

Tallinna Ülikool  
Informaatika Instituut

# HTML 5 ja CSS 3 uued võimalused ning nende töökindlus veebilehitsejates

Seminaritöö

Autor: Anton Birjukov  
Juhendaja: Jaagup Kippar

Tallinn 2012

# Sisukord

|   |    |
|---|----|
| <a href="#">Sissejuhatus</a> .....                                    | 3  |
| 1. <a href="#">HTML5-st ja CSS3-st</a> .....                          | 4  |
| 1.1. <a href="#">Lühike HTML-i Ajalugu</a> .....                      | 4  |
| 1.2. <a href="#">HTML5-e tutvustus</a> .....                          | 5  |
| 1.3. <a href="#">XHTML5-e tutvustus</a> .....                         | 6  |
| 1.4. <a href="#">CSS3-e tutvustus</a> .....                           | 6  |
| 2. <a href="#">HTML5 elementide rakendamine brauserites</a> .....     | 7  |
| 2.1. <a href="#">Brauseritest ja nende kasutamisest</a> .....         | 7  |
| 2.2. <a href="#">Üldine</a> .....                                     | 7  |
| 2.3. <a href="#">Video element</a> .....                              | 8  |
| 2.4. <a href="#">Audio element</a> .....                              | 10 |
| 2.5. <a href="#">Canvas element</a> .....                             | 12 |
| 2.6. <a href="#">SVG element</a> .....                                | 14 |
| 2.7. <a href="#">Vormid (Forms)</a> .....                             | 14 |
| <a href="#">Sisendväärtuste tüübid (Input types)</a> .....            | 14 |
| <a href="#">Vormi atribuudid (Form attributes)</a> .....              | 17 |
| 3. <a href="#">HTML5-e API-d</a> .....                                | 18 |
| 3.1. <a href="#">Veebilehe säilitamise API (Webstorage API)</a> ..... | 18 |
| <a href="#">Sessiooni säilitamine (Session Storage)</a> .....         | 18 |
| <a href="#">Kohalik säilitamine (Local Storage)</a> .....             | 18 |
| 3.2. <a href="#">Tõmba ja aset API (Drag-and-drop API)</a> .....      | 19 |
| 3.3. <a href="#">Geograafiline asukoht (Geolocation API)</a> .....    | 21 |
| 4. <a href="#">CSS3-e elemendid</a> .....                             | 22 |
| 4.1. <a href="#">CSS3 Font-face</a> .....                             | 22 |
| 4.2. <a href="#">CSS3 üleminekud (CSS3 transitions)</a> .....         | 24 |
| 5. <a href="#">Järeldus ja soovitused</a> .....                       | 25 |
| <a href="#">Kokkuvõte</a> .....                                       | 27 |
| <a href="#">Kasutatud kirjandus</a> .....                             | 28 |

## Sissejuhatus

Ilmselt on igale arvutist teadlikule inimesele tuttav hüpertekst-märgistuskeel ehk HTML (*HyperText Markup Language*). HTML on olnud arvutitega kaasas juba väga ammu ajast. See on pidevalt arenenud, kuid hetkel peab see arenema veelgi rohkem. Seda eesmärki täidabki HTML5. Veebilehtede tähenduse osakaalu kasvatab seadmete ja veebiteenuste pidev suurenemine. Inimesed veedavad juba 57% oma ajast arvuti ees, kasutades veebilehitsejat (Bechynský, 2011). Siit tuleneb ka olulisus uurida uut veebilehitsemise standardit. HTML5 on suhteliselt uus standard, mida veel arendatakse, seega on vaja pidevalt jälgida, kas seda võib kasutada juba ning millised on võimalused selle kasutamiseks. Kuna HTML on tihti mingil kujul esitatud, tehes ka keerulisemaid rakendusi, on oluline teada HTML-i võimalusi kõikidel arendajatel. Samuti uutel võimalustel, mis kaasnevad HTML5-ga ja kaskaadlaadistik kolmega (edaspidi CSS3) on suur tähtsus, kuna need hõlbustavad ja aitavad kasutada mitmeid uusi viise andmete käitlemiseks, kontrollimiseks ja valideerimiseks.

Teema valikuks kujuneski põhjus, et see on oluline teema, mida uurida lähemalt ja tutvustada teistele ka. Samuti isiklik huvi uue standardi võimaluste ja muutuste vastu.

Seminaritöö üritab leida võimalusi, mida on võimalik HTML5-s ja CSS3-s võtta hetkel kasutusele. Millised võimalused töötavad ja mis viisil need toimivad. Oluline rõhk on kaasaegsete veebilehitsejate erineval arusaamisel uutest objektidest. Samuti on ka toodud lühike tutvustus HTML-i ja CSS3-e arengust ja ajaloo. Töö ei keskendu ainult HTML5 elementidele, vaid ka selle uue standardi ühildumisele erinevate meile tuntud võimalustega, nagu näiteks SVG ja Javascript. Töös kasutatakse kaasaegseid veebilehitsejaid. Töös pole ära toodud kõiki, ääretuid laiu võimalusi, mida need keeled pakkuvad. Töös on esitletud vaid peamised ja olulisemad elemendid ning võimalused. Siin on lähtutud põhimõttest, et see on lugemiseks ja järgi tegemiseks veebistandarditest keskmiselt teadlikule inimesele.

Seminaritöö on jaotatud neljaks suureks peatükiks. Igas peatükis on esitletud katsetus, selle kood ja tulemus. Esimene peatükk tutvustab HTML-i ajalugu ja CSS3-e võimalusi. Teises peatükis on erinevad testid HTML-i enda elementidega erinevates veebilehitsejates. Kolmandas kasutatakse aga juba JavaScripti, et kasutada erinevaid HTML5-s olevaid võimalusi. Kuuendas peatükis on lühike ülevaade CSS3-e elementidest ja nende kasutamisest. Viimasena järgneb järeldus ning autori poolsed ettepanekud ja kokkuvõte.

Kasutatud näited on järgmisel URL-l: <http://www.tlu.ee/~antshki/Html5/>

# 1. HTML5-st ja CSS3-st

## 1.1. Lühike HTML-i Ajalugu

Ettekujutus hüpetekstist, millele on kogu veeb üles ehitatud, on juba rohkem kui 40 aastat vana. Juba 1945.a kirjutas elektriinsener Vannevar Bush oma artiklis „As We May Think“ masinast Memex, milles dokumendid olid omavahel ühendatud ühendustega. Ühendused olid realiseeritud mikrofilmi abil, mitte aga digitaalselt. Järgmisena kasutas 1960-ndatel D.Englebart Memex-i mudelit, et luua esimene edukas dokumente ühendav rakendus digitaalses vormis, rakendades programmi nimega NSL.

1980-ndal aastal alustas „veebi leiutaja“ Tim Berners-Lee tööd hüpertekstiga CERN-is. Ta arendas seal programmi Enquire, mis aitas tal meenutada, kes ja kus ning millise projekti kallal töötas. 1989.a pakkus ta välja interneti-põhise hüperteksti süsteemi. Hiljem liitus temaga hüperteksti arendusel Robert Cailliau.

Just Tim Berners-Lee määratles HTML-i ja kirjutas esimese WWW (*World Wide Web*) veebilehitseja ning serveri tarkvara 1990.a. detsembris, kasutades NEXTStep arvutit. Esimene avalik HTML-i kirjeldus oli dokument nimega „HTML Tags“. See selgitas kahteteist suhteliselt lihtsat elementi. HTML põhines SGML-l (standardne üldistatud märgistuskeel), mis oli rahvusvaheline programmeerimise keel alates 1986.aastast. HTML aitas korrastada ja kujundada dokumendid kasutamata selleks mingeid väga erilisi märke. HTTP protokoll transportis aga neid samu dokumente üle interneti.

