

# Sissejuhatus PHP-sse

Ivari Horm

ranger@risk.ee

- **Sissejuhatus**
- **Andmetüübid ja muutujad**
- **Jutumärgid**
- **Operaatorid**
- **Sisend/väljund**
- **Kontrollstruktuurid**
- **Mitme faili kasutamine**

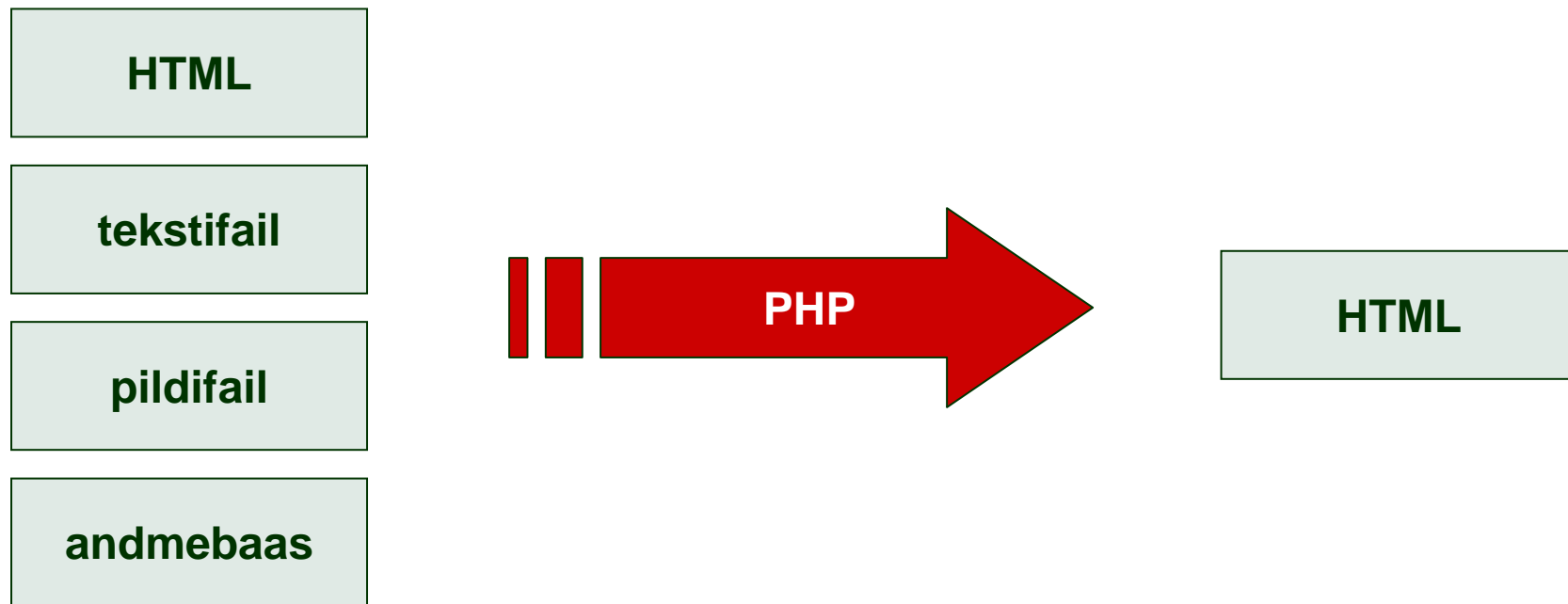
# Sissejuhatus

Ivari Horm

ranger@risk.ee

- *PHP: Hypertext Preprocessor*
- <http://www.php.net/manual/en>
- <http://lib.risk.ee/php>

- HTML kirjeldab kodulehekülje välimust
- PHP abil tekitatakse sobiv HTML-keelne dokument



**<script language="php">, </script>**

- HTML-i käsud, mille vahel peab PHP kood asuma

**<?php , ?>**

- Lühendatud variant

## PHP koodi paigutamine HTML-i

```
<html>
  <head></head>
  <body>

    <script language="php">

      //Siia tulevad PHP käsud

    </script>

  </body>
</html>
```



- Iga PHP käsk lõpeb semikooloniga
- Kui PHP-s on lubatud üks käsk, aga vaja on anda mitu käsku, tuleb nende ümber panna loogelised sulud
- Ühte käsku nimetatakse lihtlauseks

**// Kommentaar**

- Üherealine kommentaar.
- Võib PHP-s alata suvalisest kohast, kuid kestab alati rea lõpuni

