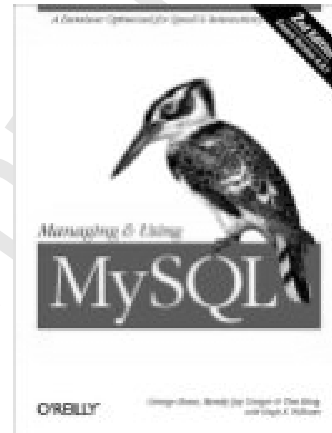
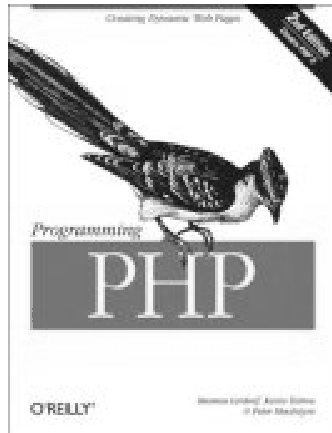
```
<?php
  while (expressie) {
    hier php-statement (s);
  }
?>
```

- Zo lang de **expressie** true blijft opleveren...
- **break**; springt uit de (nauwst omsluitende) loop
- **continue**; springt terug naar de while-test, dus slaat de staart van het body-deel over.
- Er is weer een versie met : i.p.v. { afsluiter daarbij is **endwhile**;
- Bij enkel statement accolades niet verplicht
- HTML-code ook mogelijk buiten **<?php ?>**
- Er is ook een
`do { } while (expressie);`

Figuur 4.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



For-loop

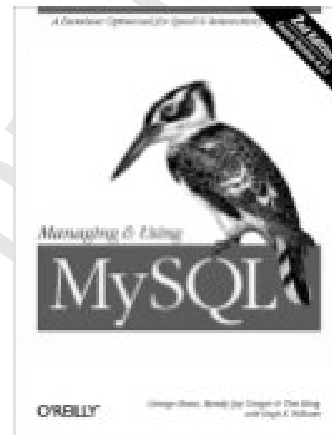
```
<?php
    for( initialisatie ; test; stap ) {
        hier php-statement(s);
    }
?>
```

- Eerst de initialisatie
Dan meteen de eerste test
Indien true: ga dan door de body-statements
Dan de stap-berekening
Dan opnieuw de test enz.
- Voorbeeld:
`for ($i = 0; $i < 10; $i++) { ... }`
- Bestaansrecht: de hele besturing staat bij elkaar; en dat maakt de zaak beter leesbaar.
- **continue** en **break** net als bij while-loop
- Versie met `:` en met **endfor**; als afsluiter
- HTML-code buiten `<? .. ?>` mag natuurlijk ook
- Initialisatie en stap mogen leeg blijven als je die elders regelt. Puntkomma's moeten blijven staan!
- Test mag ook leeg (altijd true) als je bijv. de loop met een **break** wilt verlaten. Let op leesbaarheid!

Figuur 5.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



Arrays - 1

Opslag van reeks van waarden:

```
index + value: $vrienden[0] = 'Jan';  
key + value:   $mijn_auto['kleur'] = 'rood';
```

Onder de motorkap hebben de elementen een "interne" volgorde (hashed).

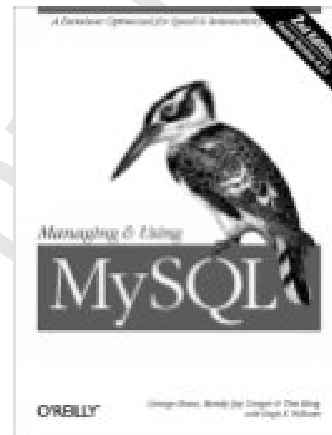
Maken kan ook met functie *array*:

```
$neefjes = array('Kwik', 'Kwek', 'Kwak');  
  
$ducks = array($neefjes, 'Donald'); ???  
$ducks[4] = 'Katrien';  
$ducks[] = 'Dagobert';  
  
$auto =  
    array('kleur'=>'rood', 'jaar' => 1971);  
$auto['merk'] = 'Ford';  
  
$doos1 = array('hamer', 'spijker');  
$doos2 = array('koffie', 'melk', 'suiker');  
$overdoos = array($doos1, $doos2);  
  
$ding = $overdoos[1][2];  
$ding = $overdoos[4]; ???
```

Figuur 6.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



Arrays - 2

Notatie van index (key)-strings is tricky...

- Quotes verplicht bij multi-word keystrings:

```
$auto['merk'] = 'Ford';  
$auto[kleur] = rood;  
$auto['maat banden'] = '75x215 Rad';
```

advies: altijd quotes gebruiken;
voorkom verwarring met namen van constanten.

- Geen quotes toegestaan bij key van een array-element dat in een string moet worden geïnterpoleerd:

```
echo "Kleur is: $auto[kleur]";
```

- Dubbelquotes verplicht bij keystring die via interpolatie moet worden samengesteld:

```
$auto['leeggewicht'] = '2350 kg';  
$auto['maxgewicht'] = '3500 kg';  
$hoe = 'leeg';  
echo "Hij weegt $hoe: "  
    $auto["${hoe}gewicht"];
```

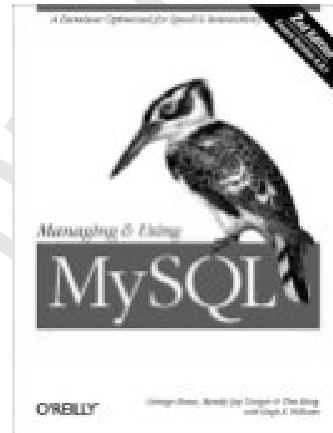
- Dus een conflict geeft:

```
echo "Weegt: $auto["${hoe}gewicht"]";
```

Figuur 7.

Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Programming PHP*" (van de auteurs R. Lerdorf, K. Tatroe en P. MacIntyre) en "*Managing & Using MySQL*" (van de auteurs G. Reese, R. Yarger en T. King) waarin deze materie verder is uitgewerkt.



Array-functies

- `foreach($mijn_array as $waarde)...`
`foreach($mijn_array as $key => $waarde)...`
Reeds behandeld
- Aantal elementen: `count($mijn_array);`
- Test op aanwezigheid (leveren true/false):
`in_array(waarde, $mijn_array);`
`array_key_exists(key, $mijn_array);`
`isset($mijn_array[key]); // waarde != NULL`
- Element verwijderen:
`$eruit = array_splice($mijn_array,`
`startpunt [, lengte], vervanging);`
splitsing ligt vóór het startpunt
startpunt opgeven als index-getal
default-lengte: hele staart weg
vervanging komt i.p.v. verwijderde deel
lengte 0 betekent: invoegen i.p.v. vervangen
- `while(list($sleutel, $waarde) =`
`each($mijn_array))..`
`while` gaat door het hele array
`each`-functie levert apart (nieuw) array per element
uit het oude array
`list`-functie splitst daarvan key en value weer uit
tot twee aparte variabelen
`reset($mijn_array)` initialiseert de `each`-functie

Figuur 8.

