



Veebiarendaja algkoolitus

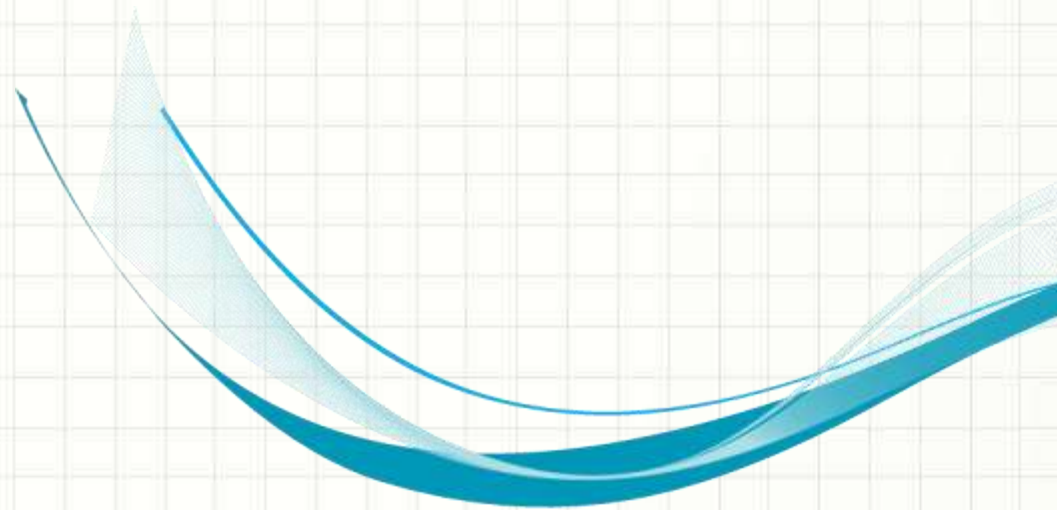
Andres Järviste
ver 2

Kursuse eesmärk

Veebiarendaja algkoolitus on mõeldud veebiarendusega alustavatele inimestele esmaseks HTML ja PHP-keele tundmaõppimiseks ja sobib hästi WordPressi aktiivsetele kasutajatele WordPressi kursuse jätkukoolituseks.

Kursusel käsitletakse standardseid veebitehnoloogiaid (HTML, CSS, Javascript, PHP), mis laiendavad omanäoliste ja unikaalsete veebilehtede tegemise võimalusi.

Koolitusel õpetatavad oskused lubavad luua standardsetel komponentidel põhinevatest veebilehtest veelgi kasutajasõbralikumaid ja omanäolisemaid veebikeskkondi.



**HTML, CSS,
JavaScript, PHP - miks
nii palju erinevaid
veebitehnoloogiaid?**



HTML

Hyper Text Markup Language

HTML keeles luuakse veebilehekülgede **sisu**

CSS

Cascading Style Sheets

Selle tehnoloogia abil luuakse
veebilehekülgede **kujundus**

Vaata: www.csszengarden.com



JavaScript

Programmeerimiskeel progammide ja interaktiivsuse loomiseks **veebilehitsejas**

Vaata: <http://www.w3schools.com/js/default.asp>

PHP

Programmeerimiskeel, mille abil saab serveris luua dünaamiliselt veebilehti (HTML, CSS, JavaScripti komplekte), mis moodustavad suuri **veebisaite ja veeebilehitsejas kasutatavaid programme**

PHP-s on loodud näiteks Wordpress

Vaata: <http://www.w3schools.com/php/default.asp>



Kasutatavad vahendid:

- **Tekstiredaktor Visual Studio Code** veebilehekülgede ja programmifailide loomiseks ja muutmiseks ning veebiserverisse laadimiseks
- **Veebilehitseja Google Chrome** ja selle arendustööriistad veebilehekülgede lähtekoodi vaatamiseks ja käigult muutmiseks
- **FTP programm Filezilla** veebilehtede laadimiseks serverisse



Kasutatavad vahendid:

- **Tekstiredaktor NotePad++ või Visual Studio Code**
veebilehekülgede ja programmifailide loomiseks ja muutmiseks ning veebiserverisse laadimiseks
- **Veebilehitseja Google Chrome ja selle arendustööriistad**
veebilehekülgede lähtekoodi vaatamiseks ja käigult muutmiseks

HTML põhitõed

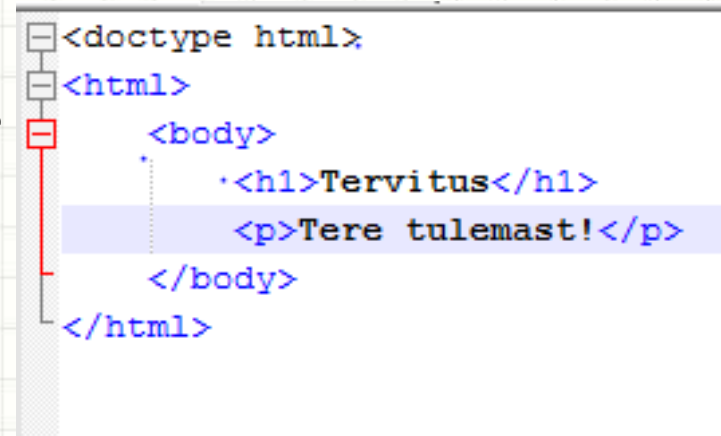
HTML

- See on keel veebilehtede sisu kirjeldamiseks
- HTML on **märgendipõhine** (markup) keel, koosnedes hulgast siltidest ehk **tagidest**
- Tagid kirjeldavad dokumendi sisu
- HTML dokument võib koosneda tagidest ja tavalisest tekstist
- HTML dokumente kutsutakse ka **veebilehtedeks**

```
<doctype html>
<html>
  <body>
    <h1>Tervitus</h1>
    <p>Tere tulemast!</p>
  </body>
</html>
```