**/\* Kommentaar \*/**

- Mitmerealine kommentaar
- Kestab kuni kommentaarimärgi lõpuni
- Võib omakorda sisaldada ka üherealisi kommentaare
- Tema sees ei tohi olla mõni teine mitmerealine kommentaar

## Kommentaarid

```
<script language="php">  
    . . .          //See on üherealine kommentaar  
    . . .  
  
    /* Siit algab mitmerealine kommentaar  
  
    //Siin on veel üks üherealine kommentaar  
  
    Siin lõpeb mitmerealine kommentaar */  
  
    . . .  
</script>
```

- Lõik PHP koodist, mis on kirjutatud loogeliste sulgude vahele
- Plokki alustavad loogelised sulud algavad üldjuhul eelmiselt realt ning lõpevad järgmisel real pärast viimast plokis olevat käsku

## Ploki kasutamise näide

```
<script language="php">
    print "This is the start of the PHP script";

    if (!my_personal_sql_connect()) {
        print "We could not connect to SQL server";
        print "I do not have the required information";
        print "I have to stop this script";
    }
    else {
        print "I successfully connected to the server";
        pass_the_query();
    }
    print "This is the end of the PHP script";
</script>
```

- **Stepping, indentation**
- Kõik ühte plokki kuuluvad käsud kirjutatakse vasakust servast ühekaugusele
- Uue ploki alustamisel nihutatakse taanet ühe võrra paremale, ploki lõpetamisel aga ühe võrra vasakule
- Normaalne taande suurus on 4 tühikut

# Andmetüübid ja muutujad

Ivari Horm

ranger@risk.ee

### Tõeväärtustüüp boolean

- Võib omada väärtust kas TRUE või FALSE

### Täisarvutüüp integer

- Suvaline täisarv vahemikust -2147483646 ... 2147483647

### Ujukomatüüp float

- Väärtused vahemikus  $\pm 1,8 * 10^{308}$

### Tekstitüüp string

- Suvaline märkide kombinatsioon
- **Kirjutatakse jutumärkide vahele!**



$$10 + 5 = 15$$



Alati kindel väärtus

$$10 + x = y$$



y väärtus muutub  
y väärtus oleneb x-st

- ✓ x ja y on muutujad, nad võivad omada erinevaid väärtusi
- ✓ Arvu 10 väärtus ei muutu, teda nimetatakse **konstandiks**

## `$muutuja_nimi = väärtus`

- Muutujaid tähistatakse PHP-s alati dollarimärgiga
- Muutuja nimi võib sisaldada ainult inglise tähestiku märke, numbreid ja alakriipsu
- Nimi ei tohi koosneda ainult numbritest ega alata tühikuga
- Soovitav on muutuja nime mitte alustada numbriga

## Muutujate väärtustamine

```
//Muutuja var_int saab väärtuseks 25:
```

```
$var_int = 25;
```

```
//Muutuja var_str väärtuseks saab tekst "Hello World":
```

```
$var_str = "Hello World";
```

```
//Muutuja var_bl saab väärtuseks TRUE:
```

```
$var_bl = TRUE;
```

# Jutumärgid

Ivari Horm

ranger@risk.ee

- PHP-s tuleb teksti (string) ümber alati panna jutumärgid

- Ühekordsed 'Hello World'
- Kahekordsed "Hello World"

- Ühekordsetes jutumärkides olevaid muutujaid ei asendata nende väärtustega
- Kahekordsetes jutumärkides olevad muutujate asemel väljastatakse nende väärtus

## Jutumärkide kasutamine

```
//Muutuja var_int saab väärtuseks 25:  
$var_int = 25;
```

```
//Muutuja var_out väärtuseks saab tekst "My age is 25":  
$var_out = "My age is $var_int";
```

```
//Muutuja var_out väärtuseks saab tekst "My age is $var_int":  
$var_out = `My age is $var_int`;
```



- Ühte tüüpi jutumärkide sees saab kasutada teist tüüpi jutumärke
- Sama tüüpi jutumärkide kasutamisel tuleb need kindlasti kirjutada paogadana

## Jutumärkide kasutamine üksteise sees

```
//Erinevad jutumärgid
$var_ds = "This is the 'Hello World' program";
$var_sd = 'This is the "Hello World" program';

//Sama tüüpi jutumärgid
$var_dd = "This is the \"Hello World\" program";
$var_ss = 'This is the \'Hello World\' program';

//Jutumärkide paarsus pole seejuures oluline
$var_d = "I'm 25 years old";
$var_s = 'I\'m 25 years old';
```

