PHP III

Ivari Horm

ranger@deepdust.com

Sissejuhatus

- **Funktsioonid**
- **Muutujate skoop**
- **Viited muutujatele**

Funktsioonid

Ivari Horm

ranger@deepdust.com

Funktsioonid

- Võimaldavad mingit kindlat ülesannet täitvad PHP koodi osad grupeerida üheks tervikuks
- Funktsioonide abil saab ühes ja samas failis olevat koodi erineval ajal ja korduvalt välja kutsuda (käivitada)

Funktsioonid

- Funktsioonid tuleb alati ära kirjeldada
- Kirjeldatud funktsioone on võimalik välja kutsuda

Kirjeldamine

```
function fun1($arg1, $arg2, …) {
    statement1;
    statement2;
    …
}
```

- **fun1** on funktsiooni nimi. Ühes käivitatavas terviklikus PHP koodis peavad funktsioonide nimed olema unikaalsed.
- Funktsiooni sees olevad PHP käsud **statement1, statement2** peavad olema alati ümbritsetud loogeliste sulgudega
- Muutujaid **arg1, arg2** nimetatakse funktsiooni **fun1** argumentideks
- Argumendid on muutujad, mis luuakse funktsiooni käigus ja mis saavad endale funktsiooni välja kutsel antud väärtused
- Argumendid kirjutatakse alati sulgudesse funktsiooni nime järele
- Kui funktsioonil argumendid puuduvad, kirjutatakse vaid tühjad sulud!

Kirjeldamine

Funktsiooni kirjeldamine

```
function pythagoras($a,$b) {
        print sqrt($a*$a+$b*$b);
}
```

Väljakutse

fun1($arg1, $arg2, …);

- **fun1** on funktsiooni nimi, mida soovitakse käivitada
- Muutujaid **arg1, arg2** nimetatakse funktsiooni **fun1** argumentideks
- Funktsiooni argumentide kohale võib kirjutada nii konkreetseid väärtusi kui ka teisi muutujaid
- Kui funktsioonile ei ole vaja argumente anda, tuleb ikkagi kirjutada tühjad sulud!
- **Funktsiooni väljakutsel ei kirjutata sõna “function”!**

Väljakutse

Funktsiooni väljakutse

```
function pythagoras($a,$b) {
        print sqrt($a*$a+$b*$b);
}

pythagoras(5,2);

$x = 3;
$y = 8;

pythagoras($x,$y);
```

Argumentide arv

- Argumentide arv määratakse funktsiooni kirjeldamisel
- Funktsiooni välja kutsudes peab väljakutses olevate argumentide arv olema võrdne funktsiooni kirjelduses antud argumentide arvuga
- Väljakutsel peab olema argumentide järjekord sama, mis funktsiooni kirjelduseski

Funktsiooni argumendid

Funktsiooni argumendid

```
function makedate($day,$month,$year) {
        print $year-$month-$day;
}

makedate('20','06','2004');

makedate('20','','2004');

//Viga:
makedate();
```

Väärtuse tagastamine

```
return $val;
```

- Käsuni jõudes katkestatakse funktsiooni töö ja väljutakse sellest
- Väljumisel tagastatakse funktsiooni välja kutsunud skripti väärtus **val**
- Mida väärtusega peale hakata on programmeerija probleem

Väärtuse tagastamine

Funktsiooni väärtuse tagastamine

```
function pythagoras($a,$b) {
	return sqrt($a*$a+$b*$b);
}

//Välja arvutatud tulemus läheb kaduma
pythagoras(5,2);

//Muutuja a saab väärtuseks √29
$a = pythagoras(5,2);

//Trükitakse välja √29 väärtus
print pythagoras(5,2);
```

Standardfunktsioonid

- PHP arendajate poolt loodud funktsioonide komplekt
- *print(), mail(), array(), sqrt()*

Standardfunktsioonid

- Paljud standardfunktsioonid nõuavad oma tööks PHP vastava mooduli lubamist
- **mysql_connect()** nõuab PHP-s MySQL-i toe olemasolu
- **ldap_get_entries()** nõuab PHP-s LDAP-i tuge

Muutujate skoop

Ivari Horm

ranger@deepdust.com

Muutujate skoop

- Muutujate kirjeldamise asukoht määrab nende muutujate skoobi
- Skoop näitab piirkonda PHP koodis, kus antud muutujate väärtusi kasutada saab
- Väljaspool skoopi antud muutujaid ei eksisteeri

Muutujate skoop

- Funktsioonides kirjeldatud muutujad hävivad funktsioonist väljumisel
- Skripti töö lõppedes hävivad kõik skriptis kasutusele võetud muutujad

Muutujate skoop

Muutujate skoop

```
$a = 2;
print $a;                                   // 2

function my_fun(){
        $a = 5;
        print $a;                           // 5
}

my_fun();

print $a;                                   // 2
```

Muutujate skoop

Muutujate skoop

```
$i=2;
print $i;                                   // 2

for ($i=0;$i<4;$i++) {
        $k = “Hello World”;
        print $k;                           // ”Hello World”
        print $i;                           // 0 ... 3
}

print $k;                                   // ”Hello World”

print $i;                                   // 4
```

Globaalmuutujad

- Väljaspool funktsioone kirjeldatud muutujad säilivad kuni programmi töö lõpuni
- Neid nimetatakse globaalmuutujateks
- Globaalmuutujad on ligipääsetavad **GLOBALS[]** massiivi kaudu

Globaalmuutujad

Globaalmuutujad

```
$a = 2;
print $a;                                   // 2

function my_fun($a){
        print $a;                           // 5
        print $GLOBALS[“a”];                // 2
        return $a;
}

my_fun(5);

print my_fun(5);                            // 5

print $a;                                   // 2
```

Include-käsk

- Kõik muutujad, mis on failis defineeritud enne INCLUDE-käsku, on nähtavad ka nimetatud käsuga lisatud failis

Include-käsk

Muutujate skoop ja INCLUDE

```
$a = 2;
print $a;                                   // 2

include “sub.inc.php”;

print $a;                                   // 3
```

sub.inc.php

```
print $a;                                   // 2

$a++;

print $a;                                   // 3
```

Soovitused

- Kasutage nii vähe globaalmuutujaid kui vähegi võimalik
- Kasutage nii globaalmuutujate kirjeldamisel kui nende väärtuste lugemisel massiivi **GLOBALS[]**

Viited

Ivari Horm

ranger@deepdust.com

Viited

- Võimaldavad muutuja sisu poole pöörduda mitme erineva nime abil

Kasutamine

```
$b = &$a;
```

- Muutujale **a** on omistatud mingi kindel väärtus
- Antud käsuga hakkab muutuja **b** viitama samale väärtusele, mis **a**
- Muutuja **a** kustutamisel väärtus säilib ning on ligipääsetav muutuja **b** kaudu
- Väärtus kustutatakse mälust ainult juhul, kui kaotatakse nii muutuja **a** kui ka **b**

Kasutamine

Viidete kasutamine

```
$x = 7;
$y = &$x;

print $y;                                   // 7

$y = 10;
print $x;                                   // 10

unset($x);
print $y;                                   // 10

unset ($y);
print $y;                                   // Nothing!
```