

PHP-kielen perusteiden tiivistelmä

Tämä luku sisältää PHP-kielen perusteet erittäin tiiviisti ilmaistuna. Oletuksena on, että lukijalla on esitietoina tavanomaisen ohjelmoinnin perusteet - opintojakson sisältö.

PHP-koodin merkintä ja kommentit

PHP-ohjelmakoodi merkitään (upotetaan) HTML-koodin sekaan tavanomaisesti seuraavalla XML-prosessointiohjeella

```
1 <h1>HTML-merkkausta</h1>
2 <?php
3 echo "XML-prosessointiohje on suositeltavin";
4 ?>
5 <p>HTML-merkkaus jatkuu</p>
```

Kommentit merkitään seuraavasti

```
1 <?php
2 echo "Testiteksti"; // Rivin loppuosa kommenttia
3
4 /* Tämä on
5     monirivinen
6     kommentti */
7
8 echo "Testiteksti"; # Rivin loppuosa kommenttia
9 ?>
```

Muuttujat

- Muuttujiin viitataan tunnuksella `$muuttuja` ja dollarimerkki muuttujan nimen alussa on pakollinen

PHP on heikosti (dynaamisesti) tyyplitetty kieli eli

- Muuttujia ei tarvitse luoda erikseen
- Muuttujan tila varataan, kun sitä käytetään ensimmäisen kerran (esim. sijoituslause)
- Muuttujille ei määritellä tietotyyppiä
- Muuttujaan voi tallentaa mitä tahansa tietotyyppiä olevia arvoja
- Muuttujan tietotyyppi riippuu tallennetusta arvosta ja saman muuttujan tietotyyppi voi vaihdella ohjelman suorituksen aikana.

Esimerkkejä muuttujista

```

1 | $muuttujanArvo = "Ari";
2 | $muuttujanarvo = "Jari";
3 | $muuttujan_arvo = "Kari";
4 | $kokolukumuuttuja = 42;
5 | $liukulukumuuttuja = 42.18;
6 | $totuus = true;

```

Taulukkomuuttujat

- Taulukon alkioon viitataan joko kokonaislukuindeksillä tai merkkijonoavaimella

```

1 | $taulu[0] = "Eka";
2 | $taulu[1] = "Toka";
3 | $taulu["sukunimi"] = "Rantala";
4 | $taulu["etunimi"] = "Ari";

```

Taulukon alustaminen sijoittamalla

```

1 | $ruoka[] = "Tomaatti"; // $ruoka[0] = "Tomaatti"
2 | $ruoka[] = "Pinaatti"; // $ruoka[1] = "Pinaatti"
3 | $ruoka[] = "Peruna"; // $ruoka[2] = "Peruna"
4 | $ruoka[] = "Salaatti"; // $ruoka[3] = "Salaatti"

```

Taulukon alustaminen array-käskyllä

```

1 | $ruoka = array("Tomaatti", "Pinaatti", "Peruna", "Salaatti");
2 | print_r($taulu); // print_r tulostaa taulukon avaimineen
3 |
4 | $menu = array("alkuruoka" => "Salaatti",
5 |             "paaruoka" => "Pihvi",

```

```
6         "jalkiruoka" => "Kiisseli");
7     print_r($menu); // print_r tulostaa taulukon avaimineen
```

Käyttäessäsi merkkijonoa avaimena kirjoita aina heitto- tai lainausmerkit näkyviin, jollet tiedä miksi ne voisi jättää pois!

Taulukot voivat olla moneen suuntaan moniulotteisia esim. seuraavat ovat ok

```
1     $taulu[0]['nimi'] = "Ari";
2     $taulu[1]['nimi'] = "Sepi";
3     $taulu['nimi'][2][10] = "Hanski";
```

Etukäteen määritellyt muuttujat

- PHP-skripteillä on käytössä ympäristöstä riippuva määrä erilaisia etukäteen määriteltyjä muuttujia. Ympäristöön vaikuttaa käyttöjärjestelmä, web-palvelin, PHP ja niiden eri asennustavat ja versiot. Näistä muuttujista osan luo web-palvelin ja osan PHP itse. Täyden listan oman ympäristösi etukäteen määritellyistä muuttujista saat tulostettua funktiolla `phpinfo()`. Tässä on lyhyesti listattuna muutamia.
- Nämä etukäteen määriteltujen muuttujien arvot ovat käytettävissä nk. superglobaaleissa assosiatiivisissa (merkkijonoavaimin viitattavissa) taulukoissa -> Arvot käytettävissä funktioissa ilman eri määrittelyjä.