HTML tagid ja - elemendid

- HTML tagid on võtmesõnad, mida tavaliselt ümbritsevad märgendid '<' ja '>'. Näiteks <html>
- Tage kasutatakse tavaliselt **paaris**, näiteks <p> ja </p>. Esimene paaris on **algustag** ja teine **lõpptag**
- Lõpptag on tavaliselt samasugune kui algustag välja arvatud märgend '/' enne tagi nime
- Tihti öeldakse, et HTML „tagid“ ja „elemendid“ on samad asjad. Rangelt võttes on erinevus selles, et element on kõik, mis jääb alg- ja lõpptagi vahele ning lisaks ka alg- ja lõpptag ise.
- Elemendi näide:
<p>See on paragrahv</p>



```
<doctype html>
<html>
  <body>
    <h1>Tervitus</h1>
    <p>Tere tulemast!</p>
  </body>
</html>
```

<!Doctype> deklaratsioon

<!Doctype> deklaratsioon aitab veebilehitsejatel erinevate HTML versioonidega loodud veebilehti õigesti näidata

Kuna veebis on palju erinevaid HTML dokumente, siis saab brauser lehti näidata õigesti vaid siis, kui ta teab, millises HTML-i versioonis leht on tehtud

| Version | Year |
|-----------|------|
| HTML | 1991 |
| HTML+ | 1993 |
| HTML 2.0 | 1995 |
| HTML 3.2 | 1997 |
| HTML 4.01 | 1999 |
| XHTML 1.0 | 2000 |
| HTML5 | 2012 |
| XHTML5 | 2013 |

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

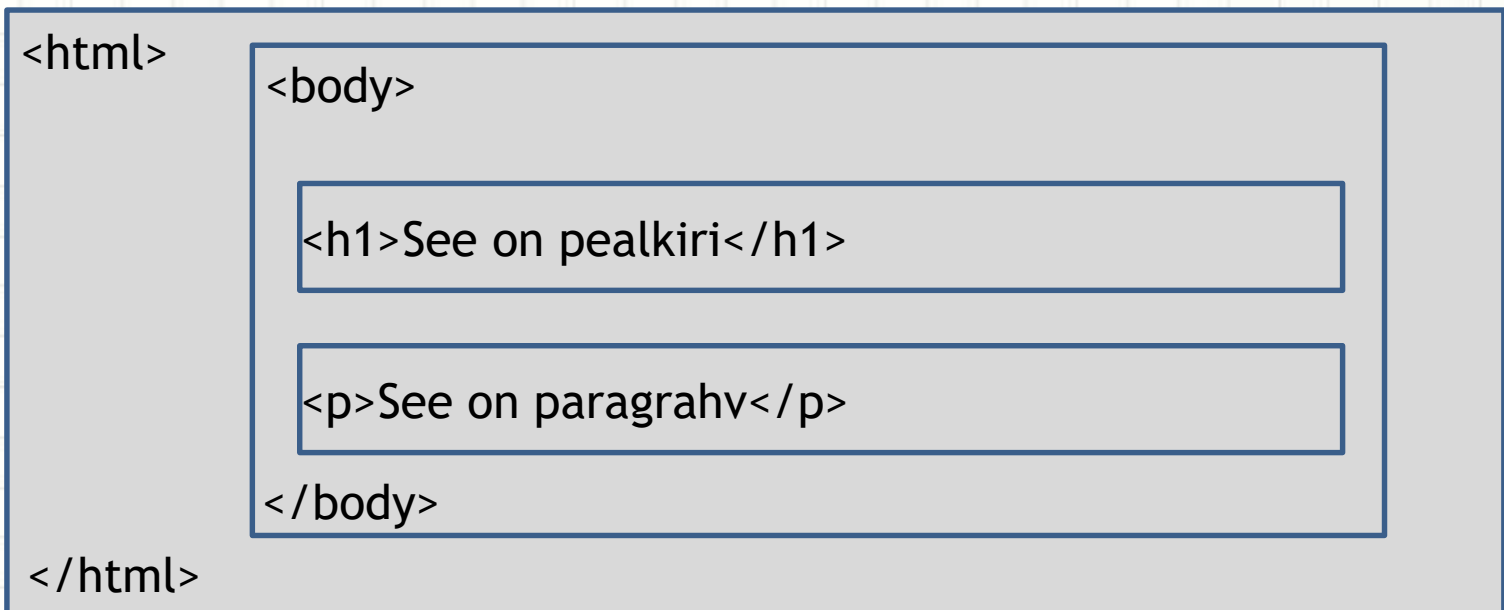
XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML lehe struktuur

- Kõik mis on tagide `<html>` ja `</html>` vahel määratleb veebilehe
- Kõik mis on tagide `<body>` ja `</body>` vahel moodustab nähtava lehe sisu

```
<doctype html>
<html>
  <body>
    <h1>Tervitus</h1>
    <p>Tere tulemast!</p>
  </body>
</html>
```

