

Algus

Vahepealsetest keerdkäikudest hoolimata paistab, et veebi koostamisel langeb põhirõhk lähiaastatel ikka HTMLi, sinna juurde kuuluva CSSi ja Javaskripti peale. HTMLi kirjutamise ning sealtkaudu saadavate andmetega saab PHP igati mugavasti hakkama. Seetõttu alustamegi õppematerjali lihtsa veebilehe loomisega ning vaikselt lisame sinna programmeerimisvõimalusi.

Tutvustav veebileht

HTML-faili tüübiks on tekstifail. See tähendab, et see fail võib sisaldada ainult teksti või muid ekraanil nähtavaid sümboleid. Tavalise tekstina olev osa paistab jutuna kasutajatele. Teksti osade eristamiseks aga saab teksti sisse panna `<` ja `>` märkide vahele HTMLi märgendeid, millega veebilehitsejat juhatada ja dokumendi struktuuri märkida.

Veebilehe esmaseks struktuuriks on deklaratsioon (`doctype`), HTML-dokument ise - mis jääb märgendite `<html>` ja `</html>` vahele. Ning dokument jaguneb päiseks (`head`) ning sisuosaks (`body`). Esimeses neist on andmed dokumendi kohta - kes tegi, millal tegi, kui kaua kehtib, märksõnad, lühikirjeldus jm. Sisuosas (`body`) olev tekst on üldiselt vaatajale nähtav - välja arvatud märgendite nimed, millega teksti juhitakse.

```
<!doctype html>
<html>
  <head>

  </head>
  <body>

  </body>
</html>
```

Head-osas paiknev `title` on näha veebilehitseja tiitliribal. Kui korraga avatud palju lehti, siis selle järgi saab vaadata, millist lehte parajasti näha soovitakse.

Sisus võib teksti jagada lõikudesse `div`-märgendite abil. Reavahetust tähistab `
`. Pealkirja märkimiseks sobivad `<h1>` koos lõpuga `</h1>` ning `<h2>` ja `</h2>`. Sügavama struktuuri puhul ka suuremate numbritega pealkirjad.

Loetelu piiritleb `` ja `` (`unordered list`). Iga loeteluelemendi ümber tuleb `` ja ``. Selliste vahenditega saabki lihtsa veebilehe kokku.

```
<!doctype html>
<html>
  <head>
    <title>Jaagupi leht</title>
  </head>
  <body>
    <h1>Jaagup</h1>
    <div>
      Sündis 03. mail 1976
      Tallinnas. <br /> Kasvas Mustamäel.
    </div>
    <h2>Huvialad</h2>
    <ul>
      <li>Kandlemäng</li>
```

```
<li>Jalgrattasõit</li>
</ul>
</body>
</html>
```

Veebileht nähtuna brauseriaknas:



PHP algus

Eelnev leht saadeti serverist veebilehitsejasse samal kujul nagu ta failina oli salvestatud. Serveripoolse rakenduse korral aga pannakse failis olev kood enne käima ning siis saadetakse töö tulemus brauserisse. PHP puhul tuleb käivitata kood panna algustähise `<?php` ning lõpu `?>` vahele. Käsklus `echo` trüüb tulemuse ekraanile

```
Arvutus:
<?php
    echo 3+2;
?>
```

Väljund lehel

Arvutus: 5

Ülesandeid

- * Hangi või tee selgeks enesele võimalus PHP-võimelises veebiserveris veebilehtede loomiseks. Windows-masinas sobib näiteks XAMPP või WAMP-nimeline komplekt.
- * Koosta tervitav leht ja vaata seda veebiserveri kaudu
- * Muuda lehe sisu ning uuendusnupu vajutuse järel veendu muutuse kajastumises ka veebilehitsejas.
- * Käivita konspektis olnud näide kahe arvu liitmise kohta.
- * Muuda arve ja tehet, kontrolli tulemusi.

Muutuja, tingimuslause

Programmeerimiskeeltes on levinud võimalus andmeid muutuja ehk märksõna alla meelde jätta. PHPs algavad muutujate nimed dollarimärgiga. See võimaldab neid hiljem vabamalt teksti sisse

panna. Lõik

```
$eesnimi="Juku";  
echo "Tere, $eesnimi!";
```

trükitab aimatavalt välja "Tere, Juku". Samuti siis

```
$eesnimi="Juku";  
echo "$eesnimi tuli hommikul kooli.<br />";
```

teatab, kes hommikul kooli tuli. Reavahetus
 lause lõpus hoolitseb, et järgnevat juttu alataks järgmiselt realt.

