

# PHP II

Ivari Horm

ranger@risk.ee

- **Failid**
- **Massiivid**
- **Eriotstarbelised massiivid**
- **Abifunktsioonid**
- **E-kirjade saatmine PHP-s**

# Failid

Ivari Horm  
ranger@risk.ee

- Arvutis olevaid faile on võimalik avada ja nendesse salvestatud andmeid PHP koodides kasutada
- Failis oleva info töötlemine toimub üldjuhul rida haaval

- Faili avamine *(open)*
- Failis oleva info lugemine *(put)*
- Faili kirjutamine *(get)*
- Faili sulgemine *(close)*

```
$fp = fopen("fn", "type");
```

- Avab faili **fn** andmete lugemiseks või kirjutamiseks
- **fp** on viit, mis näitab failis reale, kust toimub lugemine või kuhu hakatakse kirjutama
- **fn** on failinimi koos laiendiga, mille all ta kõvakettale salvestatud on
- **type** määrab ära avamismeetodi:

Tähis	Fail avatakse...	Viit asub...	Fail...
<b>r</b>	ainult lugemiseks	alguses	
<b>r+</b>	lugemiseks ja kirjutamiseks	alguses	
<b>w</b>	kirjutamiseks	alguses	tühjendatakse/luuakse
<b>w+</b>	lugemiseks ja kirjutamiseks	alguses	tühjendatakse/luuakse
<b>a</b>	kirjutamiseks	lõpus	luuakse
<b>a+</b>	kirjutamiseks	lõpus	luuakse

```
$row = fgets($fp, len);
```

- Loeb failist ühe rea ja salvestab selle muutujasse
  - **row** on muutuja, kuhu salvestatakse failist loetud rida
  - **fp** on viit failile, mis on avatud käsuga fopen()
  - **len** on baitide arv, mis failist loetakse
- 
- Lugemine toimub senikaua, kuni jõutakse väärtuseni **len**, saabub rea lõpp või faili lõpp

- Failist lugemine tuleb lõpetada, kui jõutakse faili lõppu
- Kui failiviit **fp** asub juba faili lõpus, siis kõik järgnevad lugemised annavad veateate
- Kui failiga töötamine on lõppenud, tuleb fail sulgeda



## **feof (\$fp);**

- Kontrollib, kas failiviit **fp** asub juba faili lõpus või mitte
- Kui viit asub lõpus, väljastatakse TRUE, vastasel korral aga FALSE
- **fp** on viit failile, mis avati käsuga fopen()

## **fclose (\$fp);**

- Sulgeb faili, mille viit oli **fp** ja vabastab selle muutuja
- Kui PHP skript lõpetab töö, suletakse automaatselt kõik avatud failid

## Tekstifaili kasutamise näide

```
//Avame faili data.txt ainult lugemiseks
$fp = fopen("data.txt","r");
if (!$fp) {
    print "Faili avamine ebaõnnestus";
    return;
}

//Senikaua kui ei ole jõutud faili lõppu...
while (!feof($fp)) {
    //Salvesta üks rida failist muutujasse
    $r = fgets($fp);

    //Trüki muutuja sisu välja
    print "$r <br />";
}
fclose($fp);
```

```
fputs($fp, $row, len);
```

- Salvestab muutujas oleva info faili
- **fp** on viit failile, mis avati käsuga fopen()
- **row** sisaldab andmeid, mida on vaja faili salvestada
- **len** määrab, kui mitu baiti muutujast **row** faili kopeeritakse
  
- Fail peab olema kindlasti avatud lugemiseks (vt. fopen-käsu võtmeid)

## Tekstifaili salvestamise näide

```
//Avame faili data.txt ainult kirjutamiseks
$fp = fopen("data.txt","a");
if (!$fp) {
    print "Faili avamine ebaõnnestus";
    return;
}

$r = "See rida tuleb salvestada tekstifaili\n";

fputs($fp,$r);

fclose($fp);
```

# Massiivid

Ivari Horm

ranger@risk.ee

- Nädalapäeva tähistab number 0...6
- Sellist vastavust saab esitada tabelina
- Nädalapäeva numbri saamiseks on PHP-s käsk `date()`