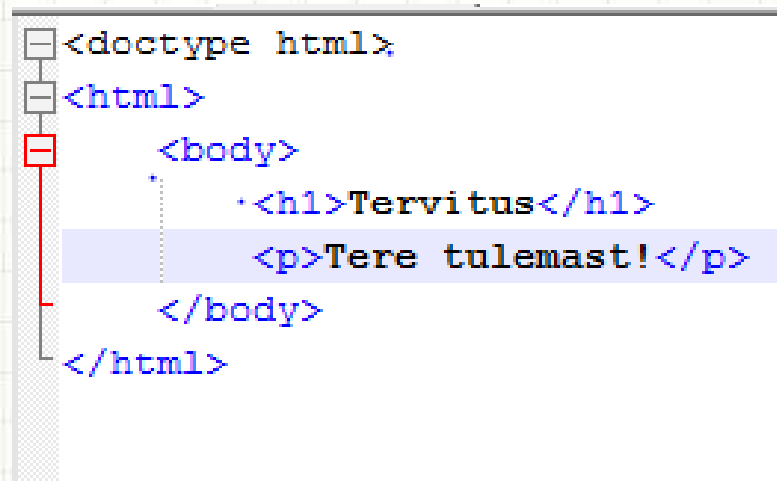


Kõige põhilisemad märgendid (tagid)

| Algtag | Elemendi selgitus |
|-------------------------------------|---|
| <code><h1></code> | <i>Esimese taseme pealkiri. Võib kasutada kuni 9 tasemt pealkirjasid kujul <hX> kus X on number 1 .. 9</i> |
| <code><p></code> | <i>Paragrahv. Peale paragrahvi tagi lõppu algab järgnev tekst uuel realt</i> |
| <code></code> | <i>Oluline tekst ehk rasvane. Varem on samal eesmärgil kasutatud tagi </i> |
| <code></code> | <i>Eriliselt välja toodav tekst. Enamalt jaolt tähendab see kursiivis teksti. Varem on kasutatud samal eesmärgil tagi <i></i> |
| <code></code> | <i>Link teisele veebilehele. Tagi sees näidatakse viide teisele veebilehel atribuudina. s.t. kujul href= „url“ Elemendi näide: <code>Otsingumootor</code></i> |
| <code>, </code> | <i> - Järjestamata loetelu. Loetelu koosneb loetelu ridade elementidest – tag . Näide:ÕunadApelsinid</i> |
| <code></code> | <i>Loetelu rida</i> |
| <code> </code> | <i>Reavahetus paragrahvi sees. NB! See tag ei vaja sulgevat tagi ning kogu element koosneb ainult algtagist.</i> |
| <code></code> | <i>Pilt veebilehel. Järgmine näide näitab pilti samast kataloogist kus on ka veebileht: <code></code> NB! Ka see element koosneb ainult algtagist. Pildi laiuse ja kõrguse muutmiseks saab kasutada atribuute width ja height</i> |

Harjutus1: HTML katsetused

- Looge oma esimene veebileht Notepad++ või Visual Studio Code abil
- Lisage pealkiri ning sisu eelmisel slaidil olevate HTML elementide abil.
- Salvestage loodud fail (faili laiend peab olema .html)
- Faili vaatamiseks leidke Exploreri abil fail ning avage see.

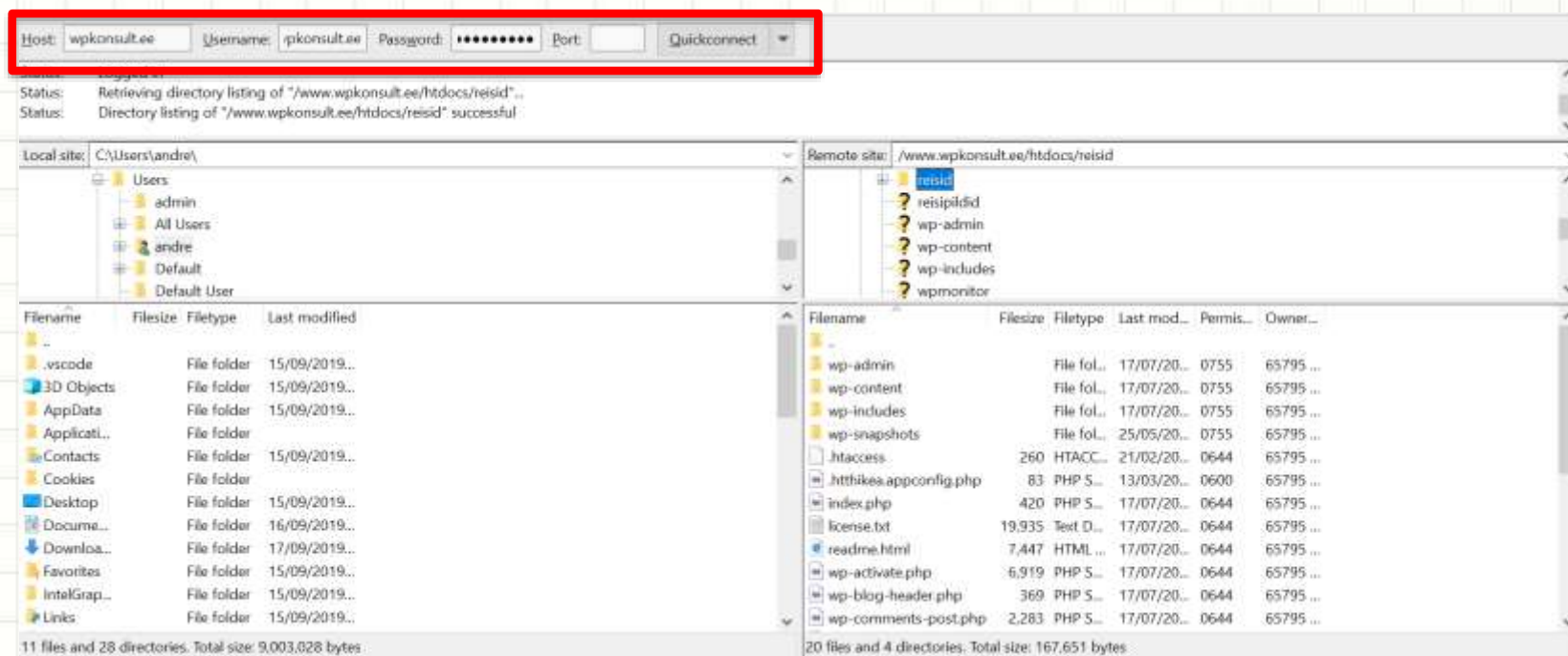


```
<doctype html>
<html>
  <body>
    <h1>Tervitus</h1>
    <p>Tere tulemast!</p>
  </body>
</html>
```

