

Algus

Veebilehestikke on loodud 1990ndatest alates. Peale andmete näitamise/vaatamise on veebis ka üsna algusest peale nendega midagi ka teha saanud. Siinses õppematerjalis püüamegi veebilehti elama panna, võimaldades kasutajal nende taga toimetada. Keskendume kliendi arvutis töötavatele Javaskripti lahendustele, mida saab kohe looma hakata - pole vaja käivitusõigust serveris ega ligipääsu andmebaasile kui keerukamaid ja riskiohtlikumaid võimalusi. Küllalt palju kasulikku ja rõõmsat saab ka kasutaja arvutis tööle panna.

Tekstilehekülg

Internetis saab vaadata kõike, mida sinna sisse pannakse ja mida vaataja lugeda suudab. Olgu see üksik pilt, helilõik või ka lihtsalt tekst. Teksti veebis kättesaadavaks tegemiseks tuleb see paigutada kausta, kus teksti lugeda saab. Lihtsamaks katsetamiseks piisab ka kohaliku masina kataloogist, kust siis hiljem sobiv fail avada.

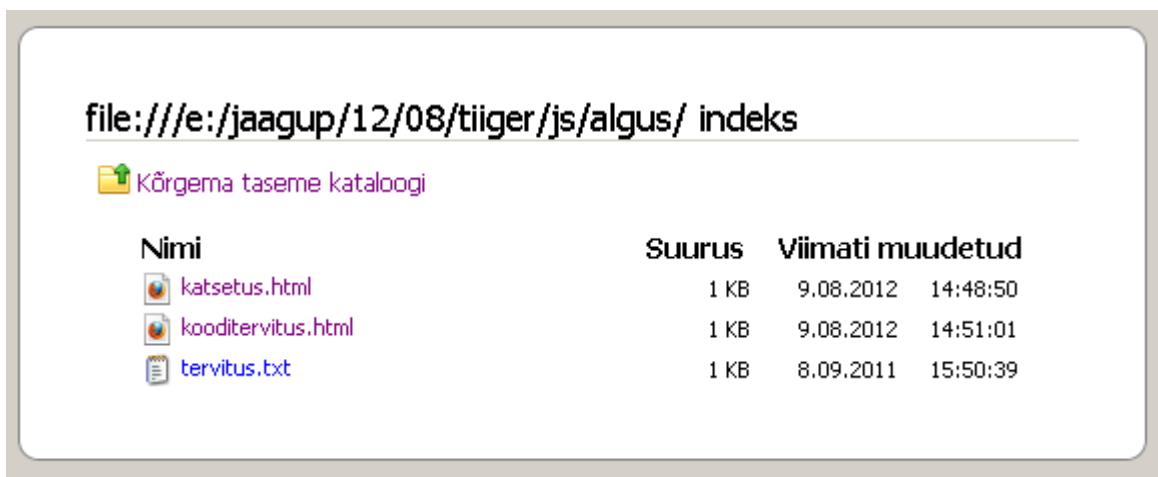
Näites loodi fail nimega

tervitus.txt

sisuga

tere

Veebilehitsejas kausta vaadates võib näha, et fail on olemas.



Faili avamisel tuleb sisu nähtavale



Ongi esimene tervitav teave edasi antud.

Millega tekstifaile luua - see sõltub juba kasutatavast arvutist ja operatsioonisüsteemist. Linuxi all kirjutajatel on arvatavasti tuttavad redaktorid nimedega vi, joe või pico. Windowsi-masinate kuulub standardkomplekti tarvikute all olev Notepad. Mõnevõrra koodikirjutamist hõlbustab ning treppida aitab eraldi lisatav Notepad++. Aga üks vastavalt maitsele leiab kirjutusprogramme piisavalt. Peab lihtsalt vaatama, et tulemuseks oleks puhtalt salvestatav tekst - sellisena ka edaspidiseid veebilehti kindlam teha.

HTML-leht

Palja tekstiga saab edasi anda jutu sisu. Tühjade ridade ning taandridadega saab väljanägemist parasjagu sättida. Kirjutusmasina-ajastul nendest moodustest täiesti piisas. Nüüd aga on inimesed harjunud vaatama mitmekülgsemat lehekülge nii väljanägemise kui võimaluste poole pealt. Aastaid on veebiga koos käinud HTMLi-nimeline keel, mis koos oma tõusude ja mõõnadega on tänapäevalgi valitsevaks jäänud jõudes nüüd oma viienda versioonini. Järgnevalt üks koodinäide, kus HTMLi abil leht kokku pandud. Mõned seletused sinna juurde.