# Operaatorid

Ivari Horm

ranger@risk.ee

- Muutujatesse salvestatud andmetega saab teha tehteid
- Tehted tehakse **operaatorite** abil
- Tehtes osalevaid muutujaid nimetatakse **operandideks**
- Korrutamine – korrutamisoperaator  
Liitmine – liitmisoperaator

- **Unaarsed**  
Operaator nõuab ühte operandi
- **Binaarsed**  
Operaator nõuab kahte operandi
- Binaarne operaator ei tähenda tehet kahendarvudega!

$$\$a = \$b$$

- Omistamine
- Muutujate **a** väärtuseks saab muutuja **b** väärtus

$$\$a + \$b$$

- Liitmine

$$\$a - \$b$$

- Lahutamine

$$\$a * \$b$$

- Korrutamine

$$\$a / \$b$$

- Jäägiga jagamine

$a == b$

- Tõene, kui **a** ja **b** on võrdsed
- **NB! Kaks võrdusmärki!**

$a <> b$ ,  $a != b$

- Tõene, kui **a** ja **b** ei ole võrdsed

$a > b$

- Tõene, kui **a** on suurem kui **b**

$a < b$

- Tõene, kui **a** on väiksem kui **b**

$a \geq b$

- Tõene, kui **a** on suurem kui **b** või võrdne **b**-ga

$a \leq b$

- Tõene, kui **a** on väiksem kui **b** või võrdne **b**-ga



## \$a++

- Inkrement
- Suurendab **a** väärtust ühe võrra
- $\$a = \$a + 1$

## \$a--

- Dekrement
- Vähendab **a** väärtust ühe võrra
- $\$a = \$a - 1$

## Operaatorite kasutamine

```
$x = 5;
```

```
$y = 10;
```

```
$z = 10;
```

```
$x == $y;
```

```
//FALSE
```

```
$x > $y;
```

```
//FALSE
```

```
$x <= $y;
```

```
//TRUE
```

```
$y <= $z;
```

```
//TRUE
```

```
$z++;
```

```
//11
```

```
$z = $x;
```

```
//TRUE, z=5;
```

**\$a.\$b**

- Konkatenatsioon (andmestruktuuride ühendamine)
- Muutujate **a** ja **b** väärtused liidetakse tekstiliselt

**-\$a**

- Märgi pööramine
- Positiivne arv muudetakse negatiivseks ja vastupidi
- Kui muutuja **a** väärtuseks on tekst, siis väljastatakse null

## Operaatorite kasutamine

```
$x = 5;
```

```
$y = -10;
```

```
$z = 'a';
```

```
$x.$y; // 5-10
```

```
$x.$z; // 5a
```

```
-$y; // 10
```

```
-$z; // 0
```

# Sisend/väljund

Ivari Horm

ranger@risk.ee

- Kõige esmane, mis tuleb uue programmeerimiskeele juures endale selgeks teha
- Kui ei tea, kuidas toimub antud keeles sisend/väljund, ei ole võimalik midagi edasi õppida

```
print ("tekst");
```

- Trükib välja jutumärkide vahel oleva teksti
- Sulud võib selle käsu puhul üldjuhul ära jätta

## Väljund

```
<script language="php">  
  
    //This is the classical HelloWorld program  
  
    print "Hello World!";  
  
</script>
```



- Print-käsu puhul mõjuvad jutumärgid samamoodi, nagu varem kirjeldatud
- Ühekordsete jutumärkide korral tuleb muutujad jutumärkidest “välja võtta”

## Jutumärkide kasutamine väljundis

```
$a = 25;  
$y = "Yes";  
  
print "I am $a years old";           // I am 25 years old  
print `I am `.$a.`years old`;       // I am 25 years old  
print "Press '$y' to accept";       // Press 'Yes' to accept  
print "Press \"$y\" to accept";      // Press "Yes" to accept  
print `Press ``.$y.`` to accept`;   // Press "Yes" to accept
```

- PHP-skripti saab juhtida kas kodulehel oleva vormi või otse URL-i abil

```
/php/skript.php
```

```
print $i;      //Trükib välja 2
```

```
print $j;      //Trükib välja 3
```

```
print $k;      //Trükib välja 4
```



**<http://devel.risk.ee/php/skript.php>**

**i = 2 & j = 3 & k = 4**



```
http://devel.risk.ee/php/main.php?i=12
```

- Serveris käivitatakse fail **main.php** ning muutujale **i** antakse seejuures väärtus 12

```
http://devel.risk.ee/php/main.php?i=12&str="HelloWorld"
```