The screenshot shows a code editor window with a tree view on the left. The tree view has three nodes: '<doctype html>', '<html>', and '<body>'. The '<body>' node is expanded, showing its children: '<h1>Tervitus</h1>' and '<p>Tere tulemast!</p>'. The '<p>Tere tulemast!</p>' line is highlighted in blue. The main editor area shows the full HTML code as shown in the pre-block above.

FTP ühenduse loomine

- Täida väljad host, username, password ja vajuta nupule Quickconnect



FTP parameetrid

- XX - Sinu number
- Host (server): wpkonsult.ee
- User (kasutaja): veebXX.wpkonsult.ee
- Password (parool): vaata tahvlile
- Sinu kodulehe aadress on nüüd:
<http://wpkonsult.ee/oppurid/veebXX/>
- Lae üles fail index.html ja proovi avada veebilink

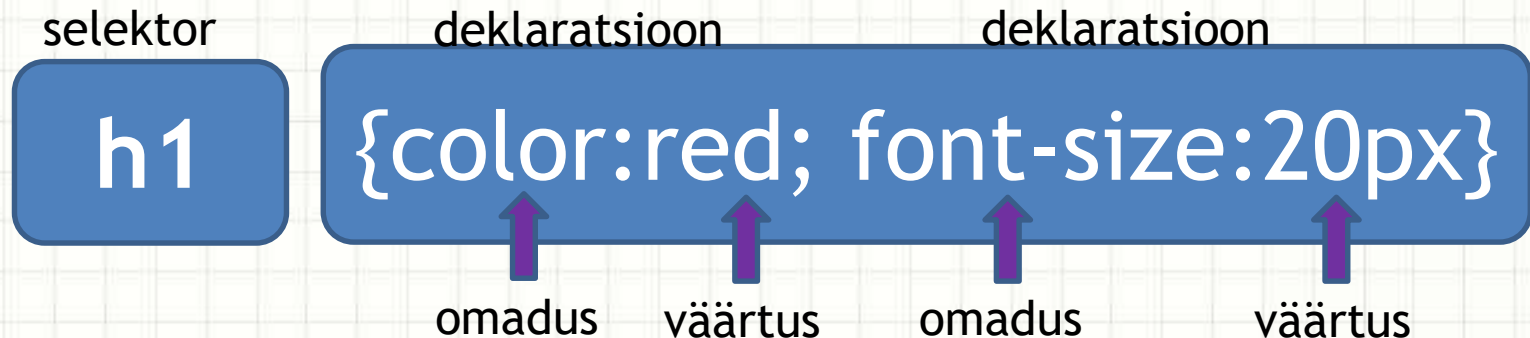
Sissejuhatus CSS-i

CSS - Cascading Style Sheets

- CSS on mehhanism, mille abil määratakse kuidas näidata HTML elemente.
- Sama sisuga veebilehel võib olla palju erinevaid kujundusi.
- CSS reegleid võib kirjeldada HTML dokumendi sees (nii päises kui kehas) elemendi `<style>` abil
- Mitmetel veebilehtedel võib olla sama kujundus.
- Seetõttu on parim praktika luua CSS definitsioone eraldiseisvate failidena. Suuremate ja taaskasutatavate tööde loomisel ei ole soovitatav hoida stiile HTML dokumendi sees

```
<doctype html>
<html>
<style>
  body {
    background-color:#CCFF99;
  }
  h1 {
    color:orange;
    text-align:center;
  }
  ul {
    font-family:"Times New Roman";
    font-size:24px;
  }
</style>
  <body>
    <h1>Tervitus</h1>
    
    <p>Tere tulemast!</p>
    <ul>
      <li>Õunad</li>
      <li>Apelsinid</li>
    </ul>
  </body>
</html>
```

CSS süntaks



- CSS koosneb **reeglitest**, millest igaüks koosneb kahest osast: selektorist ja ühest või mitmest deklaratsioonist
- Selektor on võib olla HTML element, mille stiili soovetakse määrata
- Iga deklaratsioon koosneb tavaliselt **omadusest** ja selle **väärtusest**. Deklaratsioon lõppeb alati märgiga ‘;’. Parema loetavuse pärast pannakse deklaratsioonid tavaliselt eraldi ridadele
- Reeglite ees ja järel kasutatakse kommentaare - tekste mis asuvad märgendite ‘/*’ ja ‘*/’ vahel, et loodud kood oleks lihtsalt arusaadav

Id ja Class tüüpi selektorid

- Lisaks HTML elementidele stiili määramisele, võib ka ise määrateda stiile enda kirjeldatud selektorite abil.
- Kasutatakse kahte tüüpi selektoreid: „id“ ja „class“
- „id“-tüüpi selektoreid kasutatakse ühe unikaalse elemendi stiili kirjeldamiseks ning seda tüüpi selektorid algavad sümboliga ‘#’
- Stiili kasutamiseks tuleb HTML algtagis määratleda stiil atribuudi „id“ abil. Näiteks stiili selektoriga #param1 kasutamiseks tuleb tagis kirjeldada atribuut id=„param1“
- „class“-tüüpi selektoreid kasutatakse teatud hulga elementide stiili kirjeldamiseks ning seda tüüpi selektorid algavad sümboliga ‘.’
- Stiili kasutamiseks tuleb elemendi algustagis määratleda stiil atribuudi „class“ abil.