	1	2
	Nr	Päev
1	0	Pühapäev
2	1	Esmaspäev
3	2	Teisipäev
4	3	Kolmapäev
5	4	Neljapäev
6	5	Reede
7	6	Laupäev

## Nädalapäeva numbri saamine

```
//Salvestame päevale vastava numbri muutujasse p
$p = date("w");

print $p;
```

- <http://lib.deepdust.com/php>
- **Ülesanne 1**  
Trükkida välja kuupäev kujul pp.kk.aaaa
- **Ülesanne 2**  
Trükkida välja kuupäev kujul aaaa-kk-pp tt:mm:ss



- PHP-s on vaja kasutada nädalapäevade numbrite asemel nende nimetusi
- Nimetused on vaja salvestada muutujatesse

### Nädalapäevade nimetused

```
$e="Esmaspäev" ;  
$t="Teisipäev" ;  
$k="Kolmapäev" ;  
$n="Neljapäev" ;  
$r="Reede" ;  
$l="Laupäev" ;  
$p="Pühapäev" ;
```

- Ebapraktiline
- Kui päeva tähistav number on 2, siis kuidas kuvada selle numbri asemel sõna “Teisipäev”?
- Palju muutujaid tekitab arvuti mälus kaose

- Nädalapäevade nimekiri on oma olemuselt maatriks
- Maatriksitega töötamiseks on PHP-s ja ka teistes keeltes olemas muutujad, mille tüüp on massiiv (*Array*)

- Massiivid **luuakse** (tehakse muutuja)
- Massiivid koosnevad **elementidest e. liikmetest**
- Massiivid **väärtustatakse** (igasse massiivi elementi salvestatakse mingi väärtus: tekst või number)

- Salvestame kõik nädalapäevad ühte muutujasse

## Nädalapäevade nimetuste massiivi väärtustamine

```
$np[0] = "Pühapäev" ;  
$np[1] = "Esmaspäev" ;  
$np[2] = "Teisipäev" ;  
$np[3] = "Kolmapäev" ;  
$np[4] = "Neljapäev" ;  
$np[5] = "Reede" ;  
$np[6] = "Laupäev" ;
```

- Massiivi elemendi lugemisel on vaja ära näidata nii massiivi nimi kui ka selle elemendi asukoht massiivis

## Nädalapäevade lugemine massiivist

```
//Trükime välja sõna "Esmaspäev"  
print $np[1];
```

```
//Trükime välja sõna "Reede"  
print $np[5];
```

# Massiivi väärtuste väljastamine

- Nädalapäeva numbri väljastas PHP-s `date()` käsk
- Selle muutuja abil on võimalik ekraanile trükkida nädalapäeva nimetuste massiivist õige element
- Õige element on antud päeva numbrile vastav nimetus

## Tänase nädalapäeva nime trükkimine

```
//Hangime päeva numbri
$p = date("w");

//Trükime välja nime
print $np[$p];
```

## Nädalapäeva nimede massiiv

```
$np[0] = "Pühapäev";
$np[1] = "Esmaspäev";
$np[2] = "Teisipäev";
$np[3] = "Kolmapäev";
$np[4] = "Neljapäev";
$np[5] = "Reede";
$np[6] = "Laupäev";
```



- Mitmemõõtmeliste massiivide korral on vaja ära näidata täpne elemendi asukoht massiivis

15	16	17	18
29	30	31	32

## Mitmemõõtmeline massiiv

```
$np[1][1]=15;  
$np[1][2]=16;  
$np[1][3]=17;  
$np[1][4]=18;  
$np[2][1]=29;  
$np[2][2]=30;  
$np[2][3]=31;  
$np[2][4]=32;
```

“Esimeses reas esimeses veerus asub number 15...”