\$_GET

Muuttujat, jotka ovat välitetty skriptille GET-metodilla. Esim. `$_GET['sukunimi']`

\$_POST

Muuttujat, jotka ovat välitetty skriptille POST-metodilla. Esim. `$_POST['sukunimi']`

\$_COOKIE

Muuttujat, jotka ovat välitetty skriptille evästeinä. Esim. `$_COOKIE['login']`

\$_REQUEST

Muuttujat, jotka ovat välitetty skriptille GET- ja POST-metodeilla tai evästeinä.

\$_SESSION

Muuttujat, jotka ovat välitetty skriptille käyttäen PHP:n sisäänrakennettua istunnonhallintamekanismia. Esim. `$_SESSION['login']`

\$_FILES

Muuttujat, jotka ovat välitetty skriptille File Upload-menetelmällä. Esim.
\$_FILES['filetto']

\$_SERVER

web-palvelimen generoimia muuttujia

\$_ENV

Muuttujat, jotka skripti on perinyt ajoympäristöstänsä.

Ajamalla seuraavan ohjelman saat hyvän käsityksen siitä mitä kaikkea em. muuttujilla on tarjota

```
1 | <title>PHP-esimerkki phpinfo.php</title>
2 | <?php
3 | phpinfo();
4 | ?>
```

Tietotyypit

Totuusarvo

```
1 | $mja = False; // Sijoitetaan totuusarvo FALSE muuttujalle $mj
2 | $mja = TRUE; // Merkkikoolla ei ole väliä,
```

Kun jonkin muun tyyppin muuttujaa muutetaan totuusarvoksi, niin seuraavat arvot muunnetaan boolean-tyypin FALSE-arvoksi:

- boolean: FALSE
- integer: 0
- float: 0.0
- string: tyhjä merkkijono ja merkkijono "0"
- array: tyhjä taulukko, ei alkioita
- object: olio ilman asetettuja ominaisuuksia
- NULL

Kaikki muut arvot muutetaan TRUE-arvoksi!

Kokonais- ja liukuluvut

```
1 $mja = -12; // -12 kymmenjärjestelmässä
2 $mja = 012; // Oktaaliku 12 (kymmenjärjestelmässä 10)
3 $mja = 0x12; // heksadesimaaliluku 12 (kymmenjärjestelmässä 18)
4 $mja = 1.2;
5 $mja = 1.2e3; // 1200
6 $mja = 3E-5; // 0.00003
```

Merkkijonot

Kolme tapaa määritellä

```
1 $str1 = 'Tämä on heittomerkein määritelty merkkijono';
2 $str2 = "Tämä on lainausmerkein määritelty merkkijono";
3
4 $str3 = <<<EOD
5 Tässä on
6 kolmas kolmirivinen
7 merkkijono heredoc-syntaksilla.
8 EOD;
```

Esimerkit

```
1 // Heittomerkit
2 $mja = 36;
3 $str = 'James O'Neill is $mja years old';
4 echo $str; //Tulostaa: James O'Neill is $mja years old
5
6 // Lainausmerkit
7 $mja = 54;
8 $_POST['nimi'] = "Ari";
9
10 // Seuraavat rivit tulostavat kumpikin
11 // Ari on 54-vuotias
12 echo $_POST['nimi'] . " on " . $mja . "-vuotias";
13 echo "{$_POST['nimi']} on $mja-vuotias";
14
15 // Heredoc-syntaksi
16 $mja = 54;
17 $_POST['nimi'] = "Ari";
18
```

```
19 $lomake = <<<EOLomake
20 <form action="softa.php" method="get">
21 <input type='text' name="nimi" value="{$_POST['nimi']}">
22 <input type='text' name="ika" value="$mja">
23 <input type='submit' name="nappi">
24 </form>
25 EOLomake;
26
27 // Tulostaa lomakkeen muuttujien arvoineen
28 echo $lomake;
```

Operaattorit

Aritmeettiset operaattorit

```
1 $a+$b;
2 $a-$b;
3 $a*$b;
4 $a/$b;
5 $a%$b;
6
7 // Tulostaa: 4
8 $a=3; echo ++$a;
9
10 // Tulostaa: 3
11 $a=3; echo $a++;
12
13 // Tulostaa: 2
14 $a=3; echo --$a;
15
16 //Tulostaa: 3
17 $a=3; echo $a--;
```

Sijoitusoperaattorit

```
1 $a = 36;
2 $a += 4; //kuten $a=$a+4;
3 $a -= 4; //kuten $a=$a-4;
4 $a *= 4; //kuten $a=$a*4;
5 $a /= 4; //kuten $a=$a/4;
```

```

6  $a %= 4; //kuten $a=$a%4;
7
8  $s1 = "Aku";
9  $s1 .= " Ankka";
10 //Tulostaa: Aku Ankka
11 echo $1;

```

Vertailuoperaattorit

```

1  ("36" == 36) // TRUE
2  ("36" === 36) // FALSE
3  (12 < 36) // TRUE
4  (12 > 36) // FALSE
5  (36 <= 36) // TRUE
6  (36 >= 36) // TRUE
7  (12 <> 36) // TRUE
8  (12 != 36) // TRUE
9  (12 !== 36) // TRUE
10 ("36" !== 36) // TRUE

```

Loogiset operaattorit

```

1  AND ja &&
2  OR ja ||
3  XOR
4  !