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1
{
text-align:center;
color:orange;
}
</style>
</head>
<body>
<p id="para1">Tere tulemast!</p>
<p>Seda paragrahvi stiil ei mõjuta</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
.center
{
text-align:center;
}
</style>
</head>
<body>
<h1 class="center">Pealkiri keskel</h1>
<p class="center">Paragrahv keskel</p>
</body>
</html>
```

Mõned CSS omadused

| Omadus | Kirjeldus | Võimalikud väärtused |
|-------------------------|------------------------------|---|
| <i>background-color</i> | <i>Elemendi taustavärv</i> | <i>värvikood (näiteks #CCFF99) või värvi nimetus inglise keeles, näiteks „red“</i> |
| <i>color</i> | <i>Teksti värv</i> | <i>värvikood (näiteks #CCFF99) või värvi nimetus inglise keeles, näiteks „red“</i> |
| <i>text-align</i> | <i>teksti joondamine</i> | <i>center, right, left – joondatud vastavalt kesksele, paremale või vasakule</i> |
| <i>font-family</i> | <i>Elemendi teksti font</i> | <i>Tavaliselt väärtuste komplekt. Kui brauser ei leia esimest fonti, proovib otsida teist. Näide: font-family:“Times New Roman“, Times, serif Täpsemate kombinatsioonide vaatamiseks vaata: http://www.w3schools.com/cssref/css_websafe_fonts.asp</i> |
| <i>font-size</i> | <i>Teksti fondi suurus</i> | <i>Fondi suurus näidatakse kas pikslites (ühik px) või kordajana brauseri normaalfondi suuruselt (ühik em) Näiteks: h1 {font-size:40px;} ja h1 {font-size:2.5em;}</i> |
| <i>text-decoration</i> | <i>Kuiunduste emaldamine</i> | <i>none, underlined – eemaldatud, allajoonitud. Täpsemaid ja täielikemaid CSS omaduste kirjeldusi vaata: linkidelt joone <i>text-decoration:none;}</i></i> |

- <http://www.w3schools.com/css/default.asp> (Inglise keeles)
- <http://et.wikibooks.org/wiki/Esileht> (Eesti keeles)

Harjutus2: CSS katsetused

- Lisa enda loodud lehele stiili element ja lisa mõned kujunduseeglid, mis muudavad
 - mõne elemendi taustavärvi
 - kõigi elementide tekstivärvi
- Kui sinu lehel ei ole veel linki, siis lisa see.
- Muuda linki nii, et see ei oleks automaatselt alla joonitud

```
<doctype html>
<html>
<style>
  body {
    background-color:#CCFF99;
  }
  h1 {
    color:orange;
    text-align:center;
  }
  ul {
    font-family:"Times New Roman";
    font-size:24px;
  }
</style>
  <body>
    <h1>Tervitus</h1>
    
    <p>Tere tulemast!</p>
    <ul>
      <li>Õunad</li>
      <li>Apelsinid</li>
    </ul>
  </body>
</html>
```

Stiilide kaasamine veebilehele

- **Välise failina** - kõige levinum moodus. Kasutatakse, kui on vaja sama stiili rakendada mitmes veebidokumendis või kui on soov samal veebilehel kasutada erinevaid kujundusi:

```
<link rel="stylesheet" type="text/css" href="stiil.css" media="screen" />
```

kasutatakse tagi link ning argumenti href abil näidatakse kaasatava faili asukoht. Argument **type** näitab, et link on css fail

- **Sisemise stiilikirjeldusena** - HTML faili päises kirjeldatakse tagi <style> abil stiil. Miinus - stiil on seotud konkreetse HTML-failiga. Seda moodust kasutatakse harjutuse juures
- „**Inline**“ stiil - Stiil defineeritakse tagi atribuudina:

```
<p style="color:sienna;margin-left:20px">See on paragrahv</p>
```

Seda moodust kasuta ettevaatlikult: nii kirjeldatud stiili muutmine hiljem võib olla väga töömahukas

- Stiililehed moodustavad nn. **kaskaadi**. See tähendab, et kui sama element on kirjeldatud mitmes erineva reegli abil, siis reeglina kehtib hiljem defineeritud reegel

Kastmudel (Box model)



- Kastmudel tähendab, et disaini mottes on kõik elemendid „kastid“, millel on järgmised omadused:
 - **Margin** - Vaba ala väljapool elemendi piirjoont, mille suurust saab määrata
 - **Border** - Piirjoon, mille kuju, paksust ja värvi saab määrata
 - **Padding** - Vaba ala piirjoone ja sisu vahel, mille suurust saab määrata
 - **Content** - Elemendi sisu
- Omadustega `width` ja `height` näidatakse elementide laiust ja kõrgust. Nii näidatud laius ja kõrgus kehtib sisule - omaduste `Padding`, `Border` ja `Margin` laius/kõrgus lisandub alati eraldi

Div tagid ja lehe struktuur

HTML - i DIV - tagide ja nende stiilide abil luuakse veebilehe erinevad alad nagu lehe päis, jalus, külgpaan(id), sisuala