- Igal hüperlingil on alati olemas lingi nimetus ja aadress, kuhu ta viitab
- Lingi nimetusest ja aadressist saab moodustada maatriksi
- Selles maatriksis on alati kaks veergu

- Iga massiivi rida on üks link
- Esimene element reas on nimi
- Teine on kohaviit

	1	2
	Nimi	Viide
1	Neti	www.neti.ee
2	Delfi	www.delfi.ee
3	Altavista	www.av.com

### Lingi nimetuste ja aadresside massiiv

```
$ln[1][1]="Neti";  
$ln[1][2]="www.neti.ee";  
  
$ln[2][1]="Delfi";  
$ln[2][2]="www.delfi.ee";  
  
$ln[3][1]="Altavista";  
$ln[3][2]="www.av.com";
```

- Kood on pikk (s.t. käsud on pikad)
- Tülikas kirjutada
- Raske jälgida, tekib palju vigu

- Mõistlik oleks veergu ja rida tähistavad numbrid asendada nime endaga
- Sel juhul pole tähtis, mitmendas reas ja veerus mingi element asub

	1	2
	Nimi	Viide
1	Neti	www.neti.ee
2	Delfi	www.delfi.ee
3	Altavista	www.av.com

## Lingi nimetuste ja aadresside massiiv

```
$ln["Neti"]="www.neti.ee";  
$ln["Delfi"]="www.delfi.ee";  
$ln["Altavista"]="www.av.com";
```

“Nimele “Neti” vastab link “www.neti.ee”...”

- Väheneb koodi pikkus
- Parem ülevaade, kuid mitte piisav
- Iga rea ette tuleb kirjutada muutuja nimi
- Vigade tegemise oht on endiselt suur

```
$arr = array(  
    elem1 => val1,  
    elem2 => val2, ... );
```

- Array() võimaldab PHP-s massiive luua ja neid väärtustada
- **arr** on massiiv, mis tekib antud käsu täitmisel
- Vasakul pool noolt asuvad elementide nimetused
- Paremal pool noolt on nendele elementidele vastavad väärtused
- Ühte paari **element => väärtus** nimetatakse **vastavuseks**
- Vastavused eraldatakse üksteisest komaga!

## Lingi nimetuste ja aadresside massiiv

```
$ln["Neti"]="www.neti.ee";  
$ln["Delfi"]="www.delfi.ee";  
$ln["Altavista"]="www.av.com";
```

- Asendame ülaltoodud massiivi array()-käsuga

## Array()-käsu kasutamine

```
$ln=array(  
    "Neti" => "www.neti.ee",  
    "Delfi" => "www.delfi.ee",  
    "Altavista" => "www.altavista.com");
```



- Massiivid, mis kujutavad endast vastavusi (element => väärtus), nimetatakse **assotsiatiivseteks** massiivideks (*associative array*)
- Vastavusteta massiivid on **mitteassotsiatiivsed** massiivid (*unassociative array*)

- Assotsiatiivse massiivi puhul tuleb konkreetse elemendi poole pöördumisel selle elemendi nimi **alati** kirjutada jutumärkide vahele

## Assotsiatiivse massiivi elemendid

```
$ln[ "Neti" ]="www.neti.ee" ;  
$ln[ "Delfi" ]="www.delfi.ee" ;  
$ln[ "Altavista" ]="www.av.com" ;
```

- Kuidas kõige lihtsamini elemente massiivist kätte saada?

```
foreach($arr as $key => $val)  
    lihtlause;
```

- Tsükkel, mis teostab iga massiivi elemendiga mingi operatsiooni
- **arr** on massiiv, millest on vaja väärtusi välja lugeda
- Tsükli esimesel täitmisel salvestatakse muutujasse **key** massiivi esimese elemendi nimetus ja muutujasse **val** selle elemendi väärtus
- **lihtlause** täitmisel saab kasutada äsja väärtustatud muutujaid **key** ja **val**
- Pärast **lihtlause** täitmist salvestatakse muutujatesse **key** ja **val** järgmise massiivi elemendi nimetus ja väärtus
- Protseduuri korratakse senikaua, kuni jõutakse massiivi lõppu

- Iga massiivi salvestatud viite kohta kuvatakse üks link
- Enne lingi loomist salvestatakse lingi nimi muutujasse **name** ja aadress muutujasse **ref**