HTML 1.0 oli mõeldud vaid tekstidokumentide kuvamiseks ainult teksti näitavates brauserites, mis eirasid graafilisi elemente. Alles 1996.a W3C (*World Wide Web Consortium*) kinnitas HTML 2.0-i. Teada tuntud elemendid, nagu *html*, *head*, *body*, *base*, *link*, *meta* ja *form*, lisati ametlikult HTML-i keelde alles nüüd. *Ordered list*, *inline*, *block*, *br* ja *hr* olid samuti lisatud. Samuti sai kirja pandud märke, nagu „&copy“ kujutada sümbolina „©“. Tähtsaim oli aga *img* element, millega sai graafikat kuvada brauserites Mosaic ja Cello.

HTML 3.2 ametlik soovitus ilmus 1997.a jaanuaris. Lisati tähtsad elemendid nagu *style*, *script* ja *div*. Teised, hetkel juba vananevad elemendid, olid *applet*, *center* ja *font*. Lisandus võimalus kasutada pilte hüperlinkidena. Samuti sai nüüd värve määrata RGB värvikoodide abil.

HTML 4.0 ilmus välja 1997.a detsembris. Standardis olid uued elemendid: *object*, *abbr*, *acronym*, *colgroup*, *col*, *tbody*, *tfoot* ja *thead*. Samuti sai nüüd kirjutada dokumente, mis tahes keeles, kuna lisandus *lang* atribuut. Samuti oli tähtis *event* atribuut, nagu *onload*, *onselect*, *inline*, *onkeydown*,

*onclick* jt. HTML 4.01 ilmus 1999.a detsembris, kuid suuri muutusi see endaga kaasa ei toonud, pigem selgitati lihtsalt rohkem juba kasutusel olevate atribuutide kasutamist.

Juba 2000.a jaanuaris esitles W3C kavandatud soovitus üleminekule XHTML-i peale, kuna HTML-il polnud paindlikust ja kohanemisvõimet. Selle tingis see, et keel polnud rangelt määratud ja brauseritel oli tülikas aimata, mida ühe või teise vigase lehega mõeldud on. See oli liiga piiritletud oma väheste reeglitega ning paljud HTML-i kasutavad programmid tõlgendasid seda erinevalt. XHTML pakkus aga rohkem paindlikkust, kuna see andis võimaluse XML-stiilis dokumentidest võtta kindlalt määratud vajalik info vastaval programmil ja seada see paika nii nagu programm ise tahtis.

HTML-i kasutus on laienenud. Enam ei kasutata seda ainult veebilehtede kuvamiseks, vaid ka mujal, näiteks emailide saatmisel. HTML-i asemel hakati kasutama rohkem XHTML-i, eriti pärast XHTML 1.0, XHTML 1.1 ja XHTML 2.0 ilmutumist. HTML5-t arendatakse juba koos XHTML5-ga.

## **1.2. HTML5-e tutvustus**

HTML5-e arendamine algas aastal 2004. Mitteametlik ekspertide meeskond, mis oli pärit *Apple Computer*-ist, *Mozilla Foundation*-ist ja *Opera Software*-ist lõi grupi nimega WHATWG (*Web Hypertext Application Technology Working Group*). WHATWG loodud HTML5-e spetsifikatsioon võeti lõpuks vastu W3C-e poolt aastal 2007. Tänapäeval töötavad W3C ja WHATWG partneritena HTML5-e arendamisel. HTML5-s on palju uusi elemente. Keel asendab semantiliselt ka paljusid tuntuid üldelemente. Uuteks elementideks on näiteks `<header>`, `<footer>`, `<section>`, `<nav>`, `<article>`, mis vähendavad varem tuntud `<div>` elemendi kasutamist. Paljud vananenud elemendid on kustutatud. HTML5-s on ka paljusid pluginavabu võimalusi, nagu näiteks standardiseeritud heli- ja videoliides, rastergraafika kujutamine, enda kohalik andmebaas, oma võrguvaba režiim, võimekam JavaScript jne. HTML5-e reeglite hulka kuuluvad põhimõtted, kus uued funktsioonid peaksid kasutama HTML-i, CSS-i, DOM-i ja JavaScripti ja tuleb vähendada väliste pluginate kasutamist (nagu Adobe Flash), paremini käsitleda vigu, rohkem *Markup*-i (märgistuskeelt), mis asendaks skriptimise. Samuti HTML5 peaks olema seadmest sõltumatu ja standardi arendamise protsess peaks olema avalik. HTML5 spetsifikatsioon ei ole veel lõplikult valmis. Seda arendatakse alles, ametlikult kinnitatud standard peaks valmima alles aastal 2022 (James, 2008). Siiski on võimalik kasutada HTML5 elemente juba täna, kuid teatud ettevaatlikkusega. Vaatamata sellele, et paljud tänapäeva brauserid ühilduvad järjest rohkem HTML5-ga, ei ole veel täielikku HTML5-e ühilduvust ühelgi veebilehitsejal.

### 1.3. XHTML5-e tutvustus

XHTML5 ehk laiendatav hüpertekst-märgistuskeel 5 on XML-i „versioon“ HTML5-st. XHTML5 dokument on koostatud XML MIME tüübina, näiteks *application/xhtml+xml*. Samuti nõuab XHTML5 rangemat ja hästi arenenud süntaksit. XHTML5 dokumendis HTML5-e dokumendi deklareerimine ei ole kohustuslik ja võib olla välja jäetud. XHTML5 võib kasutada näiteks selleks, et laiendada HTML5-e võimalusi mõnele XML põhisele tehnoloogiale nagu SVG (mastabeeritav vektorgraafika) ja MathML.

### 1.4. CSS3-e tutvustus

Uus versioon CSS-ist lubab palju rohkem paindlikkust, läbi oma moodulite. CSS3 on samuti arendamisel, kuid osasid võimalusi saab juba rakendada. Uusi võimalusi on üpriski palju:

- Äärejooned (*borders*): äärejoonena saab kasutada pilte või teha joonele ümarad servad.
- Tagataustad: mitmeid tagataustu saab lisada ühele elemendile, neid saab täpsemalt positsioneerida. Tagataustu saab laiendada ja siduda äärejoone sisepoolega või välispoolega. Taustade suurust saab ka suurendada või vähendada.
- Värv: Saab reguleerida värviüleminekuid (*gradients*) ja kasutada seda taustal. CSS toetab ka HSL värviväärtusi.
- Tekst: tekstile on lisatavad varjud. Toetab *text-overflow*-d ja reamurdmist.
- Transformatsioonid (*transformations*): elementide skaleerimine, pööramine, nihutamine, moondamine 3D või 2D ruumis.
- Üleminekud (*transitions*): Lihtsad dünaamilised üleminekud.
- Kast (*box*): kastile saab lisada varjud, lasta kasutajal muuta kasti suurust, sättida *overflow* nii horisontaalselt kui ka vertikaalselt. Samuti kasutada kastimudelisätteid, et paika panna viisi, kuidas kasti kõrgus ja laius muutub.
- Sisu: stiilid saavad anda sisu elemendile.
- Läbipaistvus: elementidel saab sättida läbipaistvust.
- Meedia: stiililehed, mis on määratud vaateava suurust, värvi, proportsiooni, resolutsiooni ja muid tähtsaid disaini elemente silmas pidades.
- Veebifondid: uuendab ja laiendab võimalusi, et siduda fonte, mida kasutatakse disainil.

## 2. HTML5 elementide rakendamine brauserites

### 2.1. Brauseritest ja nende kasutamisest

HTML5 on tagurpidi ühilduv, mis tähendab, et sisuliselt võib kasutada HTML4 kirjutusviise, kui sellega on harjutud. See on ka soovitatav, kuna paljudel inimestel on veel käibel HTML5-te mittetoetavad brauserid. Selles töös kasutatakse testimiseks, praegu kasutusel olevaid, kõige uuemaid veebilehitsejaid: Mozilla Firefox 10.0, Internet Explorer 9, Opera 11.61, Google Chrome 16.0.912.77, Safari 5.1.2. Kohalik arvuti, kus teste läbi viiakse on operatsioonisüsteemiga Windows 7 Home Premium 64-bit.