```
<div id="header" style="background-color:#FFA500;">  
<h1 style="margin-bottom:0;">Lehekülje pealkiri</h1></div>
```



 - DIV elementide ja nende stiilidega kujundatud lehe osad

Harjutus 3: Lehe kujunduse muutmine DIV elementide stiili muutmise abil

- Vaadake harjutuse koodifaili (järgmisel slaidil). Avage see Notepad++'iga
- Vaadake, kas te saate aru, milliseid stiilireegleid on kasutatud. Abiks võib võtta veebiraamatu: <http://et.wikibooks.org/wiki/CSS>
- Eemaldage inline-stiilikirjeldused ning eraldage need omaette stiilifaili. Liitke stiilifail algsele failile. Kontrollige, kas HTML tulemus on jätkuvalt samasugune
- Muutke lehe kujundust - näiteks tsentreerige pealkiri, muutke fontide värvust jne.

Harjutus 3: Lehe kujunduse muutmine DIV elementide stiili muutmise abil

- Harjutuse fail:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <div id="container" style="width:500px">
    <div id="header" style="background-color:#FFA500;">
      <h1 style="margin-bottom:0;">Lehekülje pealkiri</h1></div>
    <div id="menu" style="background-color:#FFD700;height:200px;width:100px;float:left;">
      <b>Menüü</b><br>
      HTML<br>
      CSS<br>
      JavaScript</div>
    <div id="content" style="background-color:#EEEEEE;height:200px;width:400px;float:left;">
      Sisu tuleb siia</div>
    <div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
      Sinufirma koduleht</div>
  </div>
</body>
```



Sissejuhatus PHP- sse

Mis on PHP?

- PHP (PHP: Hypertext Preprocessor) on serveris töötav skriptimiskeel.
- See tähendab, et PHP failid on programmid, mis käivitatakse serverarvutis ning mille tulemus saadetakse kasutaja veebilehitsejasse
- PHP fail võib sisaldada teksti, HTML tage ja skripte
- Kasutajani jõuab tavaline veebileht, mis on PHP skripti töö tulemus
- PHP skriptifailidel on tavaliselt laiend .php , .pp3 või .phtml
- PHP skriptide kasutamiseks tuleb:
 - Loodud skriptifail laadida FTP-ga serverisse ning siis brauseriga avada vastav URL
 - Installeerida enda töomasinasse veebiserver ja PHP mootor ning kasutada enda töomasinat kui serverit kopeerides veebiserveri dokumentide kataloogi skriptifail ning pöörduda selle poole veebiserveri kaudu

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
</head>
<body>
  <?php
    $kuupaev = date("d.m.Y");
    echo "<h1>Tervitus</h1>";
    echo "<p>Tere Maailm! Täna on $kuupaev</p>";
  ?>
</body>
</html>
```

Muutujad, funktsioonid, väljendid

Muutujad

- Matemaatikas ja PHP-s on nii et kui $x=3$ ja $y=4$, siis $x+y=7$
- X ja Y on siin **muutujad**
- Muutujate nimed algavad sümboliga '\$'
- Tehakse vahet suurtel ja väikestel tähtedel ja ei tohi sisaldada tühikuid
- Muutujatele võib omistada erinevat tüüpi väärtuseid: numbreid, teksti, kuupäevi või muutujate hulki ehk massiive.
- Muutujatega saab sooritada tehteid - liita (+) saab numbreid, ja kuupäevi, korrutada ja jagada numbreid (*, /), muutujate väärtusi üksteisega võrrelda (==). Kasutamine: **Muutuja = OmistatavVäärtus**

Funktsioonid

- Käskluste jada, mis võib tagastada konkreetse väärtuse. Näiteks `date(„d.m.Y“)` leiab ja tagastab käesoleva kuupäeva. Funktsioonid on alati kujul **nimi(parameeter1, võimalik_parameeter2, ...)**

Väljendid

- Funktsiooni(de) väljakuts(ete) , muutujate ja tehete kogum, mis moodustab loogilise terviku. Tarvaliselt on üks väljend ühel real- **Väljend lõppeb alati semikooloniga (;)**

PHP käsklused/väljendid

- PHP keeles on rida keelele omaseid käsklusi/väljendeid. Näiteks `echo „Mingi tekst“` väljastab teksti *Mingi tekst*.

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
</head>
<body>

<?php

$kuupaev = date("d.m.Y");
echo "<h1>Tervitus</h1>";
echo "<p>Tere Maailm! Täna on $kuupaev</p>";
?>

</body>
</html>
```

PHP tekstid ehk stringid

- Tekstmuutujaid ehk stringe kasutatakse tekstide hoidmiseks ja muutmiseks
- Lisaks aritmeetilistele tehetele ehk operaatoritele on olemas veel hulk muid operaatoreid
- Näiteks operaatorit ‘.’ kasutatakse stringide liitmiseks. Näiteks kui muutujale \$tekst omistada väärtuseks
\$tekst = ‘Õunad’ . ‘Aplesinid’
Saame tulemuseks ÕunadApelsinid
- Tekst kirjeldatakse sümbolite “ või ‘ abil.
- Käskusega echo saame väljastada ka muutujate väärtusi: näiteks
echo \$tekst
väljastab antud juhul teksti ‘ÕunadApelsinid’
- Teksti sees saab kasutada ka muutujaid, kuid ainult siis kui tekst on defineeritud “ abil - vaata kõrvalasuvat näidet, kus teksti sees on kasutatud muutujat \$kuupaev. Samas võtavad tekstid mis on defineeritud ‘ abil vähem arvutiressursse.