Esimene rida

```
<!DOCTYPE html>
```

annab veebilehitsejale teada, et millist tüüpi dokumenti oodatakse.

Leht ise jaguneb kaheks suureks osaks: head ning body. Esimeses neist on andmed lehe kohta - hiljem samuti ettevalmistatud koodilõigud. Teise tuleb lehe sisu ise. Andmeks lehe kohta on ka title ehk pealkiri. Edasi soovitavalt kasutatav kooditabel. Täpitähtedega murede vältimiseks oleks see hea sättida utf-8 peale (ning ühtlasi hoolitseda, et ka oma redaktor selles kodeeringus tähed väljastaks). Notepad++ juures kohe eraldi selle jaoks menüü olemas.

Teksti kujundamise jaoks on loodud omaette astmeliste laadilehtede keel nimega CSS. Siin näites antakse teada, et kõik ülemise/esimese taseme pealkirjad joondatakse keskele (`text-align: center`). Kel aga on põhjust kujundusega rohkem tegelda, võib lisateavet hankida veebist nt. <http://www.w3schools.com/> õpetuste hulgast. Või siis Tiigrihüppe abil eraldi valmivast veebikujunduse õppematerjalist.

```
<html>
  <head>
    <title>Kujunduse katsetus</title>
    <meta charset="UTF-8" />
    <style type="text/css">
      h1{
        text-align: center;
      }
    </style>
  </head>
```

Sisupoole andmed on lehel üldiselt näha. Kõigepealt siis pealkiri, mis viisakasti `<h1>` ja `</h1>` märgendite vahele pandud. Märgend `h1` tähendab esimese taseme ehk suurimat pealkirja (heading 1). Näitamise mõttes loodud loetelu (`` ehk unordered list), mille sees siis üksikud punktid ``-käsuga (list item). Ning üldjuhul kui mõni märgend algab, siis peab see ka kusagil lõppema. Nii on pealkirja sabas kaldkriipsuga `</h1>` ning loetelu lõpus ``.

Et HTML on tavaline tekstivorming, siis tuleb kõik lisaandmed seal sees tavalise tekstina edasi anda. Nii ka viide teisele lehele on lihtsalt käsklus nimega `a` (anchor), millel atribuudina küljes `href`

(hyperlink reference) koos aadressiga, kuhu veebilehitseja siis vastava teksti vajutamisel suunatakse.

```
<body>
  <h1>Katsetuste leht</h1>
  <ul>
    <li>Kooli röömus algus</li>
    <li>Veebi <a href="http://www.neti.ee/">otsinguleht</a></li>
    <li>HTMLi <a href="http://validator.w3.org/">validaator</a></li>
  </ul>
</body>
</html>
```

