

HUOM.

- Kyselyn mukaan 50/113 ei ole tehnyt PHP:tä
- Tätäkin oppii vain tekemällä **itse**
- Esimerkkejä syntaksin sijaan!

PHP historiaa

- 1995 “Personal Home Page Tools”
 - Kokoelma kaikenlaista metodia ja yritelmiä
 - Uudelleenkirjoitettu myöhemmin
- Voidaan upottaa HTML:n sekaan
- Suosituin palvelinohjelmointikieli
 - Wikipedia, Facebook, IRC-Galleria...
- LAMP-alusta
 - **L**inux käyttöjärjestelmän päällä
 - **A**pache WWW-palvelin
 - **M**ySQL –tietokanta
 - **P**HP tuki

Ongelmia?

- Laitoksella PHP:n ajaminen on erityisen tuskallista
 - I harjoituksissa tehtiin toivottavasti kaikille kätevä ”wrapperi”
 - Myöhemmin kävi ilmi, että PHP:stä on eri versioita asennettuna...
 - Tämä tuskailu ei varsinaisesti ole PHP:n vika...
- <http://blogs.cs.helsinki.fi/itmanager/2008/05/08/voi-phpta-voi-eihan-se-pida-vetta-mutta-kyllahan-se-soi/>

Syntaksi

- C/Java, kuten JavaScript
 - Mutta välillä ihan omanlaisensa
- Case sensitive
- Lauseen erottimena puolipiste
- Tyypittämätön ja tyyppi voi vaihtua
- Kommentit kuten JavaScriptissä
- Koodi joka on `<?php` ja `?>` välissä suoritetaan, muu voi olla vaikka HTML:ää

Muuttujat

- Muuttujan nimeä edeltää aina \$
 - \$song = "Waterloo";
- echo "Next song is \$song";
 - "String substitution" jos ""-llä määritelty string
- Katenointi on piste .
 - echo "Next song is ".\$song;
 - \$song .= " and SOS"
- Vakiot
 - define(VAKIO, "arvo");

Muuttujat ja näkyvyys

- Ei ihan niin kummallinen kuin JavaScriptissä, mutta silti:
 - Paikallisia
 - Staattisia paikallisia
 - Globaaleja
 - Superglobaaleja
- `<?php $abba = "Waterloo"; ?>`
 `<vaikka html-koodia>`
 `<?php echo($abba); ?>`

Tietotyypit ja totuusarvot

- Kokonaisluvut, liukuluvut, merkkijonot, totuusarvot
- Totuusarvot
 - Epätotta ovat: false, "", "0" ja null
 - Vertailussa merkkijono ensin numeroksi
 - ("hei" > 10) → (0 > 10)
- Identtisyys (==) ja yhtäsuuruus (===)

```
$a = "false";
```

```
$b = true;
```

```
( $a == $b) mutta ei ($a === $b)
```

Funktiot, oliot

- Funktion (melkein) kuin JavaScriptissä
- ```
function tervehdi($nimi) {
 echo "Hei $nimi";
}
```
- PHP:ssä on ”oikeammat” oliot kuin JavaScriptissä.
  - Kuitenkaan luennoilla ei mennä oloihin, funktiot riittävät.
  - Niille joita kiinnostaa:
    - [http://wiki.mureakuha.com/wiki/PHP-opas#Oliot\\_ja\\_luokat](http://wiki.mureakuha.com/wiki/PHP-opas#Oliot_ja_luokat)
  - Ja harjoituksissa saa kysyä!

# Include ja require

- Vähän kuten SSI:n includet, voidaan sisällyttää mikä tahansa tiedosto mutta (**Huom**)
- `include("tiedosto.php");`
  - Ei keskeytä suoritusta jos tiedostoa ei löydy
- `require("tiedosto.php");`
  - Keskeyttää suorituksen jos ei löydy
- **Huom:** Jos tiedostossa on esimerkiksi
  - `<?xml ...` alkuinen rivi, niin seuraukset ovat mielenkiintoisia.
    - Korjaus: lue tiedosto muuttujaan ja tulosta echolla!
- Oho, includella voi hakea myös <http://:n> yli (!)

# Taulukot

- Taulukkojen käsittely tärkeää, sillä PHP tarjoaa omia taulukoita CGI-ympäristömuuttujat ja GET/POST parametrit käsittelyyn
  - `$_SERVER`, `$_REQUEST`, `$_POST`, `$_GET`...
  - `$_GET["avain"]`
- Taulukot määritellään `array()` :llä
  - `$taulukko = array(10,02);`
  - `$assotaulukko = array(1=>10, "aaa" => "bee");`
  - Taulukkoja voidaan yhdistää: `$t1 + $t2;`
- `foreach()` sopii kätevästi taulukon läpikäyntiin:
  - `foreach ($taulukko as $arvo) { echo $arvo }`
  - `foreach ($taulukko as $avain => $arvo) { echo "$avain: $arvo" }`

# Valmis DIME-BACKEND

- Harjoitustyöhön PHP:tä ei välttämättä tarvitse tehdä, kunhan muut osa-alueet on tehty ”riittäväällä laajuudella”
- Huomenna esitellään valmis komponentti, jota voi käyttää toteuttamatta itse tietokantajuttuja

# Esimerkkejä

- Lomakkeen käsittely
- Tiedoston lähetys
- Sessiot

# Super Dangerous PhAP

```
1 <?php
2
3 // SUPER DANGEROUS PhAP FRAMEWORK 12/2008
4 // NOT FOR PRODUCTION USE
5
6 // Public Domain by Matti Paksula matti.paksula@iki.fi
7
8 $action=$_GET['action'];
9 $controller_name=$_GET['controller'];
10
11 $controller_code = file_get_contents("controllers/".$controller_name.".php");
12 $controller = ucfirst($controller_name);
13
14 eval($controller_code);
15 eval("$controller::$action();");
16
17 include("views/".$controller_name."/".$action.".html");
18
19 ?>
```