- Mitme muutuja algväärtustamisel tuleb nende vahele kirjutada ampersand (&)
- Serveris käivitatakse PHP-fail **main.php**, muutujale **i** antakse algväärtus 12 ja muutujasse **str** salvestatakse tekst "HelloWorld"

# Kotrollstruktuurid

Ivari Horm

ranger@risk.ee

- Võimaldavad juhtida PHP skripti tööd
- Jätta teatud tingimustel osa skriptist vahele
- Täita teatud tingimustel mõnda skriptilõiku mitu korda

- Tingimuslause
- Korduslause
- Valikulause
- ...

```
if (con)
  lihtlause 1;
else
  lihtlause 2;
```

- **con** on avaldis, mida tuleb kontrollida
- **lihtlause1** täidetakse juhul, kui tingimus on tõene (TRUE)
- **lihtlause2** täidetakse juhul, kui tingimus on väär (FALSE)



## Tingimuslause kasutamine

```
if ($new_mail==1)
    print "Sulle on tulnud uus kiri";
else
    print "Sinu postkastis pole uusi kirju";
```

## Mitme lihtlausega tingimus

```
if (!$auth_user) {
    print "Teil ei ole õigust antud lehekülge vaadata";
    print "Logige sisse kasutajaga, kellel see õigus on";
}
else {
    print "Teil on luba antud lehte külastada";
    print "Väljalogimiseks vajutage nupule \"Logout\"";
}
```

## while(con) lihtlause;

- **Lihtlause** täidetakse senikaua, kui tingimus **con** on tõene
- Enne **lihtlause** iga täitmist kontrollitakse tingimust **con**
- Kui tingimus on tõene, täidetakse **lihtlause**
- Kui tingimus on väär, asutakse järgmisi PHP käske täitma
- Kui tingimus **con** on väär juba enne esimest täitmist, siis **lihtlause** ei täideta kordagi

## WHILE-lause kasutamine (1)

```
while ($i<6) {  
    print $i;  
    print "<br />";  
    $i=$i+1;  
}
```

## WHILE-lause kasutamine (2)

```
$i=2;  
while ($i) {  
    print "Rida $i";  
    print "<br />";  
}
```

```
for(eval1; con1; eval2)  
  lihtlause;
```

- Käsk **eval1** täidetakse üks kord FOR-käsuni jõudmisel
- Tingimust **con1** kontrollitakse iga kord enne **lihtlause** täitmist
- Kui **con1** on tõene, täidetakse **lihtlause**
- Kui **con1** on väär, lõpetatakse kogu protsess ja asutakse FOR-käsust järgmisi käske täitma
- Käsk **eval2** täidetakse iga kord pärast **lihtlause** täitmist

## Korduslause kasutamine

```
for ($i=0; $i<6; $i=$i+1) {  
    print $i;  
    print "<br />";  
}
```

## Inkremendi kasutamine

```
for ($i=0; $i<6; $i++) {  
    print "Rida $i";  
    print "<br />";  
}
```

## Kõige lühem variant eelnevast

```
for ($i=0; $i<6; $i++)  
    print "Rida $i <br />";
```

```
switch ($var) {  
  case val1:  
    lause1;  
    lause2;  
    break;  
  case val2:  
    lause3;  
    break;  
  default:  
    lause4;  
    break;  
}
```

- Valikulauseni jõudmisel kontrollitakse muutuja **var** väärtust
- Kui **var** väärtus on **val1**, täidetakse **lause1** ja **lause2**
- Kui **var** väärtus on **val2**, täidetakse **lause3**
- Valikute **val1**, **val2** arv ei ole piiratud
- Valik **default** täidetakse, kui ükski muu võimalik variant ei sobinud
- Käsu **break** abil väljutakse switch-lausest

## Valikulause kasutamine

```
switch ($day_nr) {  
    case 1:  
        print "Täna on esmaspäev";  
        break;  
    case 2:  
        print "Täna on teisipäev";  
        break;  
    case 3:  
        print "Täna on kolmapäev";  
        break;  
    default:  
        print "Täna on mingi muu päev";  
        break;  
}
```