Ning selgema pildi saamiseks lehe kood tervikuna.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Kujunduse katsetus</title>
    <meta charset="UTF-8" />
    <style type="text/css">
      h1{
        text-align: center;
      }
    </style>
  </head>
  <body>
    <h1>Katsetuste leht</h1>
    <ul>
      <li>Kooli röömus algus</li>
      <li>Veebi <a href="http://www.neti.ee/">otsinguleht</a></li>
      <li>HTMLi <a href="http://validator.w3.org/">validaator</a></li>
    </ul>
  </body>
</html>
```

Välja näeb viisakalt ja lihtsalt vormindatud lehena.

Katsetuste leht

- Kooli röömus algus
- Veebi [otsinguleht](http://www.neti.ee/)
- HTMLi [validaator](http://validator.w3.org/)

Viimane validaatori viide on siia juurde täiesti mõttega pandud. Kel lehele esimene pilt ette saadud, võiks kontrollida, et leht ikka igapidi viisakalt kokku pandud on ning vajalikud märgendid oma kohtadel. Muidu võib tulemusena ootamatuid üllatusi juhtuma hakata. Kontroll küllaltki lihtne. Kui enamik korras, näidatakse rohelist vastuslehte. Kui aga märgatavaid muresid süntaksi poole pealt, siis tuleb vastus punane ning tasub hakata ridade kaupa veateateid ja soovitusi lugema. Mõnigikord ei pruugi viga olla sama mis näidatakse, kuid enamjaolt on osutataval real või paar rida kõrgemal midagi viltu läinud, mida tasub siis kohendada.

Ülesandeid

- Tee näited läbi, veendu nende avanemises veebilehitsejas.
- Koosta veebileht enese tutvustamiseks - kus ja millal sündinud, kus koolis käinud, millistest keeltest (kasvõi natuke) aru saad, mis oskustega saad sõpradele kasulik olla.
- Usinamatele: Tee failid võimaluse korral kättesaadavaks mõnes avalikus veebiserveris (nt. zone.ee). Vaata neid ise ning lase oma sõbral vaadata. Muuda sisu natuke ning vaata uuesti tulemust.

Kasutajat tervitav leht

Eelnev leht näeb välja ikka samasugune. Kui tahta lehe sisu muuta, siis tuleb lehe loojal sinna uus jutt kirjutada ning serverisse vaatajatele üles panna.

Lehele kirjutatud programmikood võimaldab aga kasutajale reageerida, lihtsaimal juhul ta nime küsida ning selle järgi tervitada. Selline suhtlus vajab natuke sättemist ette.

Levinud sisestuselemendiks on HTMLi märgend nimega input. Teksti jaoks on ta tüübiks text. Programmikoodi abil elemndi poole pöördumiseks tasub temale määrata ka id. Nõnda siis sobib veebilehe koodi sisse rida

```
<input type="text" id="eesnimi" />
```

HTML-fail kokku näeks välja siis järgmine:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Andmete sisestus</title>
  </head>
  <body>
    <h1>Katsetuste leht</h1>
    Palun eesnimi:
    <input type="text" id="eesnimi" />
  </body>
</html>
```

ja lehele tekib sisestuskast.

Katsetuste leht

Palun eesnimi:

Tahtes lehel midagi juhtuma panna, tuleb arvutile teada anda, mille peale midagi toimuma hakkab. Toimetust saab mugavalt käivitada nupu abil, selle jaoks ka sobiv rida lehele.

```
<input type="button" value="OK" />
```

Kuvamiseks sobib lehele div-kiht id-ga vastus.

```
<div id="vastus">
  Vastuse koht.
</div>
```

Kõige tähtsamas lõigus tuleb veebilehitsejale seletada, et mida nupuvajutuse peale tuleb tegema hakata. Veebilehele programmikoodi lisamiseks sobib head-ossa skriptiplokk

```
<script>

</script>
```

Sinna sisse saab siis hakata arvuti jaoks käsklusi kirjutama. Käsklusi ootava lehe blankett näeb välja järgmine:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Andmete sisestus</title>
    <script>

    </script>
  </head>
  <body>
    <h1>Katsetuste leht</h1>
    <div>
      Palun eesnimi:
      <input type="text" id="eesnimi" />
      <input type="button" value="OK" />
    </div>
    <div id="vastus">
      Vastuse koht.
    </div>
  </body>
</html>
```

Soovime, et tervitatakse inimest, kelle nimi tekstivälja sisse kirjutatakse. Selleks tuleb eesnimi tekstiväljast välja võtta, juurde lisada tervitussõna ning kogu lugu kasutajale kuvada.

Veebilehe elemendile, millel on id-atribuut, saab ligipääsu küsida Javaskripti käsklusega

```
document.getElementById
```

Elementide sisu küsimiseks ja muutmiseks on neil omadus innerHTML. Kui soovin vastusplokki kirjutada lihtsalt Tere, siis see näeb välja

```
document.getElementById("vastus").innerHTML="Tere";
```

Paljas käsklus aga niisama iseenesest ei tea käivituda. Selleks tuleb käsklus panna nimega funktsiooni sisse ning nupuvajutuse peale see funktsioon tööle panna.

```
function tervita(){
```

```
        document.getElementById("vastus").innerHTML="Tere";
    }
}
```

Näites siis tervita on funktsiooni nimi. Ümarsulud kuuluvad funktsiooni nime juurde. Ning järgnevate loogeliste sulgude vahele pannakse käsud, mis funktsiooni käivitamisel tööle pannakse. Nupuvajutuse peale funktsiooni käivitamiseks tuleb nupu onclick-atribuudis öelda, et milline funktsioon käima läheb. Sedakorda siis ainuke mis meil on ehk funktsioon nimega tervita.