	1	2
	Nimi	Viide
1	Neti	www.neti.ee
2	Delfi	www.delfi.ee
3	Altavista	www.av.com

### Linkide kuvamine FOREACH-tsükli abil

```
foreach ($ln as $name => $ref)
{
    print "<a href=\""$ref\""$>$name</a>" ;
}
```

# Eriotstarbelised massiivid

Ivari Horm

ranger@risk.ee

- Mõningaid massiive loob PHP skripti töö käigus ise
- Üldjuhul hoitakse nendes kasutaja poolt skriptile ette antud andmeid ning veebiserveriga seotud infot

- **\_GET[]**  
Sisaldab URL-i ja vormide kaudu sisestatud muutujaid ja nende väärtusi
- **\_POST[]**  
Sisaldab vormide kaudu sisestatud muutujaid ja nende väärtusi, kui vormi käsus sisaldub atribuut *method="post"*
- **\_SERVER[]**  
Sisaldab veebiserveri kohta käivat infot



- Tekitame vormi, milles on kaks tekstivälja ning üks saatmisnupp. Tekstiväljadele salvestatud info antakse edasi skriptile **request.php** muutujate **j** ja **k** kaudu

## Vormi loomine, mis kasutab GET-meetodit

```
<form action="request.php">
  <input type="text" name="j" />
  <input type="text" name="k" />
  <input type="submit" name="sent" value="Saada" />
</form>
```

- Skripti käivitamisel satuvad muutujad **j** ja **k** massiivi **\_GET[]** elementideks ning nende väärtused vastavate elementide väärtusteks

## GET-massiivi kasutamine

```
print $_GET["j"];  
print $_GET["k"];  
print $_GET["sent"];
```

- Kui vormile on lisatud atribuut **method="post"**, ei ole muutujaid näha URL-is ning need satuvad PHP skripti käivitamisel **\_POST[]** massiivi elementideks

## Vormi loomine, mis kasutab POST-meetodit

```
<form action="skript.php" method="post">
  <input type="text" name="j" />
  <input type="text" name="k" />
  <input type="submit" name="sent" value="Saada" />
</form>
```

- **\_GET[]** massiiv on antud juhul tühi. Selles ei sisaldu ei j ega k!

## POST-massiivi kasutamine

```
print $_POST["j"];  
print $_POST["k"];  
print $_POST["sent"];
```

# Abifunktsioonid

Ivari Horm

ranger@risk.ee

- Eriotstarbeliste massiivide sisu ning muude serveri parameetrite kohta saab infot käsuga **phpinfo()**
- Paljudes serverites on turvalisuse huvides antud käsk keelatud

```
print_r($var_name);
```

- Trükitab välja muutuja sisu kasutajasõbralikul kujul
- **var\_name** on muutuja nimi, mille sisu soovitakse näha
- Tüüpiliselt kasutatakse antud käsku massiivide korral

## print\_r()-käsu kasutamine

```
$ln=array(
    "Neti" => "www.neti.ee",
    "Delfi" => "www.delfi.ee",
    "Altavista" => "www.altavista.com");

print "<pre>";

//Trükime välja massiivi ln struktuuri
print_r ($ln);

print "</pre>";
```



# E-mailide saatmine PHP-s

Ivari Horm

ranger@risk.ee

- PHP skriptidest on võimalik saata mistahes meiliaadressile e-kirju

```
mail("recipient", "subject", "content", "headers");
```

- Saadab PHP-skriptist e-kirja kindlaksmääratud aadressil
- **recipient** on kirja saaja aadress
- **subject** sisaldab e-kirja teemat
- **content** on kirja sisu
- **headers** võimaldab määrata täiendavaid päiseid, mida kirjale lisatakse:

Täiendav päis	Kirjeldus
From: mailiaadress	Kirja saaja näeb saatja aadressina skripti kirjutaja poolt määratud aadressi

## E-maili saatmine PHP skripti kaudu

```
$recipient = "user@deepdust.com";  
$subject = "Testikiri";  
$body = "See on testikiri";  
$headers = "From:ranger@deepdust.com";  
  
mail($recipient,$subject,$body,$headers);
```