Kasutatavate brauserite statistika:

**Mozilla Firefox**-i kasutab **37,1%**, millest FF10-t 0,8%, FF9-t 17,8%, FF8-t 7,2%, FF3.6-t 5%.

**Internet Explorer**it kasutab **20,1%**, millest IE9-t 5,3%, IE8-t 10,5%, IE7-t 3,1%.

**Google Chrome** kasutab **35,3%**, millest C16-t 30,5%.

**Safarit** kasutab **4,3%**, millest S5-t 4,2%.

**Operat** kasutab **2,4%**, millest O11 -1,7%.

("w3schools.com," jaanuar 2012)

On olemas skriptid nagu näiteks „Modernizer“, mis on *open-source JavaScript-i library*, mis lubab HTML5 ja CSS3 elementide kasutamist vanematel brauseritel (IE6+, Firefox 3.5+, Opera 9.6+, Safari 2+, Chrome). Kuid antud katsetustes ei kasuta ma seda võimalust. Üldiselt on soovitatav jälgida tavaid, mis lubaks võimalikult paljudel kasutajatel näha veebilehitsejates veebilehte nii nagu see on mõeldud. Viimane lause tähendab seda, et märgistused peaksid olema alati lõpetatud nagu varasemates HTML-i versioonides ning soovitatavalt peaksid olema rakendatud abistavad skriptid, mis lubaksid vanematel veebilehitsejatel lugeda uut HTML5-e märgistust.

### 2.2. Üldine

HTML5 on tõstutundetu, mis tähendab sisuliselt, et võib kasutada suuretähelisi märgistusi ja väikesetähelisi märgistusi või isegi kasutada neid koos ja HTML5 ikkagi valideerib selle koodi. Näiteks see kood on tehniliste tingimustega vastavuses:

`<H1> HTML5 tekst </h1>`

Ennem tuntud sulgev märgistus pole enam vajalik. Näiteks lõikude lisamine seda moodi on lubatud:

`<p> Olen 1.lõik`

`<p> Olen 2.lõik`

`<p> Olen 3.lõik`

Atribuudile jutumärkide lisamine pole kohustuslik. Alltoodud kood on tehniliste tingimustega vastavuses:

`<img src=pilt1.jpg alt=Linnu pesa>`

Samuti, ei pea muutma oma vanu harjumusi, kuna HTML5 toetab ka vanu koodi kirjutamise mooduseid, kus igale elemendile järgnes sulgev märgistus. Samuti toetatakse ka jutumärkide kasutamist.

Pika *doctype* asemel on nüüd väga lühike märgistus.

Enne:

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

Nüüd vaid:

```
<!doctype html>
```

### **2.3. Video element**

Praeguseni pole olnud kindlat standardit videote või filmide näitamiseks veebis. Tänapäeval on suurem osa videofailidest kuvatud pluginate kaudu, nagu näiteks Adobe Flash Player ja QuickTime. Siin võivad tekkida probleemid, kuna erinevatel veebilehitsejatel on erinevad pluginad.

HTML5-s on loodud aga hea moodus, kuidas lisada veebilehele videofaili, kasutades `<video>` elementi. Lisaks on palju atribuute. Näiteks *control* lisab videole nupud nagu mängi, paus ja heli reguleerimine. Soovitav on märkida ära video mõõdud, mida video veebilehel hõivab, kuna siis reserveeritakse see ala veebilehe laadimise ajal. Ilma selleta, aga veebilehe paigutus muutub video laadimise ajal. Kuna aga video kodekrite toetus on erinev erinevatel veebilehitsejatel, tuleks lisada *source-i* mitu erineva laiendiga faili. Tabelis 1.1 on toodud kodekrite toetus brauserites. Soovitavalt oleks lisada vähemalt kaks formaati, mille vahel veebilehitsejad saaksid valida.



| Veebilehitseja      | MP4 (Kõrge kvaliteet) | WebM(Kõrge kvaliteet) | Ogg (Mitte nii hea kvaliteet kui Mp4-l) |
|---------------------|-----------------------|-----------------------|---|
| Internet Explorer 9 | Jah                   | Ei                    | Ei                                      |
| Firefox 4.0         | Ei                    | Jah                   | Jah                                     |
| Google Chrome 6     | Jah                   | Jah                   | Jah                                     |
| Apple Safari 5      | Jah                   | Ei                    | Ei                                      |
| Opera 10,6          | Ei                    | Jah                   | Jah                                     |

(Tabel 1.1, "w3schools.com", 2011)

Samuti on HTML5-l olemas märgend <track>, mille abil saab lisada jooksvat teksti meedia elementidele nagu <audio> ja <video>. Näiteks on võimalik nii lisada subtiitreid. Kuid hetke seisuga ei toeta ükski veebilehitseja seda võimalust veel.

Video elemendi katsetamine kohalikus masinas. Fail on Video.html (kood 1.1). Koodis on määratud erinevad atribuudid, mille väärtusi saab muuta ja vastavalt jälgida, kas brauser toetab seda või ei.

```

<video
  width="320"
  height="240"
  autoplay="autoplay"
  preload="none"
  poster="Pilt.jpg"
  muted="muted"
  loop="loop"
  controls="controls">
  <source src="Video.mp4" type="video/mp4" />
  <source src="Video.ogg" type="video/ogg" />
  Teie veebilehitseja ei toeta video elementi.
</video>

</body>
</html>

```

(Kood 1.1, Video.html)

Antud katsetuse tulemusi võime näha alltoodud tabelis 1.2.

| Veebilehitseja/atribuut      | autoplay | preload | poster | muted | loop | laiend |
|------------------------------|----------|---------|--------|-------|------|--------|
| Mozilla Firefox 10.0         | jah      | jah     | jah    | ei    | ei   | .ogg   |
| Internet Explorer 9          | jah      | jah     | ei*    | ei    | jah  | .mp4   |
| Google Chrome<br>16.0.912.77 | jah      | jah     | jah    | jah   | jah  | .mp4   |
| Opera 11.61                  | jah      | jah     | jah    | jah   | jah  | .ogg   |
| Safari 5.1.2                 | -        | -       | -      | -     | -    | -      |

(Tabel 1.2)

\*IE9 näitas *poster*-iga pilti video peal, kuid videot mängida ei saanud.

Antud testi läbis kõige paremini Opera 11. Ettearvamatult käitunud Safari 5.2 katsetuse tulemuseks on aga hoopis väljund „Teie veebilehitseja ei toeta video elementi“. Kuigi olen käitunud vastavalt juhistele ja lisanud MP4 koodekiga video, mida tabel 1.1 järgi peaks Apple Safari 5 toetama. Ka võttes antud video elemendis atribuute vähemaks, olukord ei muutunud. Teised veebilehitsejad mängivad ühte faili kahest ilma tõrgeteta.

Video.mp4 videofaili info:

Koodek: H264-MPEG-4 AVC

Eraldusvõime: 1280x720 pikslit

Kaadrisagedus: 25 kaadrit/sekundis

Samas tänapäeval on võimalus kasutada ka väliseid video mängijaid, mis põhinevad HTML5-l, nagu näiteks Video.js, mis koosneb JavaScriptist ja CSS-ist ning aitab parandada erinevate veebilehitsejate vahel tekkivaid tõrkeid ning lisab võimalusi mängida videot täisekraanil, subtiitritega ja erinevate *skin*-idega.