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
</head>
<body>
  <?php
    $kuupaev = date("d.m.Y");
    echo "<h1>Tervitus</h1>";
    echo "<p>Tere Maailm! Täna on $kuupaev</p>";
  ?>
</body>
</html>
```

Harjutus 4: Esimesed PHP katsetused

- Looge kõrvaloleva näite eeskujul oma esimene PHP skript
- Omistage kahele muutujale numbrilised väärtused. Liitke need
- Väljastage tulemus koos selgitava tekstiga
- Laadige skript FTP-ga serverisse ja vaadake, milline on tulemus

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
</head>
<body>
  <?php
    $kuupaev = date("d.m.Y");
    echo "<h1>Tervitus</h1>";
    echo "<p>Tere Maailm! Täna on $kuupaev</p>";
  ?>
</body>
</html>
```


If .. Else käsklus

- Tihti esineb olukordi kus sõltuvalt mingist tingimusest on tarvis ühel juhul teha üht ja teisel juhul teist.
- Sellistes olukordades kasutatakse If .. Else väljendit
- Ehk kui on täidetud mingi tingimus, täidetakse mingi kood ning kui tingimus ei ole täidetud, täidetakse alternatiivne kood.

```
$d=date("D");  
if ($d=="Fri")  
{echo "Head nädalavahetuse algust!";  
}  
else  
{ echo "Head päeva!";  
}
```

```
If (tingimus)  
{ kood, mis täidetakse kui tingimus  
on täidetud  
}  
Else  
{ kood, mis täidetakse kui tingimus  
ei ole täidetud  
}
```

Tsüklid - käsklus For

- Tsükleid kasutatakse kui mingit koodi on vaja täita korduvalt.
- For tsüklit kasutatakse juhul, kui on ette teada, mitu korda tsüklit täidetakse
- Parameetrid:
 - Init - kood mis täidetakse tsükli alguses. Tavaliselt kasutatakse loenduri algväärtuse seadmiseks
 - Condition - tingimus, mida kontrollitakse iga tsükli täitmise järel. Kui tingimus on täidetud (ehk TRUE), siis täitmist jätkatakse. Kui ei, on tsükel lõppenud
 - Increment - kood, mis täidetakse iga tsükli täitmise järel. Tavaliselt kasutatakse loenduri väärtuse suurendamiseks

```
for (init; condition; increment)
{
    code to be executed;
}
```

```
for ($i=1; $i<=5; $i++)
{
    echo "Number on " . $i . "<br />";
}
```

Harjutus 5: Tingimused ja tsüklid

- Lisage eelmises harjutuses loodud programmi tsükkel, mis näiteks sõltuvalt nädalapäevast väljastab mingitel päevadel ühe ja teistel teise tervitusteksti
- Mida peaks tegema et muuta tingimust nii, et tervitustekst oleks enne 2012 aastat ühesugune ja peale seda teistsugune?
- Looge tsükkel, mille abil kuvage numbreid 1-st 5-ni
- Mida peaks tegema, et näidata numbreid 100-st 150-ni?

```
$kuupaev = date("d.m.Y");  
echo "<h1>Tervitus</h1>";  
$d=date("D");  
echo "<p>";  
if ($d=="Fri")  
{echo "Head nädalavahetuse algust!";  
}  
else  
{ echo "Head päeva!";  
}  
echo "</p>";  
for ($i=1; $i<=5; $i++)  
{  
    echo "Number on " . $i . "<br />";  
}  
?>
```

Massiivid

- Massiiv on muutuja, milles talletatakse rohkem kui ühte väärtust
- Esimeses näites toodud muutujaid ei ole mugav ühes tsüklis töödelda.
- Massiiv defineeritakse kujul:
`$massiivi_nimi = array(väärtus1, väärtus2, ...)`
- Massiivi üksikuid elemente saab väärtustada või kasutada indeksi abil kujul
`$massiivi_nimi[indeks]`, milles esimese elemendi indeks on 0, teisel 1 jne

Näide 1: Hulga puuviljade salvestamisel muutujatesse võib teha nii:

```
$puuvili1="Õun";  
$puuvili2="Apelsin";  
$puuvili3="Pirn";  
$puuvili4="Ploom";
```

Näide 2: Massiivi abil saame ühe tsükliga väljastada kõigi massiivi elementide väärtused või ka üksikuid väärtusi:

```
echo "<h3>Hinnatavad puuviljad</h3>";  
$puuviljad = array("Õun", "Aplesin", "Pirn", "Ploom");  
for($i=0;$i<=3;$i++)  
{ echo "Puuvili $i: " . $puuviljad[$i]. "<br>";  
}  
echo "<h3>Tulemus:</h3>";  
echo "Parimad puuviljad on " . $puuviljad[0] .  
" ja " . $puuviljad[2];
```

