

# Virheenetsintä: PHP

## Syntaksivirheet

Helpoiten huomattavat virheet PHP-kielessä ovat ns. syntaksivirheitä. Kirjoitat jotain, jota PHP:n ”kielioppi” ei hyväksy, ja selain heittää sinulle virheilmoituksen. Syntaksivirheiden virheilmoitukset alkavat ”parse error” tai ”syntax error”.

Mikäli näkyvissä ei ole PHP-osioita eikä virheilmoituksia, virheraportointi on ehkä poissa päältä. Sen pitäisi kyllä olla oletuksena päällä XAMPP-palvelimessa.

Lisää tässä tapauksessa PHP-tiedostosi alkuun seuraavat rivit:

```
<?php
    ini_set("display_errors",1);
    error_reporting(E_ALL);
?>
```

## Missä virhe on?

Virheilmoitus antaa rivinumeron, mutta virhe voi olla myös edellisellä rivillä. Erityisesti jos se on puuttuva puolipiste rivin lopussa tai puuttuva aaltosulku lohkon lopussa.

## Tavallisia virheitä

- Jokaisella avaavalla sululla tulee olla sulkeva sulku – ja oikeassa paikassa. Useimmissa editoreissa voit napsauttaa avaavaa sulkua, ja se näyttää sinulle sulkevan sulun. If- ja while-lauseiden ehtoihin tulee helposti väärä määrä.
- Käytettäessä lainausmerkkejä tulee myös herkästi virheitä. Niitä on yksi liikaa tai yksi liian vähän jossakin. Koodi on kopioitu ja liitetty jostakin, ja ”lainausmerkit” eivät olekaan lainausmerkkejä (esimerkiksi Word- tai PowerPoint-tiedostosta kopioidessa tulee vinot lainausmerkit, joita ei yleensä tunnisteta). Useimmat editorit auttavat värikoodaamalla lainausmerkeissä olevat merkkijonot eri tavalla kuin koodin.
- Jos muuttujan nimestä puuttuu alusta dollarimerkki, se tulkitaan merkkijonoksi ja lopputulos saattaa olla syntaksivirhe.
- Sulkeva PHP-tägi puuttuu, jolloin selain luulee HTML-koodiasi PHP:ksi.
- ”Warning: Undefined variable” tarkoittaa, että käytät muuttujaa, jolle et ole antanut arvoa. Virheilmoituksessa on muuttujan nimi ja millä rivillä sitä käytetään. Tämä on varoitus, joten se ei kaada koko sivua, mutta tällaisetkin virheet kannattaa korjata.
- ”Warning: Undefined array key” tai ”undefined index” liittyy aina taulukoihin. Erityisesti \$\_POST- ja \$\_GET-tilukkoissa tarkista, että vastaavan niminen kenttä on lähetetty sivulle. Tämä esiintyy myös, jos lomaketta ei ole lähetetty lainkaan (tarkista aina isset-funktiolla).
  - \$\_SESSION-tilukossa syy saattaa olla myös, että sessiota ei kyseisellä sivulla ole käynnistetty session\_start-metodilla.
- ”Call to undefined function” tarkoittaa, että yrität käyttää funktiota jota ei ole olemassa. Tarkista funktion määritelmän ja funktiokutsun kirjoitusasu. Tämä on todennäköisesti kirjoitusvirhe.

- "Fatal error: Call to a member function on a non-object" tai "Notice: Trying to get property of non-object". Tämä virheilmoitus saattaa tulla vastaan tehtävissä, joissa käytetään tietokantaa. Muissa tehtävissä ei käytetä olioita. Todennäköisesti SQL-lauseessasi on virhe, ja sen ajaminen palauttaa NULL-olion.

### SQL-virheiden paikantaminen

SQL-virheiden paikantamiseen on kaksi tapaa, joita yleensä käytetään enemmän tai vähemmän yhdessä.

Ensinnäkin voit tulostaa SQL:n virheilmoituksen. Jos tietokantayhteyden sisältävä muuttujasi on nimeltään \$yhteys, se tapahtuu käskyllä:

```
<?php echo $yhteys->error; ?>
```

Yleensä virheilmoitus on, että SQL-lauseessasi on syntaksivirhe. Jos virheilmoitus ei näytä SQL-lausetta, tulosta se itse echo-käskyllä. Esim.

```
<?php
$rek = "ABC-123";
$sql = "select * from auto where rekisterinro = ". $rek;
echo $sql; // tulostaa "select * from auto where rekisterinro = ABC-123
// eli hipsut unohtuivat merkkijonon ympäriltä
?>
```

### Logiikkavirheet

Syntaksivirheitä hankalampia paikannettavia ovat logiikkavirheet. Koodisi kielioppi on kyllä oikein, mutta sen suoritus ei etene haluamallasi tavalla. Tähänkin on muutamia hyviä temppuja, joihin ei tarvita debuggeria.

### Meneekö suoritus tänne?

Jos jokin ei näytä tapahtuvan, selvitä ensin, eteneekö suoritus kyseiselle riville. Tämä tehdään sijoittamalla tulostuskäskyjä sopiviin paikkoihin. Esim.

```
<?php
function testiFunktio($param) {
    echo "Testi: Suoritus eteni funktioon";
    // funktion koodia
    if($param == "Uusi") {
        echo "Testi: Parametri 'uusi'.";
        // funktion koodia
    }
}

testiFunktio("uusi");
?>
```

Sillä ei ole väliä, mitä tulostukset ovat. Tärkeintä on nähdä, meneekö suoritus oikeaan paikkaan. Tässä tapauksessa, jos ensimmäinen tulostus tapahtuisi, mutta toinen ei, if-lauseen ehto olisi väärin (kuten se

esimerkissä onkin). Testitulostukset poistetaan sitä mukaa, kun on varmistettu, mihin asti koodin suoritus pääsee.

Jos tulostuksia ei näy sivulla, johon on tehty HTML/CSS-asemointia, tarkista myös sivun lähdekoodi. Se saattaa olla piilossa jonkin toisen elementin alla.

### **Tulosta muuttujan arvo**

Jos jotain pitäisi tapahtua, mutta ei tapahdu, ja koodin logiikka näyttää oikealta, seuraava askel on selvittää, minkä muuttujan arvo mättää. Tämä tehdään yksinkertaisesti tulostamalla muuttujan arvo. Yksittäiset muuttujat voi tulostaa echo-käskyllä, taulukoihin käytetään print\_r-funktiota. Esim.

```
<?php
    echo $muuttuja;
    print_r($_POST);
?>
```

### **Tehtävä**

Tallenna tämä dokumentti itsellesi johonkin helposti löydettävään paikkaan. Siitä on paljon hyötyä buginmetsästyksessä, kun koodaat PHP-kielellä. Palauta dokumentti myös palautuskansioon todisteena, että olet tutustunut siihen.