## 2.4. Audio element

HTML5-l on ka standard helifailide mängimiseks. Nagu video puhulgi kasutati erinevaid pluginaid nagu näiteks Adobe Flash. HTML5 kasutab `<audio>` elementi. See on äärmiselt mugav ja annab väga head võimalused failide lisamiseks oma veebilehele.

| Veebilehitseja      | MP3 | Wav | Ogg |
|---------------------|-----|-----|-----|
| Internet Explorer 9 | Jah | Ei  | Ei  |
| Firefox 4.0         | Ei  | Jah | Jah |
| Google Chrome 6     | Jah | Jah | Jah |
| Apple Safari 5      | Jah | Jah | Ei  |
| Opera 10,6          | Ei  | Jah | Jah |

(Tabel 1.1, "w3schools.com", 2011)

Audio elemendi katsetamine kohalikus masinas. Fail on Audio.html (kood 1.2). Lihtsas katsetuses testitakse, kas *preload*, *loop*, *controls*, *autoplay* töötavad ning mis faililaiendit brauser eelistab. Tulemused on tabelis 1.3. Brauser Apple Safari 5.1.2 andis väljundiks, et ei toeta audio elementi.

```

<!DOCTYPE html>
<html>
<body>

<!--
Audio süntaks:
<audio preload="auto|metadata|none">
<audio loop="loop">
<audio controls="controls">
<audio autoplay="autoplay">
-->

<audio controls="controls" preload="auto" loop="loop" autoplay="autoplay">
  <source src="Lugu.mp3" type="audio/mp3" />
  <source src="Lugu.wav" type="audio/wav" />
  Teie veebilehitseja ei toeta audio elementi.
</audio>

</body>
</html>

```

(Kood 1.2, Audio.html)

| Brauser/atribuut     | autoplay | preload | loop | laiend   |
|----------------------|----------|---------|------|----------|
| Mozilla Firefox 10.0 | jah      | jah     | ei   | .wav     |
| Internet Explorer 9  | jah      | ei      | jah  | .mp3/wav |
| Google Chrome 16     | jah      | jah     | jah  | .mp3/wav |
| Opera 11.61          | jah      | ei      | jah  | .wav     |
| Safari 5.1.2         | -        | -       | -    | -        |

(Tabel 1.3)

## 2.5. Canvas element

Canvas on uus element, mida kasutatakse graafiliste objektide kujutamiseks veebilehel. HTML5-e canvas kasutab JavaScripti. Canvas on iseenesest nelinurkne ala, mille igat pikslit saab kontrollida, ehk siis anda sellele mingi väärtus ja vastavalt kujutada mingit graafilist objekti sellel. Canvasel on palju erinevaid meetodeid, et sellele joonistada kastikesi, ringe, jooni, kujundeid ja lisada pilte.

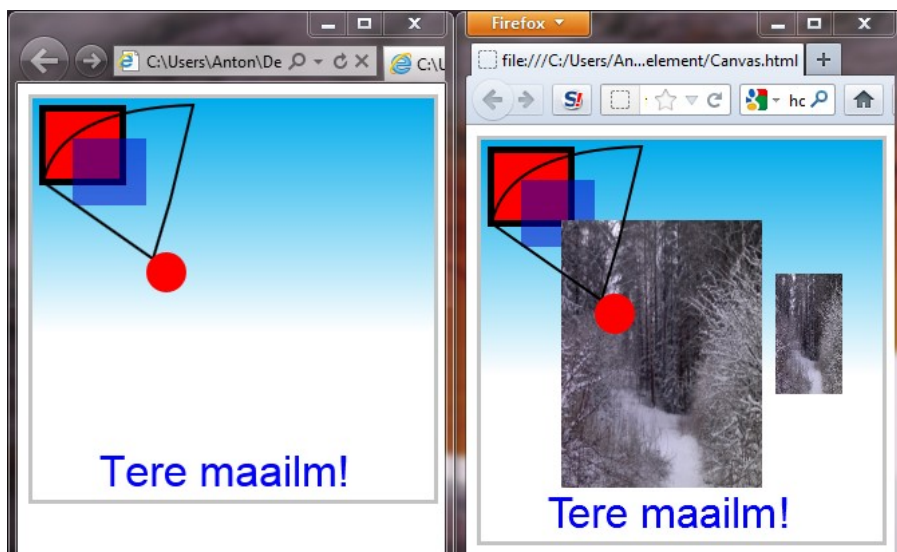
Canvas elemendi katsetamine kohalikus masinas. Fail on Canvas.html (*kood 1.3*). Tulemust, mida kood esitleb, on näha piltidel 1.1 ja 1.2. Üldiselt on antud võimalus väga mugav näiteks mingisuguste märgistuste tegemiseks mõnel pildil.

```
<!DOCTYPE html><html><body>
<canvas id="MinuCanvas" width="300" height="300" style="border:3px solid #c3c3c3;">
Teie veebilehitseja ei toeta canvas elementi.</canvas>
<script type="text/javascript">
var c=document.getElementById("MinuCanvas");
var ctx=c.getContext("2d");
var lineaarneuleminek = ctx.createLinearGradient(0,0,0,350);
lineaarneuleminek.addColorStop(0, '#00ABEB');
lineaarneuleminek.addColorStop(0.5, '#fff');
ctx.fillStyle = lineaarneuleminek;
ctx.fillRect(0,0,300,300);
ctx.fillStyle = "rgb(900,0,0)";
ctx.lineWidth = 10;
ctx.strokeRect(10,10,55,50);
ctx.fillRect(10,10,55,50);
ctx.fillStyle = "rgba(0,0,200,0.5)";
ctx.fillRect(30,30,55,50);
var pilt=new Image();
pilt.src="Pilt.jpg";
ctx.drawImage(pilt,60,60,150,200);
ctx.drawImage(pilt,220,100,50,90);
ctx.beginPath();
ctx.moveTo(8,62.5);
ctx.lineTo(90,120);
ctx.lineTo(103,76);
ctx.lineTo(120,5);
ctx.quadraticCurveTo(20,8,8,62.5);
ctx.lineWidth = 2;
ctx.stroke();
ctx.fillStyle = "blue";
ctx.font="24pt Helvetica";
ctx.fillText("Tere maailm!", 50, 290);
ctx.fillStyle="#FF0000";
ctx.beginPath();
ctx.arc(100,130,15,0,Math.PI*2,true);
ctx.closePath();
ctx.fill();
</script> </body></html>
```

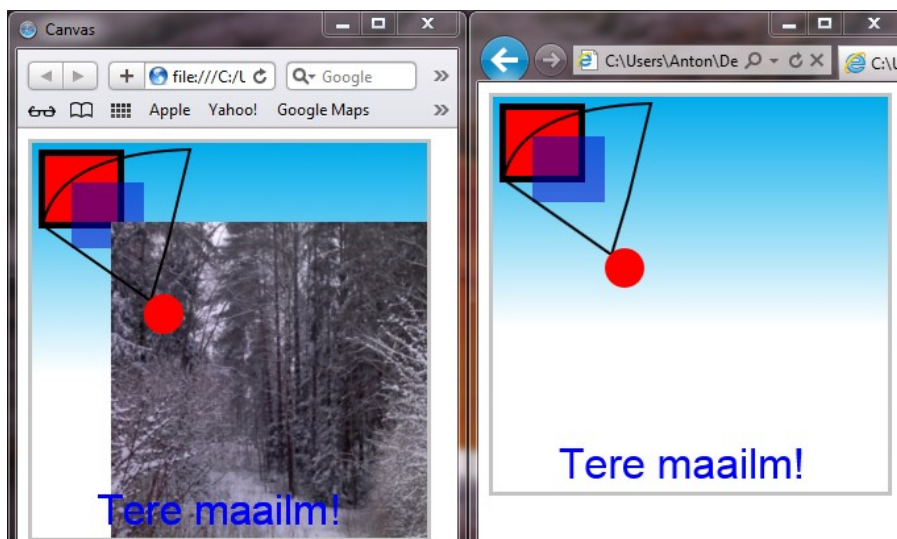
(Kood 1.3, Canvas.html)

| Brauser/atribuut     | Pilt vähendatud mõõdus | Pilt originaalmõõdus |
|----------------------|------------------------|----------------------|
| Mozilla Firefox 10.0 | jah                    | jah                  |
| Internet Explorer 9  | ei                     | ei                   |
| Google Chrome 16     | ei                     | jah                  |
| Opera 11.61          | jah                    | jah                  |
| Safari 5.1.2         | ei                     | jah                  |

(Tabel 1.4)



(Pilt 1.1, kus on IE9 vasakul ja FF10 paremal ning pildi on etteantud mõõtudes)



(Pilt 1.2, kus on Safari5.2 vasakul ja IE9 paremal ning pilt on originaalmõõdus)

Katsetuses erinevate kujundite, läbipaistvuse, värviülemineku ja teksti kujutamine õnnestus. Pilti näitasid etteantud mõõtudega aga ainult 2 veebilehitsejat. Pärast algkoodi muutmist, kustutades pildile etteantud mõõdud, ei näidanud pilti *Canvas* peal ainult Internet Explorer 9.

