VENTSPILS DIGITĀLAIS CENTRS

MAINĪGO APGABALS

PHP

M.Kalneja

Mainīgo apgabals (variable scope) ir PHP koda apgabals, kurā attiecīgais mainīgais ir pieejams. PHP valodā ir divi mainīgo apgabali. Globālie mainīgie ir pieejami tieši PHP koda izpildes laikā. Tas ir, jūs varat tiem piekļūt tik ilgi, kamēr neesat kādā funkcijas izsaukumā vai objektorientētas klases metodē.

Atšķirībā no citām programmēšanas valodām, PHP funkcijām ir pašām savs mainīgo apgabals.
Piemēram:

```
<?php
function test( ) {
    echo $a;
}

$a = "Hello World";
test( );
?>
```

Ja paskatīsimies skriptu darbībā, redzēsim, ka nekas netiek izvadīts (ja neskaita kļūdu paziņojumus). Tas ir tāpēc, ka mainīgais $a, kuru mēģinam izmantot test() funkcijā ir pilnīgi atšķirīgs no globālā $a, kuru esam izveidojis tieši pirms funkcijas izsaukšanas.

Lai jūs varētu funkcijas iekšienē piekļūt globālā apgabala mainīgajiem, jums ir par to jāpaziņo funkcijai katram atsevišķam mainīgajam. Paziņošana notiek ar global atslēgas vārdu.

```php
<?php
function test( ) {
    global $a;
    echo $a;
}

$a = "Hello World";
test( );
?>
```

Ievērojiet, ka nav desmit reižu jāraksta atslēgas vārds global.

Tam var izmantot mainīgo atdalīšanu ar komatu:

```
<?php
function test( ) {
    global $a, $b, $c;
    echo $a;
    echo $b;
    echo $c;
}

$a = "Hello World!";
$b = "How are you?";
$c = "Bye!";
test( );
?>
```

Tieši tāpat jūs varat izmantot arī PHP iebūvēto globālo mainīgo masīvu.
Piemērs:

```
<?php
function test( ) {
    echo $GLOBALS['a'];
}

$a = "Hello World";
test( );
?>
```

Pēdējā piemērā $GLOBALS masīvs ir superglobāls, kas nozīmē to, ka tas ir pieejams visos mainīgo apgabalos, bez īpašas vajadzības to definēt kā īpaši, lai varētu izmantot funkcijas iekšienē.

VENTSPILS DIGITĀLAIS CENTRS

STATISKIE MAINĪGIE

M.Kalneja

PHP atbalsta arī funkciju mainīgo definēšanu kā statiskus mainīgos. Statisks mainīgais saglabā savu vērtību starp funkciju izsaukumiem, taču ir joprojām sasniedzams attiecīgās funkcijas iekšienē.

Statiskajiem mainīgajiem var piešķirt arī sākuma vērtību. Šī sākuma vērtības piešķiršana notiek tikai pašu pirmo reizi, kad static deklarācija tiek izpildīta. Statiskie mainīgie bieži vien tiek lietoti kā skaitītāji.

```
function hitcount( ) {
    static $count = 0;

    if ($count == 0) {
        echo "This is the first access to this page";
    } else {
        echo "This page has been accessed $count times";
    }
    $count++;
}
```

M.Kalneja

VENTSPILS DIGITĀLAIS CENTRS

AR WEB SAISTĪTI MAINĪGIE

PHP automātiski izveido mainīgos visiem datiem, kas tiek saņemti HTTP pieprasījumā. Tie var būt GET, POST, cookie un vides (environment) mainīgie. Šie mainīgie tiek ievietoti vai nu PHP globālajā simbolu tabulā vai arī vienā no varākiem superglobālajiem masīviem, atkarīgā no register_globals uzstādījuma php.ini failā.

Sākot ar PHP versiju 4.2.0, register_globals noklusētā vērtība ir off. Ar šādu vērtību visi dažādie mainīgie, kas iepriekš bija pieejami globālajā simbolu tabulā, tagad tiek sadalīti pa superglobālajiem masīviem. Ir noteikta kopa ar superglobālajiem masīviem, tie nevar tikt izveidoti ar parasta PHP skripta palīdzību. Lietojamais masīvs ir atkarīgs no mainīgā avota.

$_GET

GET metodes mainīgie. Šie mainīgie parasti tiek norādīti pieprasījuma saitē (request *URL*). Piemēram, saite: http://www.example.com/script.php?a=1&b=2, tiks uzstādīti masīva $_GET elementi $_GET['a'] un $_GET['b'] ar attiecīgajām vērtībām 1 un 2

$_POST

POST metodes mainīgie. Parasti ar šo metodi dati tiek iegūti no parastām HTML formām ar post metodi datu nosūtīšanai.

$_COOKIE

Visi cookie, ko pārlūkprogramma atsūta līdz ar pieprasījumu. Cookie nosaukums pārtop par masīva indeksu, bet vērtība - par attiecīgajam indeksam piesaistīto vērtību.

$_REQUEST

Šis masīvs satur visas iepriekšējo trīs masīvu vērtības (GET, POST un cookie). Ja mainīgais parādās vairākos no šiem masīviem, kārtība, kādā tie tiek importēti $_REQUEST masīvā, tiek uzrādīta variables_order parametrā php.ini failā. Noklusētā vērtība šim parametram ir „GPC“, kas nozīmē to, ka vispirms tiek ielasītas vērtības no GET metodes, tad no POST metodes un visbeidzot cookie vērtības.

$_SERVER

VENTSPILS DIGITĀLAIS CENTRS

Šos mainīgos uzstāda jūsu web serveris. Parasti tiek uzstādītas tādas lietas kā DOCUMENT_ROOT, REMOTE_ADDR, REMOTE_PORT, SERVER_NAME, SERVER_PORT un daudzas citas.

Lai iegūtu pilnu sarakstu, apskatieties uz savu phpinfo()

izvada rezultātu, vai arī iedarbiniet šādu skriptu:

```
<?php
foreach($_SERVER as $key=>$val) {
    echo '$_SERVER['.$key."] = $val\n";
}
?>
```

VENTSPILS DIGITĀLAIS CENTRS

$_ENV

Visi mainīgie, kas attiecas uz vidi (environment), kuri ir uzstādīti startējot jūsu web serveri, būs pieejami šajā masīvā.

$_FILES

Failu augšupielādei pēc RFC 1867 katram atsevišķajam failam atbilstošā informācija būs pieejama šajā masīvā. Piemēram, faila augšupielādes forma, kas satur:

<input name="userfile" type="file">

Pieņemsim, ka jūs augšupielādējat kādu savu bildi,

$_FILES masīvs izskatīsies šādi:

$_FILES['userfile']['name'] => photo.png

$_FILES['userfile']['type'] => image/png

$_FILES['userfile']['tmp_name'] => /tmp/phpo3kdGt

$_FILES['userfile']['error'] => 0

$_FILES['userfile']['size'] => 158918

Ievērojiet, ka „error“ ir jauns lauks PHP versijām, kas jaunākas par 4.2.0 un tā vērtības ir:

- 0 - kļūdu nav, fails tika augšupielādēts
- 1 - augšupielādētais fails pārsniedz upload_max_filesize parametra vērtību php.ini failā
- 2 - augšupielādētā faila izmērs pārsniedz MAX_FILE_SIZE parametru, kas norādīts HTML formā
- 3 - baitu skaits, kas ir mazāks par norādīto augšupielādējamā faila izmēru
- 4 - fails netika augšupielādēts

VENTSPILS DIGITĀLAIS CENTRS

Jautājumi ????