Valiku jaoks on käsklus if. Tingimus pannakse ümarsulgude sisse. Kui tingimus vastab tõele (praegusel juhul kinganumber on väiksem kui nelikümmend), siis täidetakse järgnevate looksulgude vahele paigutatud plokk. Sama lugu ka alumise tingimusega.

```
<?php  
$eesnimi="Juku";  
echo "$eesnimi tuli hommikul kooli.<br />";  
  
$kinganumber=37;  
if($kinganumber<40){  
    echo "$eesnimi saab veel lasteosakonnast kingi<br />";  
}  
if($kinganumber>45){  
    echo "Kodanik $eesnimi kingad sobivad naelakastideks.";  
}  
?>
```

Väljund ekraanil:

Juku tuli hommikul kooli.
Juku saab veel lasteosakonnast kingi

Ülesandeid

- Lisa muutuja perekonnanime tarbeks. Anna sinna väärtus ja kasuta seda lausetes.
- Koosta eraldi leht, kus kirjas temperatuur. Väljasta, kas vesi on sellel temperatuuril vedel või muutub jääks.

Kellaeg

Igal lehe avamisel näitamine on kindel kontroll selle kohta, et veebiserveris lehe loomiseks midagi toimub. Kuna samast serverist võidakse lehti vaadata mitmelt poolt, siis on tarvilik teada anda, millise ajavööndi kellaega soovitakse. Edasi juba käsklus date annab soovitud tulemuse. Sulgudes olevad tähed näitavad, kuidas kellaega vormistada. Tekst ikka jutumärkide vahel. Suur H palub tunde näidata 24-tunni süsteemis, i tähendab minuteid ning s sekundeid. Koolonid seal vahel trükitakse lihtsalt välja.

```
<!doctype html>  
<html>  
  <head>  
    <title>Kellaleht</title>
```

```
</head>
<body>
  <h1>Kellaeg</h1>
  <?php
    date_default_timezone_set("Europe/Tallinn");
    echo date("H:i:s");
  ?>
</body>
</html>
```

Väljund:

18:31:39

Ülesandeid

- Näita kellaajast vaid tunde ja minuteid
- Näita välja kuupäev. Y tähistab aastat, m kuud ning d päeva.

Tollikalkulaator

Nagu arvuti sõnastki välja lugeda võib, on arvutamine selle masina juures tähtis ülesanne. Vahel saab arvutada olemasolevate andmete põhjal. Küllalt sageli aga on vaja kasutajalt algandmete määramiseks sisestust. Andmete sisestamiseks on veebilehe jaoks olemas vorm ehk sisestuselementide komplekt. Form-elementi action-atribuudina määratakse, kuhu aadressile andmed töötlemiseks saadetakse. Kui sihtkohaks on küsimärk, siis tuleb sisestus samale aadressile, kus vormgi avanes.

```
<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
  </head>
  <body>
    <h1>Tollikalkulaator</h1>
    <form action="?">
      Monitori diagonaal tollides:
      <input type="text" name="tollid" />
      <input type="submit" value="OK" />
    </form>
  </body>
</html>
```

Tollikalkulaator

Monitori diagonaal tollides:

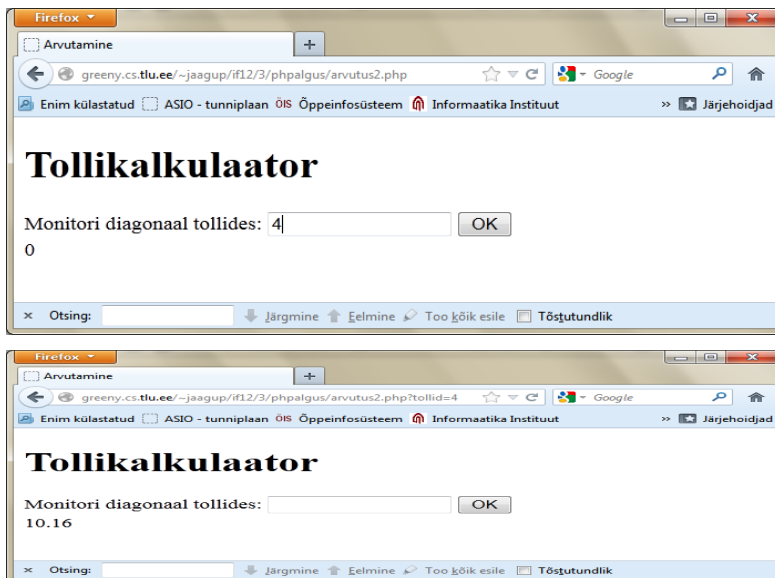
Andmete kasutamiseks tuleb nad kinni püüda. Muutuja \$_REQUEST kaudu saab vormi sisestatud väärtused kätte.

```
echo $_REQUEST["tollid"]*2.54;
```

korrutab saanud tollide arvu 2,54ga ning väljastab tulemuse sentimeetrites.

```
<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
  </head>
  <body>
    <h1>Tollikalkulaator</h1>
    <form action="??">
      Monitori diagonaal tollides:
      <input type="text" name="tollid" />
      <input type="submit" value="OK" />
    </form>
    <?php
      echo $_REQUEST["tollid"]*2.54;
    ?>
  </body>
</html>
```