## 2.6. SVG element

SVG ehk *Scalable Vector Graphics* on keel, mis lubab kirjeldada 2d graafikat ja graafilisi elemente XML-is. Tavaliselt paljud veebilehitsejad näitavad SVG-d sama moodi kui kõiki teisi pildiformaate. Osad vajavad näiteks Adobe SVG Viewer-i installeerimist. HTML5 aga lubab lisada svg-d otse veebilehele, kasutades `<svg>` elementi.

SVG elemendi katsetamine kohalikus masinas. Fail on `Svg.html` (kood 1.4). Kood kujutab vektorgraafikat, nagu on kujutatud pildil 1.3.



(Pilt 1.3, `Svg.html` väljund)

```
<!DOCTYPE html>
<head>
<title>SVG</title>
<meta charset="utf-8" />
</head>
<body>
<svg height="200" xmlns="http://www.w3.org/2000/svg">
  <rect width="300" height="100" fill="orange" />
  <circle cx="50" cy="50" r="50" stroke-width="4" fill="red" />
  <polyline points="0,0 0,30 80,30 20,40 80,40 40,60" fill="yellow" />
</svg>
</body>
</html>
```

(Kood 1.4, `Svg.html`)

Lihtsa SVG testi läbisid Mozilla Firefox 10.0, Internet Explorer 9, Opera 11.61, Google Chrome 16.0.912.77 ja Safari 5.1.2 ilma probleemideta.

## 2.7. Vormid (*Forms*)

### Sisendväärtuste tüübid (*Input types*)

HTML5-l on paljud uued võimalikud sisendväärtuste tüübid. Need aitavad paremini kontrollida

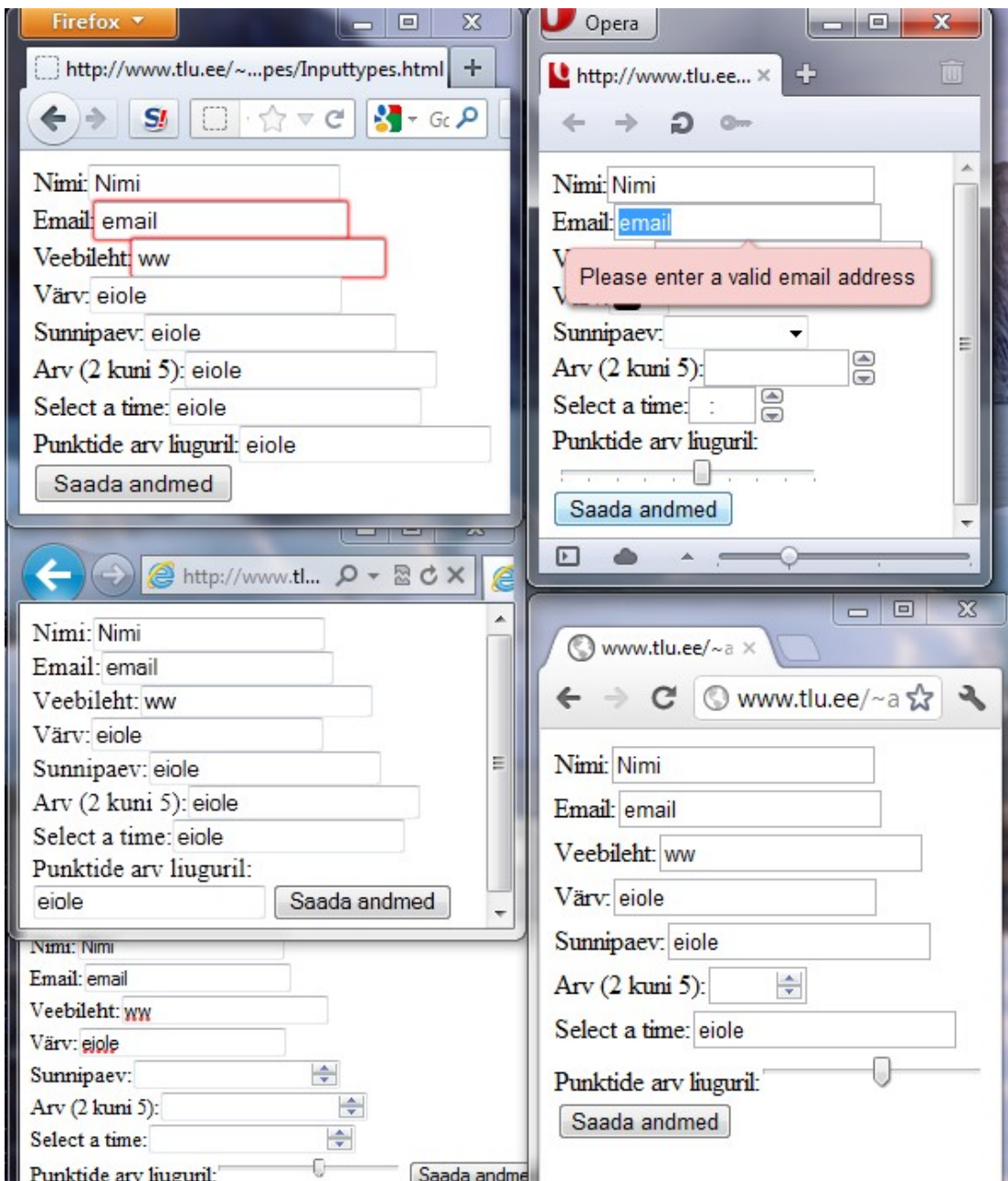
sisestatud andmeid ja vastavalt aktsepteerida neid kui vaja ja eirata, kui sisestatud väärtus ei vasta nõuetele. Kõik uued veebilehitsejad ei pruugi toetada uusi sisendväärtusi, kuid neid võib kasutada ikkagi, sest sisendlahter käitub sel juhul lihtsalt kui tavaline teksti lahter. Kuid näiteks Apple iPhone-l avaneb juba vastav klaviatuur, kui sisendväärtuse tüüp on määratud lahtril. Näiteks, kui on vaja sisestada email, iPhone'l ilmub klaviatuuril kohe *spacebar*-i kõrvale „@“ märk. Kui seda ei juhtuks, peaks kasutaja eraldi muutma klaviatuuri asetust, et pääseda ligi „@“ märgile.

Fail `Inpputtypes.html` (kood 1.5) käivitati `tlu.ee` serveris, milles on lahtrid tüüpidega: *text*, *email*, *url*, *color*, *date*, *number*, *time*, *range*. Selles testis kontrollime, kuidas käituvad veebilehitsejad, kui neile anda sisendväärtuste tüübid.

```
<!DOCTYPE HTML>
<html lang="en">
<body>
  <form id="Andmed" method="post" action="#">
    Nimi:<input type="text" name="nimi" tabindex="20"><br />
    Email:<input type="email" name="email" tabindex="30"><br />
    Veebileht:<input type="url" name="leht" tabindex="40"><br />
    Värv:<input type="color" name="varv" tabindex="50"/><br />
    Sunnipäev:<input type="date" name="kuupaev" tabindex="60"/><br />
    Arv (2 kuni 5):<input type="number" name="arv" min="2" max="5" tabindex="70"/><br />
    Select a time:<input type="time" name="usr_time" tabindex="80" /><br />
    Punktide arv liuguril:<input type="range" name="points" min="1" max="10" tabindex="90"/>
    <input type="submit" name="submit" id="submit" tabindex="80" value="Submit your review">
  </form>
</div>
</body>
</html>
```

(Kood 1.5, `Inputtypes.html`)

Tulemused on esitatud pildil 1.4 ja tabelis 1.5. Esimesena näitab Firefox punase tähistusega, kui üritada saata andmeid, et veebilehe ja e-maili lahtrisse sisestatud andmed pole õiges vormis ning palub need uuesti sisestada. Samamoodi käitub ka Opera ja Google Chrome. Internet Explorer ja Safari võtavad mistahes vormis vastavad andmed vastu ja lubavad neid saata. Ilmselgelt on näha pildilt 1.4, et kõige paremini toetab erinevaid sisendväärtuste tüüpe Opera veebilehitseja. Kahjuks aktsepteerivad Opera, Google Chrome ja Firefox e-maili ja veebilehe puhul ka sellised sisendid nagu „email@a“ ja „http://ww“, mis näitab seda, et ilmselgelt ei saa täielikult usaldada neid valideerimissüsteeme veel ning tuleb rakendada ka serveripoolseid kontrollimehhanisme.