```
<input type="button" value="OK" onclick="tervita();" />
```

Pärast täiendusi näeb kool välja järgmine

```
<!DOCTYPE html>
<html>
  <head>
    <title>Programmikoodi katsetus</title>
    <script>
      function tervita(){
        document.getElementById("vastus").innerHTML="Tere";
      }
    </script>
  </head>
  <body>
    <h1>Katsetuste leht</h1>
    <div>
      Palun eesnimi:
      <input type="text" id="eesnimi" />
      <input type="button" value="OK" onclick="tervita()" />
    </div>
    <div id="vastus">
      Vastuse koht
    </div>
  </body>
</html>
```

Enne nupuvajutust paistab vastuse kihil olema "Vastuse koht", pärast programmikoodi abil pandud "Tere".

Katsetuste leht

Palun eesnimi:

Vastuse koht

Katsetuste leht

Palun eesnimi:

Tere

Nõnda siis

```
document.getElementById("vastus").innerHTML=
  "Tere, "+document.getElementById("eesnimi").value;
```

tulemuseks on, et kihi "vastus" sisuks (innerHTML) saab tervitus inimesele, kelle eesnimi tekstivälja sisestati.

Katsetuste leht

Palun sisesta oma eesnimi:

Vastuse koht.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Programmikoodi katsetus</title>
    <meta charset="UTF-8" />
    <script type="text/javascript">
      function tervita(){
        document.getElementById("vastus").innerHTML=
          "Tere, "+document.getElementById("eesnimi").value;
      }
    </script>
  </head>
  <body>
    <h1>Katsetuste leht</h1>
    Palun sisesta oma eesnimi:
    <input type="text" id="eesnimi" />
    <input type="button" value="OK" onclick="tervita();" />
    <div id="vastus">
      Vastuse koht.
    </div>
  </body>
</html>
```

Katsetuse tulemus näha ka ekraanil.

Katsetuste leht

Palun sisesta oma eesnimi:

Tere, Jaagup

Ülesanne

* Loo ekraanile kaks tekstivälja: üks eesnime, teine perekonnanime jaoks. Tervita kasutajat mõlemat nime pidi.

Tollid sentimeetriteks

Arvuti nimigi räägib, et masin võiks mõista arvutada. Tänapäeva programmeerimiskeelte puhul

tuleb vahel enne otsida kohta, kuidas masin arvutama panna, aga üldjuhul on see täiesti võimalik.

Järgnev lihtne näide arvutab tolle sentimeetriteks. Kordajaks 2,54 - ühele tollile vastab nõnda palju sentimeetreid. See tähendab, et toll on pikem ning sama pikkuse peale on tolle vähem. Programmeerimiskeeltes kasutatakse koma asemel punkti - see tava Inglise/Ameerika poolt tulnud. Kui kasutajaliideses mõnikord säetakse väljanägemine kasutaja asukohale vastavaks, siis programmeerimiskeele sees on ta ikka ühtmoodi.

Andmeid saab kasutajalt ikka samuti kätte. Tema sisestab numbrid tekstivälja. Tekstivälja küljes on id, mille järgi saab välja ja selle sees oleva väärtuse küsida. ning edasi võib teha juba sobivaid tehteid. Avaldis

```
document.getElementById("tollidearv").value*2.54+" cm";
```

teatab, et võetakse tekstikasti nimega "tollidearv" väärtus ja korrutatakse see 2.45-ga. Juurde liidetakse veel tühik ja cm, mis tähtedena kolmekesi jutumärkide vahel. Nõnda kasutajal pärast ekraanilt parem lugeda, et mida väljastatud vastus tähendab.

```
<!doctype html>
<html>
  <head>
    <title>Arvutamine</title>
    <meta charset="utf-8" />
    <script>
      function arvuta(){
        document.getElementById("vastus").innerHTML=
          document.getElementById("tollidearv").value*2.54+" cm";
      }
    </script>
  </head>
  <body>
    <h1>Arvutamine</h1>
    <div>
      Mitu tolli:
      <input type="text" id="tollidearv" />
      <input type="button" value="Sisesta" onclick="arvuta()" />
    </div>
    <div id="vastus">

  </div>
</body>
</html>
```

Tulemus ekraanil:

Arvutamine

Mitu tolli:

10.16 cm

Ülesandeid

- * Loo leht, millega arvutatakse sentimeetreid tollideks. Tolle on sentimeetritest 2.54 korda vähem.
- * Koosta uuele lehele kalkulaator, kus poehinna puhul näidatakse, kui palju sellest moodustab käibemaks (20%). Vihe: kui hind letil on 1 euro ja 20 senti, siis 1 euro saab kaupmees ning käibemaksuks läheb 20 senti (20% suuruse maksu puhul).
- * Paiguta lehele kaks tekstivälja. Ühte kirjutatakse toote hind, teise ostetav kogus. Väljastatakse vajalik summa.