Harjutus 6: Massiivide kasutamine

- Lihtsam
 - Loo programm, mis kõrvaloleva näite alusel loob massiivi ja näitab massiivi elemente
 - Mida peaks tegema, et lisada massiivi 3 uut elementi ning näidata neid vastupidises järjekorras?
- Keerulisem
 - Muutke harjutuses 4 valmistatud lehe makett PHP failiks
 - Muutke menüü väljastamine dünaamiliseks nii et menüü valikud on defineeritud massiivis ning need kuvatakse tsükli abil

```
<html>
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
</head>
<body>
<?php
echo "<h3>Hinnatavad puuviljad</h3>";
$puuviljad = array("Õun", "Aplesin", "Pirn", "Ploom");
for($i=0;$i<=3;$i++)
{   echo "Puuvili $i: " . $puuviljad[$i]. "<br>";
}
echo "<h3>Tulemus:</h3>";
echo "Parimad puuviljad on " . $puuviljad[0] .
      " ja " . $puuviljad[2];
?>
</body>
</html>
```

Funktsioonide loomine

- PHP-s on üle 700 funktsiooni
- Programmid jaotatakse loogilisteks osadeks isekirjeldatud funktsioonide abil
- Funktsioonid kirjeldatakse vastavalt Näitele
- Funktsioonil võivad (aga ei pea) olema sisendparameetrid
- Funktsioonid võivad (aga ei pea) tagastama väärtuse

```
Function funktsiooniNimi(par1, par2, ...)  
{ täidetavad koodiväljendid  
}
```

```
<?php  
function kirjutaNimi()  
{  
    echo "Eefraim Kolmeteistkümnes";  
}  
function liida($x,$y)  
{  
    $summa=$x+$y;  
    return $summa;  
}  
  
echo "Minu nimi on ";  
kirjutaNimi();  
echo "<br>4 + 20 võrdub: " . liida(4,20);  
?>
```

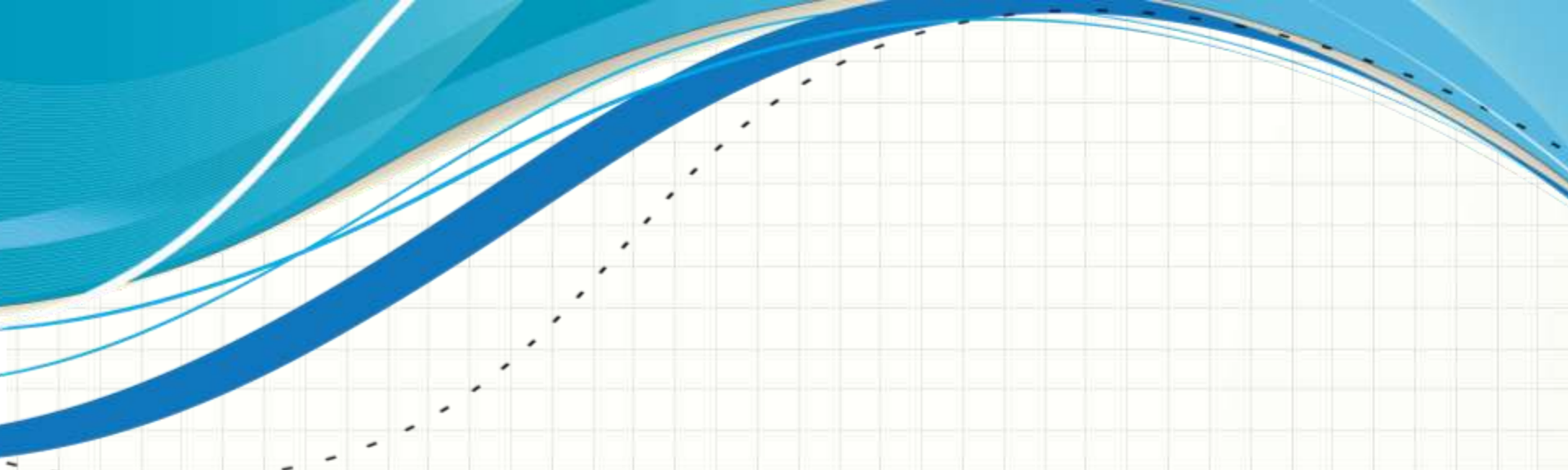
Sissejuhatus Javascripti

Javascripti harjutused

- PHP-s on üle 700 funktsiooni
- Programmid jaotatakse loogilisteks osadeks isekirjeldatud funktsioonide abil
- Funktsioonid kirjeldatakse vastavalt Näitele
- Funktsioonil võivad (aga ei pea) olema sisendparameetrid
- Funktsioonid võivad (aga ei pea) tagastama väärtuse

```
Function funktsiooniNimi(par1, par2, ...)  
{ täidetavad koodiväljendid  
}
```

```
<?php  
function kirjutaNimi()  
{  
    echo "Eefraim Kolmeteistkümnes";  
}  
function liida($x,$y)  
{  
    $summa=$x+$y;  
    return $summa;  
}  
  
echo "Minu nimi on ";  
kirjutaNimi();  
echo "<br>4 + 20 võrdub: " . liida(4,20);  
?>
```

Mis toimub Wordpressis „kapoti all“?

Failid

- Failid ilma milleta ei saa
 - Style.css - stiilifail
 - index.php - üldine fail, mis kirjeldab üldise lehtede näitamise loogika
 - header.php - Nii HTML-lehe <head> - osa kirjeldus kui ka veebilehe nähtava päise kirjeldamine
 - page.php - Lehe näitamise loogika
 - footer.php - Lehe jalus
- Wordpress defineerib reeglid, milliseid faile ja millises järjekorras kasutatakse lehtede, postituste, kategooriate, esilehe jm näitamiseks.
- Need reeglid on kirjeldatud siin:
http://codex.wordpress.org/Template_Hierarchy



Head katsetamist!

www.nort.ee

**N**.O.R.T.
Õppekeskus