(Pilt 1.4, testi ajal tehtud pilt, kus on kujutatud sisendid ja vastav veebilehitsejate käitumine.)



| Veebilehitseja/<br>Sisendtüüp | Internet<br>Explorer 9 | Mozilla<br>Firefox 10.0 | Google<br>Chrome 16 | Apple Safari 5                            | Opera 10,6                             |
|-------------------------------|------------------------|-------------------------|---------------------|---|--|
| text                          | jah                    | jah                     | jah                 | jah                                       | jah                                    |
| email                         | ei                     | jah                     | jah                 | ei  | jah                                    |
| url                           | ei                     | jah                     | jah                 | ei  | jah                                    |
| color                         | ei                     | ei                      | ei                  | ei  | jah (visuaalne<br>värvivalija)         |
| date                          | ei                     | ei                      | ei                  | jah                                       | jah (rikkalik<br>kalendri liides)      |
| number                        | ei                     | ei                      | jah                 | jah                                       | jah                                    |
| time                          | ei                     | ei                      | ei                  | jah (esialgne<br>väärtus on<br>hetke aeg) | Jah ( esialgne<br>väärtus on<br>00:00) |
| range                         | ei                     | ei                      | jah                 | jah                                       | jah                                    |

(Tabel 1.5)

### Vormi atribuudid (*Form attributes*)

Vormi atribuudid lubavad lisada sisendtüüpidele erinevaid omadusi ja lisavõimalusi. Failis Formattributes.html (kood 1.6) kasutatakse omadusi nagu: *autofocus*, *autocomplete*, *placeholder*, *required*, *date max ja min*, *step*, *datalist*.

```

<!DOCTYPE HTML>
<html lang="en">
<body>
  <form id="Andmed" method="post" action="#">
    Nimi:<input type="text" name="nimi" autofocus="autofocus" autocomplete="on" width="48" height="48" tabindex="20"><br />
    Email:<input type="email" name="email" novalidate="novalidate" tabindex="30"><br />
    Veebileht:<input type="url" name="leht" placeholder="veebileht URL-ina" tabindex="40"><br />
    Värv:<input type="color" name="varv" required="required" tabindex="50"/><br />
    Sunnipäev:<input type="date" name="kuupaev" max="1979-12-31" min="2000-01-02" tabindex="60"/><br />
    Arv (2 kuni 5):<input type="number" name="arv" min="2" max="16" step="2" tabindex="70"/><br />
    Select a time:<input type="time" name="usr_time" tabindex="80" /><br />
    Punktide arv liuguril:<input type="range" name="points" min="1" max="10" tabindex="90"/><br />
    Sisesta veebilehitseja:<input list="Veebilehitsejad" />
    <datalist id="Veebilehitsejad">
      <option value="Internet Explorer">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
    <input type="submit" name="submit" id="submit" tabindex="80" value="Submit your review">
  </form>
</div>
</body>
</html>

```

(Kood 1.6, Formattributes.html)

| <i>Veebilehitseja/<br/>Sisendtüüp</i> | Internet<br>Explorer 9 | Mozilla<br>Firefox 10.0 | Google<br>Chrome 16 | Apple Safari 5 | Opera 10,6 |
|---------------------------------------|------------------------|-------------------------|---------------------|----------------|------------|
| autofocus                             | ei                     | jah                     | jah                 | jah            | jah        |
| autocomplete                          | ei                     | jah                     | jah                 | jah            | ei         |
| placeholder                           | ei                     | jah                     | jah                 | jah            | jah        |
| required                              | ei                     | jah                     | jah                 | ei             | jah        |
| date max/min                          | ei                     | ei                      | ei                  | jah            | jah        |
| step                                  | ei                     | ei                      | jah                 | jah            | jah        |
| datalist                              | ei                     | jah                     | ei                  | ei             | jah        |

(Tabel 1.6)

### 3. HTML5-e API-d

#### 3.1. Veebilehe säilitamise API (*Webstorage API*)

Kohalikud andmed on kiiremini kättesaadavad, samuti ei ole nii palju koormust serveril endal. Informatsiooni saab ka teada näiteks ilma sisselogimiseta. Ei ole enam vaja kasutada küpsiseid.

##### **Sessiooni säilitamine (*Session Storage*)**

Kontrollib andmete säilitamist üheks sessiooniks ühes aknas.

##### **Kohalik säilitamine (*Local Storage*)**

Kontrollib andmete säilitamist mitmele aknale ja kauemaks kui ühele sessioonile. HTML5-e spetsifikatsioonis pole kirjas maksimaalset mahtu, mida kohalik säilitamine saab kasutada. Siiski on see suurem kui küpsistel. Neil on umbes 4 KB. Suurem osa brausereid annavad ühele veebihostile kohalikuks säilitamiseks aga umbes 5 MB. Samas küpsiste ees on ka järgmine eelis, et *Local Storage* andmed saadetakse serverile ainult, kui server seda nõuab. See aitab hoida kokku aega ja interneti ribalaiust, kuna andmeid ei saadeta ilma küsimata edasi-tagasi. *Local Storage* objektid säilitatakse ka nagu küpsisedki kõvakettal. Seetõttu pole kõige turvalisem viis andmete säilitamiseks.

Failid `WebStorageLocal.html` ja `WebStorageSession.html` (kood 1.7) käivitati tlu.ee serveris.

```

<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
if (localStorage.pagecount)
{
    localStorage.pagecount=Number(localStorage.pagecount) +1;
}
else
{
    localStorage.pagecount=1;
}
document.write("Localstorage külastusi on " + localStorage.pagecount + "
.");
</script>
</body>
</html>
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
if (sessionStorage.pagecount)
{
    sessionStorage.pagecount=Number(sessionStorage.pagecount) +1;
}
else
{
    sessionStorage.pagecount=1;
}
document.write("Sessionstorage külastusi on " + sessionStorage.pagecount
+ " .");
</script>
</body>
</html>

```

(Kood 1.7, WebStorageLocal.html ja WebStorageSession.html)

Kõik veebilehitsejad läbisid selle testi probleemideta ja väljastasid akna sulgemisel vastavad väärtused. *Session Storage* vastavalt alustas lugemist algusest, peale igat lehe sulgemist, kuid *Local Storage* säilitas info brauseris ja mäletas ka eelmiste külastuste arvu.

### 3.2. Tõmba ja aseta API (*Drag-and-drop API*)

„Lohista ja aseta“ on väga lihtne võimalus HTML5-s. See annab võimaluse vajutada elemendile ja lohistada see teise asukohta. HTML5-s on see võimalus standard ja igat elementi võib lohistada.