Tulemus näha pildina:



Eelmine näide oli küll võimalikult lihtne, aga mõnigate puudustega. Lehe avamisel ilmub sinna salapärase ümmargune null. Vastavalt serveri seadetele raskemal juhul isegi veateade andmete puudumise kohta. Seega on hea leht viisakamaks teha. Aitab järgmine lõik:

```
if(empty($_REQUEST["tollid"])){
    echo "Ootan sisestust.";
} else {
    echo $_REQUEST["tollid"]." tolli on ".
        ($_REQUEST["tollid"]*2.54)." cm.";
}
```

Inimkeelne tõlge. Kui saabu vaid tolle pole, siis trükitakse, et "Ootan sisestust.". Muul juhul trükitakse, et mitu tolli on mitu sentimeetrit. Kusjuures empty on selline käsklus, mis loeb tühjaks nii parameetri täieliku puudumise (näiteks lehe esmakordsel avamisel, kus keegi ei teagi tolle sisestada) kui lihtsalt tühjaks jäänud teksti (mis juhtub siis, kui vajutada OK-nupule ilma tollide arvu sisestamata). Leht tervikuna.

```
<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
  </head>
  <body>
    <h1>Tollikalkulaator</h1>
    <form action="?">
      Monitori diagonaal tollides:
      <input type="text" name="tollid" />
      <input type="submit" value="OK" />
    </form>
    <?php
      if(empty($_REQUEST["tollid"])){
        echo "Ootan sisestust.";
      } else {
        echo $_REQUEST["tollid"]." tolli on ".
          ($_REQUEST["tollid"]*2.54)." cm.";
      }
    ?>
  </body>
</html>
```

Vastuses siis kõigepealt oodatakse sisestust.

Tollikalkulaator

Monitori diagonaal tollides:

Ootan sisestust.

Hiljem antakse viisakas vastus.

Tollikalkulaator

Monitori diagonaal tollides:

4 tolli on 10.16 cm.

Ülesandeid

- Koosta kalkulaator sentimeetritest tollide arvutamiseks. Jagamismärgiks on kaldkriips /.
- Koosta kalkulaator, kus sisestus on meetrites. Väljastatakse, mitu kilomeetrit see on, mitu detsimeetrit, mitu sentimeetrit ning mitu millimeetrit see on.

Ostusumma arvutamine

Sisestatavaid väärtusi võib olla mitu. Nende kättesaamiseks tuleb nad panna eri nimedega tekstiväljadesse. Järgnevas näites on ühe välja nimeks `hind` ja teise nimeks `kogus`. Neid kahte korrutades saadakse kokku ostusumma.

```

<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
  </head>
  <body>
    <h1>Summa kalkulaator</h1>
    <form action="??">
Kauba tükihind:
      <input type="text" name="hind" />
Ostetav kogus:
      <input type="text" name="kogus" />
      <input type="submit" value="OK" />
    </form>
    <?php
      if(empty($_REQUEST["hind"]) or empty($_REQUEST["kogus"])){
        echo "Ootan sisestust.";
      } else {
        echo $_REQUEST["hind"]*$_REQUEST["kogus"];
      }
    ?>
  </body>
</html>

```

Summa kalkulaator

Kauba tükihind:
Ootan sisestust.

Ostetav kogus:

Summa kalkulaator

Kauba tükihind: Ostetav kogus:

Valik rippmenüüst

Tekstiväli on sisestuseks mugav teha. Kuna aga kasutaja pääseb sinna kõike sisestama, siis võib kergesti sisse sattuda ka sobimatut teksti. Kasutajale piiratud arvu valikute andmiseks sobib rippmenüü. Rippmenüü loob element tüübist select. Elemendi nime järgi saab endiselt PHP kaudu kasutaja sisestuse kätte.

```

<select name="hind">
  <option value="">Vali kaup ...</option>
  <option value="0.70">Leib</option>
  <option value="0.60">Sai</option>
  <option value="0.50">Piim</option>
</select>

```

Atribuudi value juurde kirjutatud väärtus läheb serverisse edasiseks töötamiseks, tekst enne option- elemendi lõppu jääb näha kasutajale oma valiku tegemisel. Allpool saab arvutada endist moodi.

```

<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
  </head>
  <body>
    <h1>Summa kalkulaator</h1>
    <form action="??">
Kaup:
      <select name="hind">