Kolme arvu aritmeetiline keskmine

Näitena juurde veidi pikema arvutusvalemiga kalkulaator.

Hinna või tollide arvu leidmiseks tekstiväljast saadud väärtusi korrutades saab Javaskript ise aru, et neid tuleb arvudena käsitleda. Tähtede omavahelisest korrutamise pole üldiselt erilist kasu ning seetõttu korrutamistehte peale muudetakse enne nad sisestatavast tekstist arvutatavateks arvudeks ning korrutamise peale saadakse mõistlik vastus kätte. Kui aga liitmiskäsuna kirjas plussmärk, siis ei tea arvuti, kas ta peaks panema tähti üksteise otsa või liitma leitud arvulisi väärtusi. Ehk siis kas $3+2$ on 5 või hoopis 32. Javaskripti puhul valitakse tavajuhul kusjuures viimane vastus. Et arvuti teaks neid arvudena liita ja arvestada, et kolm pluss kaks on viis, selleks aitab käsklus `parseInt`.

```
var a1=parseInt(document.getElementById("arv1").value);
```

teatab, et tuleb võtta tekstivälja arv1 väärtus ning see `parseInt` käskluse abil mälus täisarvuks teisendada. Tulemus talletatakse muutuja ehk märksõna `a1` alla. Sõna `var` seal ees näitab, et tegemist uue muutujaga. Nõnda saab ka teised väärtused tekstiväljadest arvuna kätte ning pärast seda võib valemitega toimetama hakata ja lõpuks tulemuse väljastada.

```
<!doctype html>
<html>
  <head>
    <title>Keskmete arvutamine</title>
    <script>
      function arvuta(){
        var a1=parseInt(document.getElementById("arv1").value);
        var a2=parseInt(document.getElementById("arv2").value);
        var a3=parseInt(document.getElementById("arv3").value);
        var aritmeetilineKeskmine=(a1+a2+a3)/3;
        var tulemus="Aritmeetiline keskmine: "+aritmeetilineKeskmine;
        document.getElementById("vastus").innerHTML=tulemus;
      }
    </script>
  </head>
  <body>
    <h1>Keskmesed</h1>
    Arvud: <br />
    <input type="text" id="arv1" />
    <input type="text" id="arv2" />
    <input type="text" id="arv3" />
    <input type="button" value="Arvuta" onclick="arvuta()" />
    <div id="vastus">

    </div>
  </body>
</html>
```

Keskmised

Arvud:

<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="2"/>	<input type="button" value="Arvuta"/>
--------------------------------	--------------------------------	--------------------------------	---------------------------------------

Aritmeetiline keskmine: 4

Kalkulaatorite ideid

Üheks Javaskripti levinud rakenduseks on mitmesugused veebilehel töötavad kalkulaatorid. Kui valem on teada, siis pole muud kui andmed sisse ja vastus käes. Et selline oskus edaspidi libedalt läheks, tasub ise vähemalt üks arvutusvalem välja mõelda või üles otsida ning selle kohta kalkulaator teha. Kui omal paremaid mõtteid ei tule, siis mõne idee saab julgesti siit.

- Bensiinimüük
- Kehamassi indeks
- Mõõõtühikud - hulgateisendus
- Vahemaa läbimiseks kuluv aeg
- Söögi kalorsus / toitainete sisaldus
- Elektrienergia arvutused
- Keskmiste arvutamine (aritmeetiline/geomeetriline/mediaan)
- Palgakalkulaator (üldsumma, bruto, neto, sotsiaalmaks, töötuskindlustus)
- Ruutvõrrandi nullkohad (ruutjuure käsk Math.sqrt())
- Kahe muutujaga lineaarvõrrandsüsteemi lahend