Fail DragandDrop.html (kood 1.8) käivitati tlu.ee serveris. Failis on võimalik eraldi olev pilt lohistada musta piirjoonega ümbritsetud kastikesse.

```

<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
#div1 {width:300px;height:300px;padding:10px;border:2px solid #000000;}
</style>
<script type="text/javascript">
function Kirjutaule (ev)
{
ev.preventDefault();
}

function Tomba (ev)
{
ev.dataTransfer.setData("Text",ev.target.id);
}

function Aseta (ev)
{
var Andmed=ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(Andmed));
ev.preventDefault();
}
</script>
</head>
<body>

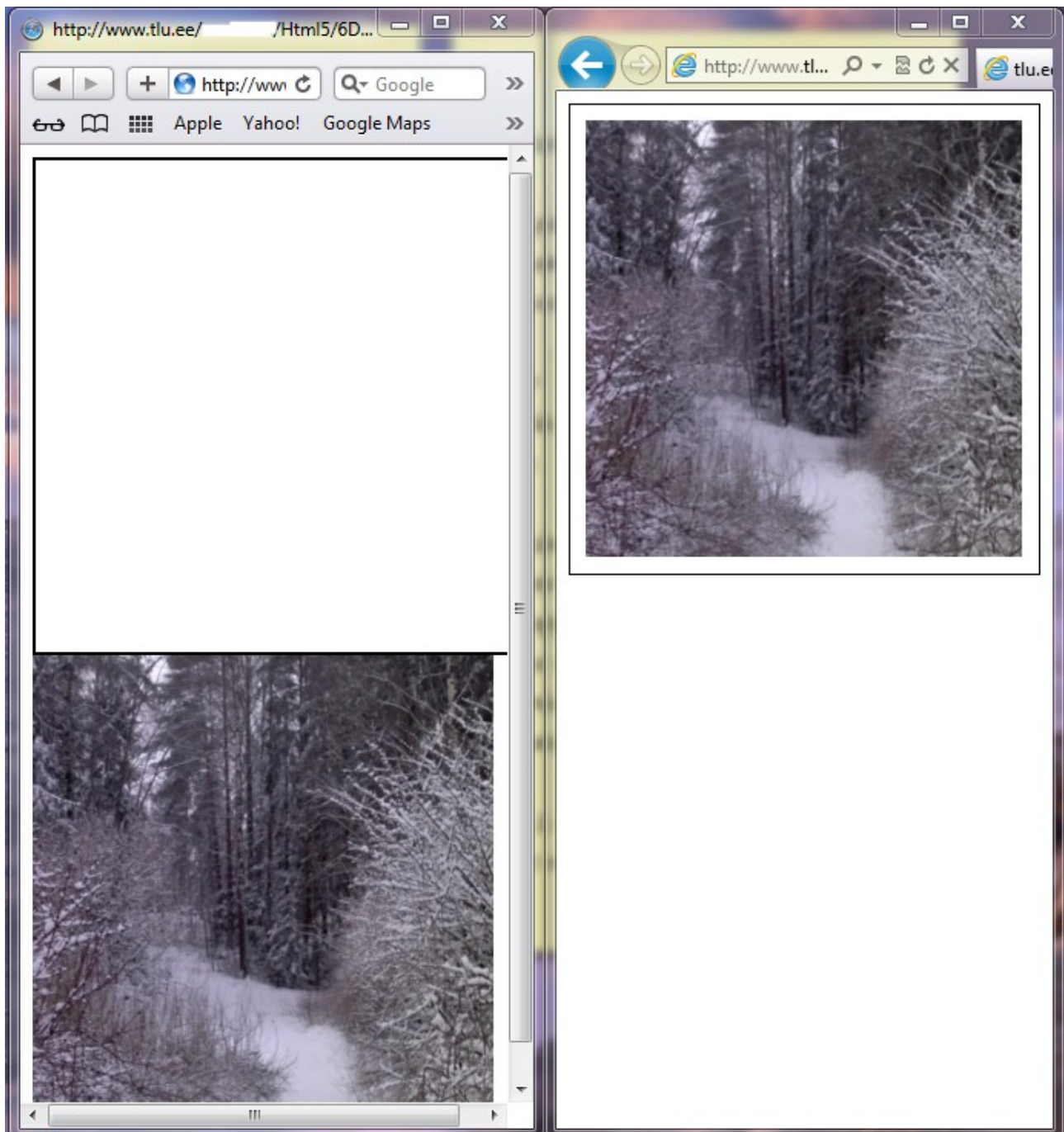
<div id="div1" ondrop="Aseta(event)"
ondragover="Kirjutaule(event)"></div>


</body>
</html>

```

(Kood 1.8, DragandDrop.html)

Kolm veebilehitsejat viiest läbisid lihtsa testi probleemideta. Häтта jäid Opera ja Safari, mis ei lasknud objekti selle koodiga lohistada.



(Pilt 1.5 , DragandDrop.html Safari 5.1.2-s vasakul ja IE 9-s paremal.)

### 3.3. Geograafiline asukoht (Geolocation API)

Lubab arendada kasutaja asukohta kasutavaid veebilahendusi ning vastavalt kasutada asukoha informatsiooni. Lahendus küsib ka luba kasutajalt , et saada teada kasutaja asukoht, mis lahendab ka privaatsusprobleemi. Hetkel on see API arendamise käigus.

Fail Geo.html (kood 1.9) käivitati kohalikus arvutis. See lihtne test väljastab juhul, kui brauser toetab *Geolocation API*-d meile meie laius- ja pikkuskraadid. Vastavalt, kui aga veebilehitseja ei

toeta võimalust on väljundiks teade, et veebilehitseja ei toeta *Geolocation*-it

```
<!DOCTYPE html>
<html>
<body>
<button onclick="AvaldaAsukoht()">Minu asukoht</button>
<script>
function AvaldaAsukoht()
{
  if (navigator.geolocation)
  {
    navigator.geolocation.getCurrentPosition(NaitaAsukoht);
  }
  else
  {
    var outputStr2 ="Brauser ei toeta Geolocation-it.";
    document.getElementById("Proov").innerHTML=outputStr2;}
  }
function NaitaAsukoht(asukoht)
{
  var outputStr =
  "Laiuskraadid on: " + asukoht.coords.latitude +
  "<br />Pikkuskraadid on: " + asukoht.coords.longitude;
  document.getElementById("Proov").innerHTML=outputStr;
}
</script>
<p id="Proov"></p>
</body>
</html>
```

(Kood 1.9, Geo.html)

Probleem tekkis ainult Apple Safari 5.2-ga, mis küsis küll luba asukoha avaldamiseks, kuid andes nõusoleku, ei väljastanud ta mitte mingit tulemust. Kõik muud brauserid toetasid asukoha määramise võimalust.

## 4. CSS3-e elemendid

### 4.1. CSS3 Font-face

Enne CSS3-e veebidisainerid pidid kasutama *web-safe* fonte ehk neid fonte, mis olid installeeritud juba kasutaja arvutisse. CSS3-s saab veebidisainer kasutada mistahes fonti ta ise tahab, kuid viide peab olema litsentseeritud fondile, mis on .ttf või .otf formaadis. Fondi kasutamine toob endaga kaasa selle fondi allalaadimise kasutaja arvutisse, kui seda veel pole seal. Firefox, Chrome, Safari ja Opera toetavad .ttf ja .otf laiendeid. IE9+ toetab .eot (Embedded OpenType) laiendit.

Fail FontFace.html (kood 1.10) käivitati tlu.ee serveris. Faili sees on *@fontface* reegel nimega „Fonditest“, fondi failid laetakse serverisse. Esimene fondi fail .ttf on kõikidele veebilehitsejatele väljaarvatud Internet Explorerile, millele on eraldi antud .eot fondi fail. Vastavalt, kui .ttf fail ei sobi veebilehitsejale, laetakse järgmine .eot fail. Selles näites kasutatud fondid on vabavaralise litsentsiga.