```

Muut operaattorit

```

1  $str = "No" . " hei!";
2  echo $str; // "No hei!"
3
4  print(36/0); //Tulostaa virheen
5  @print(36/0); //Ei tulosta mitään
6
7  $str = `netstat -an`;
8  echo "<pre>$str</pre>";

```

Aliohjelmat eli funktiot

alv-esimerkki.php

```
1 <title>ALV-esimerkki</title>
2 <?php
3 // Pääohjelma, ohjelman suoritus alkaa tästä
4 $alviton_loppusumma = 100;
5
6 // Funktion anna_alv kutsu:
7 $arvonlisavero = anna_alv($alviton_loppusumma);
8
9 // Lisätään ja tulostetaan arvonlisäverollinen hinta
10 $alvillinen_hinta = $alviton_loppusumma + $arvonlisavero;
11 echo "ALVillinen hinta on: $alvillinen_hinta"; // Tulostaa
12
13 // Funktion anna_alv määrittely
14 function anna_alv($maara) {
15     $alv = $maara * 0.24;
16     return $alv;
17 }
18 ?>
```

muuta-saldoa.php

```
1 <title>Parametrin välittäminen viittauksena -esimerkki</title>
2 <?php
3 // muuta-saldoa.php
4
5 // Pääohjelma
6 $saldo = 100;
7 $muutos = -40;
8
9 // $saldo välitetään nyt viittauksena, koska
10 // niin on määritelty funktion esittelyssä
11 // Huomaa, että funktion palauttamaa NULL-arvoa
12 // ei käytetä (sijoiteta) mihinkään.
13
14 muuta_saldo($saldo, $muutos);
15 echo "Saldo: $saldo<br>
16 "; // tulostaa: 60
17
18 // Vähennetään $saldoa toistamiseen
```



```

19 muuta_saldo($saldo, $muutos);
20 echo "Saldo: $saldo<br>
21 ";    // tulostaa: 20
22
23 // Huomaa &-merkki muodollisen parametrin edessä: Viittaus
24 function muuta_saldo(&$local_saldo, $muutos) {
25     $local_saldo = $local_saldo + $muutos;
26     // Täällä ei ole return-lausetta, palautetaan siis NULL
27 }
28
29 ?>

```

Staattinen muuttuja

static-counter.php

```

1 <title>Staattinen muuttuja laskurina</title>
2 <?php
3 // static-counter.php
4
5 function huonoLaskuri() {
6     if (isset($count)) {
7         $count++;
8     } else {
9         $count = 0;
10    }
11    echo "huonoLaskuri: $count<br>";
12 }
13
14 function hyvaLaskuri() {
15     static $count = 0; // sijoitetaan 0 ensimmäisellä kerralla
16     $count++;
17     echo "hyvaLaskuri: $count<br>";
18 }
19
20 // Molempia funktioita kutsutaan erikseen 3 kertaa
21 for ($i = 1; $i <= 3; $i++) {
22     huonoLaskuri();
23 }
24
25 for ($i = 1; $i <= 3; $i++) {

```

```
26     hyvaLaskuri();  
27 }  
28 ?>
```

Ohjelman tulostus on seuraava:

```
huonoLaskuri: 0  
huonoLaskuri: 0  
huonoLaskuri: 0  
hyvaLaskuri: 1  
hyvaLaskuri: 2  
hyvaLaskuri: 3
```

Ohjausrakenteita

IF-esimerkki

```
1  <title>Esimerkki If-rakenteesta</title>  
2  <?php  
3  // if-esimerkki.php  
4  $nimi = "Kaveri";  
5  $ika = 36;  
6  if ($ika < 0) // Jos ika on pienempi kuin nolla {  
7      echo "$nimi ei vielä ole syntynyt  
8  ";  
9  }  
10 else if ($ika < 18) { // muutoin jos ika on pienempi kuin 18  
11     echo "$nimi ei ole täysi-ikäinen  
12 ";  
13 }  
14 else {  
15     echo "$nimi on täysi-ikäinen  
16 ";  
17 }  
18 ?>
```