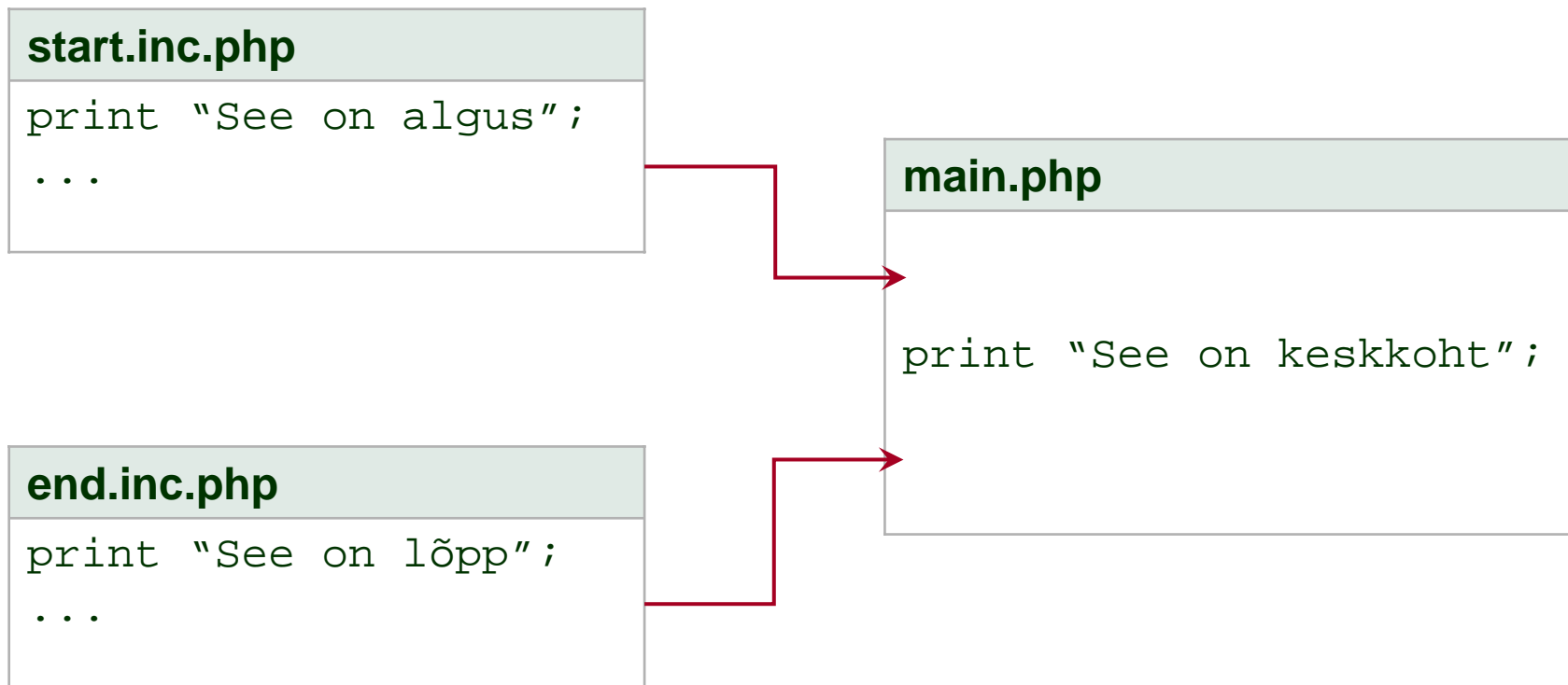
- Miks on vaja valikulauses kasutada break-käsku?
- <http://lib.risk.ee/php>



# Mitme faili kasutamine

Ivari Horm

ranger@risk.ee



```
include("sf1");
```

- Failis nimega **sf1** olev tekst lisatakse PHP koodis selle koha peale, kus asub include-käsk
- Faili otsitakse samast kaustast asub skriptifail, milles include-käsk tehti
- Otsitakse ka süsteemsetest kaustadest, mis on PHP konfiguratsioonifailis määratud
- Antud käsu puhul võib sulud ära jätta

## Failide lisamine main.php skriptifaili

```
include "start.inc.php";
```

```
print "See on keskkohht";
```

```
include "end.inc.php";
```

- Kataloogipuus saab liikuda allapoole, kui faili nime ette kirjutada kataloogi nimi  
*dir1/dir2/file.php*
- Ülespoole liikumiseks tuleb kirjutada topelpunkt  
*../.../other\_dir/file.php*
- Säärast kirjaviisi saab kasutada ka include-lauses

**dirname("sf1");**

- Väljastab teekonna alates juurkataloogist, mida mööda on vaja liikuda, et jõuda failini **sf1**
- Kui "**sf1**" asemele kirjutada **\_\_FILE\_\_**, kuvatakse teekond antud skriptifaili kohta, milles dirname-käsk asub
- Kõige kindlam viis include-käsku kasutada

**Failide lisamine main.php skriptifaili, kasutades dirname-käsku**

```
include dirname(__FILE__)."/start.inc.php";  
  
print "See on keskkohht";  
  
include dirname(__FILE__)."/end.inc.php";
```