```

```

        <option value="">Vali kaup ...</option>
        <option value="0.70">Leib</option>
        <option value="0.60">Sai</option>
        <option value="0.50">Piim</option>
    </select>
    Ostetav kogus:
        <input type="text" name="kogus" />
        <input type="submit" value="OK" />
</form>
<?php
    if(empty($_REQUEST["hind"]) or empty($_REQUEST["kogus"])){
        echo "Ootan sisestust.";
    } else {
        echo $_REQUEST["hind"]*$_REQUEST["kogus"];
    }
?>
</body>
</html>

```

Tulemus lehel:

Summa kalkulaator

Kaup: Ostetav kogus:

Ootan

- Leib
- Sai
- Piim

Ülesandeid

- Pane näide tööle
- Muuda tooteid ja hindu, katseta tulemust.
- Loo toodetest kaks rippmenüüd eri nimedega. Näita, kui suur summa tuleb kummagi toote kohta eraldi ning kui palju kahe toote peale kokku.

Andmed massiivist

Paar harva muutuvat väärtust on hea veebilehe sisse kirjutada. Kui aga andmeid rohkem, neid kasutatakse mitmes kohas või nad kipuvad sageli muutuma, siis on tavaline, et andmete kirjapaneku ja kasutamise kohad erinevad. Andmed on mugav kirja panna eraldi andmebaasi või eraldi faili. Et seda pole veel õpitud, siis piirdume ühise massiiviga, mida ka vajadusel mitmel pool kasutada saab.

Massiivi loomiseks sobib käsklus `array()`. Edasi võib sinna ükshaaval andmed sisse panna. PHP lubab massiivi võtmena kasutada ka teksti. Kui kirjutatakse `$kaubad["vorst"]="2.50"`, siis muutujaks on `$kaubad`, võtmeks "vorst" ning väärtuseks "2.50".

```

$kaubad=array();
$kaubad["vorst"]="2.50";
$kaubad["juust"]="3.00";
$kaubad["kartul"]="0.45";

```

Pärast võimalik tsükliga andmed läbi käia. Käsk `foreach` võtab ükshaaval ette kõik võtme ja väärtuse paarid (mis siinsel juhul on `$nimetus` ja `$hind`) ning lubab nendega tsükli keha sees (ehk

looksulgude vahel) toimetada. Tulemusena trükitakse välja kõik valikud nõnda, et nimetused jäävad kasutajale silma ette valida. Hinnad aga saadetakse pärast valiku tegemist ning sisestusnupule vajutamist serverisse.

```
foreach($kaubad as $nimetus => $hind){
    echo "<option value='$hind'$nimetus</option>";
}
```

Serveris tulemuse kokku arvutamine käib ikka endisel moel.

```
<?php
    $kaubad=array();
    $kaubad["vorst"]="2.50";
    $kaubad["juust"]="3.00";
    $kaubad["kartul"]="0.45";
?>
<!doctype html>
<html>
    <head>
        <title>Arvutamine</title>
    </head>
    <body>
        <h1>Summa kalkulaator</h1>
        <form action="?">
            Kaup:
            <select name="hind">
                <option value="">Vali kaup ...</option>
                <?php
                    foreach($kaubad as $nimetus => $hind){
                        echo "<option value='$hind'$nimetus</option>";
                    }
                ?>
            </select>
            Ostetav kogus:
            <select name="kogus">
                <option>Vali kogus</option>
                <option>1</option>
                <option>2</option>
                <option>3</option>
                <option>4</option>
                <option>5</option>
            </select>
            <input type="submit" value="OK" />
        </form>
        <?php
            if(empty($_REQUEST["hind"]) or empty($_REQUEST["kogus"])){
                echo "Ootan sisestust.";
            } else {
                echo $_REQUEST["hind"]*$_REQUEST["kogus"];
            }
        ?>
    </body>
</html>
```

Summa kalkulaator

Kaup: juust ▾ Ostetav kogus: Vali kogus ▾ OK

Ootan sisestust.

Vali kogus
1
2
3
4
5

Ülesandeid

- Pane näide tööle, muuda andmeid, kontrolli tulemust.
- Pane lehele kaks valikupaari kaupade ja koguste tarbeks. Näita kummagi paari tarbeks summa eraldi ning lõppu kogusumma.
- Koosta massiiv, mille võtmeteks on liini bussipeatuste kaugused algpeatusest ning väärtusteks vastavate bussipeatuste nimed. Koosta samade andmetega kaks eri nimedega rippmenüüd. Kasutaja valib kaks peatust, talle teatatakse, kui suur on nende peatuste vahe kilomeetrites.
- Lisaks eelmisele on kolmandas rippmenüüs valik, kas tegemist on tava-, kiir- või ekspressliiniga. Peidetud väärtustena kuuluvad sinna juurde kilomeetrihinnad. Arvestatakse kokku sõidu maksumus.