Switch-esimerkki

```

1  <title>Esimerkki Switch-rakenteesta</title>
2  <pre>
3  <?php
4  // switch-esim.php
5  $n1 = 36;
6  $n2 = 10;
7  $action = "erotus";
8
9  switch ($action)
10 {
11     case "summa": // Jos muuttujan $action arvo on "summa", su
12         echo $n1, " + ", $n2, " = ", $n1 + $n2, "
13 ";
14         break;
15     case "erotus":
16         echo $n1, " - ", $n2, " = ", $n1 - $n2, "
17 ";
18         break;
19     case "tulo":
20         echo $n1, " * ", $n2, " = ", $n1 * $n2, "
21 ";
22         break;
23     case "osamaara":
24         echo $n1, " / ", $n2, " = ", $n1 / $n2, "
25 ";
26         break;
27     default:
28         echo "Virhe: Laskutoimitusta ei ole määrittelty";
29         break;
30 }
31 ?>
32 </pre>

```

Ohjelma tulostaa:

36 - 10 = 26

For-esimerkki

```

1  <title>Taulukon läpikäynti For-rakenteella</title>
2  <?php

```

```

3 //for-esimerkki.php
4
5 $ruoka[] = "Tomaatti"; //ruoka[0]
6 $ruoka[] = "Pinaatti"; //ruoka[1]
7 $ruoka[] = "Salaatti"; //ruoka[2]
8 $ruoka_lkm = count($ruoka);
9
10 for($i = 0; $i < $ruoka_lkm; $i++) {
11     print("Ruoka $i on
12     $ruoka[$i]<br>
13 ");
14 }
15 ?>

```

Foreach-esimerkit

```

1 <title>Taulukon läpikäynti Foreach-rakenteen avulla</title>
2 <?php
3 //foreach-esim1.php
4
5 $taulu[0] = "Rantala";
6 $taulu[1] = "Rasku";
7 $taulu[2] = "Rautanen";
8
9 // Ei tarvitse välittää lukumäärästä:
10 foreach ($taulu as $arvo) {
11     echo "$arvo<br>";
12 }
13
14 ?>

```

```

1 <title>Hajautustaulukon läpikäynti Foreach-rakenteen avulla</
2 <?php
3 // foreach-esim2.php
4
5 $taulu["sukunimi1"] = "Rantala";
6 $taulu["sukunimi2"] = "Rasku";
7 $taulu["sukunimi3"] = "Rautanen";
8

```

```

9  foreach ($taulu as $arvo) {
10     echo "$arvo<br>";
11 }
12 ?>

```

```

1  <title>2-ulotteisen taulukon läpikäynti Foreach-rakenteella</
2  <pre>
3  <?php
4
5  // foreach-esim3.php
6
7  // Tulostetaan 2-ulotteisen taulukon kaikki arvot
8  // foreach-rakenteella
9
10 $taulu[0]["nimi"]      = "Rantala";
11 $taulu[0]["gsmnro"]    = "040 001";
12 $taulu[0]["paikkakunta"] = "Jyväskylä";
13
14 $taulu[1]["nimi"]      = "Rasku";
15 $taulu[1]["gsmnro"]    = "040 002";
16
17 foreach ($taulu as $uloinavain => $sistaulu) {
18     echo "***** Henkilön nro $uloinavain tiedot: *****";
19     // Käydään läpi sisemmät taulut vuorollaan:
20     foreach ($sistaulu as $avain => $arvo) {
21         echo " $avain = $arvo";
22     }
23 }
24
25 ?>
26 </pre>

```

- foreach siirtää aluksi taulukon sisäisen osoittimen automaattisesti alkuun (ei tarvita reset-funktiota)
- foreach käyttää alkuperäisen taulukon kopiota. Tällöin taulukon läpikäynti ei muuta alkuperäisen taulukon sisäisen osoittimen paikkaa

While-esimerkki

```

1  <?php
2  $lkm = 1;
3  $kerrat = 10;
4
5  while ($lkm <= $kerrat) {
6      echo "Terve $lkm kerran<br>
7      ";
8      $lkm++;
9  }
10
11  ?>

```

Do-While -esimerkki

```

1  <?php
2  $lkm = 1;
3  $kerrat = 10;
4  do {
5      echo "Terve $lkm kerran<br>
6      ";
7      $lkm++;
8  } while ($lkm <= $kerrat);
9  ?>

```

Break -lause

- Break-lause lopettaa toistorakenteen toiston
- Esim. seuraava ohjelma tulostaa "vain" neljä kertaa "Terve...":

```

1  <?php
2  $lkm = 1;
3  $kerrat = 10;
4  do
5  {
6      if ($lkm == 5)
7      {
8          break;
9      }
10     echo "Terve $lkm kerran<br>

```

```
11 | ";  
12 |     $lkm++;  
13 | } while ($lkm <= $kerrat);  
14 | ?>
```

Jätetty tarkoituksella tyhjäksi

© #AriRantala