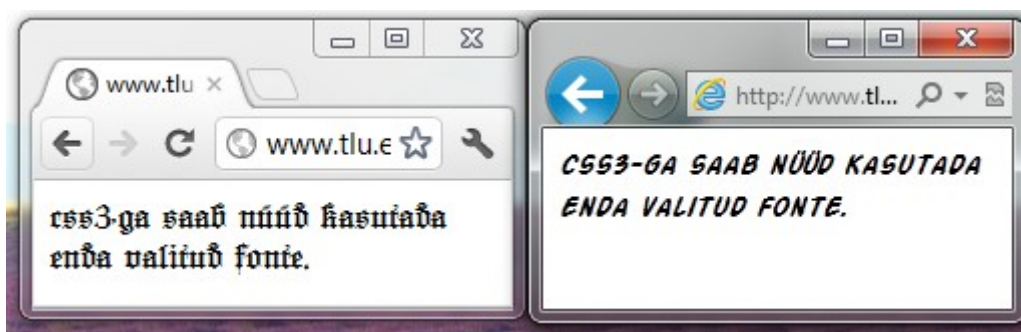
```
<!DOCTYPE html><html>
<head>
<style type="text/css">
@font-face
{
font-family: Fonditest;
src: url('Dearest_open-webfont.ttf')
      ,url('Action_Man_Bold_Italic-webfont.eot'); /* IE9+ */
}

div
{
font-family:Fonditest;
}
</style>
</head>
<body>

<div>
css3-ga saab nüüd kasutada enda valitud fonte.
</div>
```

(Kood 1.10, FontFace.html)

Kõik toimis nii nagu ennustatud. Kõik veebilehitsejad ,väljaarvatud IE9, kaustasid .ttf fonti. Ja brauserid kuvasid teksti probleemideta. Kusjuures oli märgatav umbes 1 sekundiline paus fondi laadimisel IE9-s.



(Pilt 1.6, Google Chrome ja IE9 kasutasid erinevaid fondi faile)

## 4.2. CSS3 üleminekud (CSS3 transitions)

CSS3 annab võimaluse kasutada efekti, kui muudame ühte stiili teistsuguseks, ilma Adobe Flash-i ja JavaScripti abita. Fail FontFace.html (*kood 1.11*) käivitati tlu.ee serveris. Faili sisu on järgmine: taustaks on üks pilt, millel on 20 pikseline ruut, millele hiirega liikudes venib selle üks külg 400 pikseliseks 2 sekundi jooksul järkjärgult.

Kõik veebilehitsejad läbisid selle lihtsa testi lihtsalt ja ühtemoodi, väljaarvatud Internet Explorer 9, mis ei oodanud kahte sekundit, et teha stiili muutust ja venitada kollast kastikest. Nimelt IE9 venitas kastikese 400 piksli pikkuseks silmapilkselt. Siit järeldus, et selline lihtne transformeerimine ei ole toetatud Internet Explorer 9-s.

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
body{
background:url(Pilt.jpg);
-moz-background-size:300px 300px; /* Firefox 3.6 */
background-size:300px 300px;
background-repeat:no-repeat;
}
div
{
width:20px;
height:20px;
background:yellow;
transition:width 2s;
-moz-transition:width 2s; /* Firefox 4 */
-webkit-transition:width 2s; /* Safari ja Chrome */
-o-transition:width 2s; /* Opera */
}
div:hover
{
width:400px;
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

(*Kood 1.11*, FontFace.html)



## 5. Järeldus ja soovitused

Tööst võib järeldada ilmselgelt, et tuleb olla ettevaatlik, kui kasutada uut standardit HTML5 või CSS3. Selle tingib nähtud katsetuste tulemused, mis näitavad, et samad tähistused ja ootused, on erinevalt interpreteeritud erinevates veebilehitsejates. Samuti on töös äärmiselt lihtsad katsetused ning keerulisemate lahenduste puhul peaks rohkem vaeva nägema, et veenduda nende kindluses. Kuid vaadates brauserite kasutamise statistikat, näeme, et vähesed kasutavad vaid kõige uuemaid tarkvaralahendusi, seega pole töös käsitletud uued standardid veel kasutamiseks kõigile inimestele maailmas.

Soovitav on kindlasti kasutada erinevaid tänapäeval leiduvaid skripte, mis annavad laiemat toetust ka vanematele veebilehitsejatele. Kindlasti peaks mõtlema ka uue süntaksi dubleerimisele, kus näiteks tuvastataks kasutaja lehitseja ja vastavalt suunatakse ta kaasaegsele lehele või siis vanemat süntaksit kasutavale lehele. Järgmine oluline punkt oleks kasutaja teavitamine, et ta peaks kasutama selle veebilehe külastamiseks soovitatavalt teatud versioonist uuemat veebilehitsejat. Samuti on abistav sellel lehel kasutatud standardite kirja panek ja näitamine, et targem kasutaja teaks, miks tal töötab või mistõttu tal ei tööta teatud funktsioonid antud lehel. Kindlasti inimestele, kes soovivad minna kaasa uuendustega, soovitaks jätta kättesaadavaks kliendile ka vanem versioon antud lehest, vähemalt mingiks ajaks, kui on võimalik. Kasutajana peaksid inimesed alati kasutama kõige uuemaid veebibrausereid. Kuid tihti on ka probleemid, kus automaatne uuendus väljalülitatud või mingil muul põhjusel tarkvara ei uuendu. Kui isik tahab kasutada aga kõiki tänapäeva võimalusi, siis peaks aegajalt jälgima, kas veebibrauser on uuendatud või ei. Samuti mõne erilise võimaluse nagu näiteks *Geolocation API* jaoks tuleks kindlasti vaadata üle enda brauseri sätteid, et ei oleks keelatud saata oma asukohta. Üldiselt HTML5-e kasutamiseks oleks soovitatav kasutada Opera brauserit ja järgmine valik oleks Google Chrome või Firefox.

Järgnevas tabelis 1.7 on kujutatud ülevaatlikult antud töös läbiviidud testid. Esimene jah või ei vastavas lahtris kujutab autori soovitus, kas antud element on sobilik kasutamiseks antud brauseris. Järgnev väärtus on kokkuvõtte sellest, kas antud veebilehitseja läbis talle eelpool töös antud testid või ei läbinud neid. Allpool on toodud kokkuvõtte „jah“ väärtustest kummagi tulemuse kohta kokku. Autor järgis antud soovitusel andmisel põhimõtet, et elementi saab kasutada lihtsalt ja põhilised ootuspärased funktsioonid toimivad.

| <i>Veebilehitseja/Testitud<br/>omadus</i> | Internet<br>Explorer 9 | Mozilla<br>Firefox 10.0 | Google<br>Chrome 16 | Apple Safari<br>5 | Opera 10,6 |
|---|------------------------|-------------------------|---------------------|-------------------|------------|
| Video                                     | jah/ei                 | jah/ei                  | jah/jah             | ei/ei             | jah/jah    |
| Audio                                     | jah/ei                 | jah/ei                  | jah/jah             | ei/ei             | jah/ei     |
| Canvas                                    | jah/ei                 | jah/jah                 | jah/ei              | jah/ei            | jah/jah    |
| SVG                                       | jah/jah                | jah/jah                 | jah/jah             | jah/jah           | jah/jah    |
| Vormid (Forms)                            | ei/ei                  | jah/ei                  | jah/ei              | jah/ei            | jah/ei*    |
| Veebilehe säilitamise<br>API              | jah/jah                | jah/jah                 | jah/jah             | jah/jah           | jah/jah    |
| Tõmba ja aseta API                        | jah/jah                | jah/jah                 | jah/jah             | ei/ei             | ei/ei      |
| Geograafiline asukoht                     | jah/jah                | jah/jah                 | jah/jah             | ei/ei             | jah/jah    |
| CSS3 @Font-Face                           | jah/jah                | jah/jah                 | jah/jah             | jah/jah           | jah/jah    |
| CSS3 üleminekud                           | ei/ei                  | jah/jah                 | jah/jah             | jah/jah           | jah/jah    |
| Kokku jah-e                               | 8/5                    | 10/7                    | 10/8                | 6/4               | 9/7        |

(Tabel 1.7, kokkuvõtte tabel)

\*ainult *autocomplete* ei töötanud

## Kokkuvõte

Kokkuvõtteks võib öelda, et HTML-i ajalugu on väga pikk ning antud muutus on teretulnud. 1945.a pärit arvutite vaheliste ühenduste visioon on saanud teoks läbi HTML standardi. HTML 1.0, HTML 3.2, HTML 4.0 on olnud suurepärased eelkäijad HTML5-le. Kuid uusim versioon erineb eelnevatest päris palju.

Käesolevas töös võib vaadata, kuidas äärmiselt lihtsad näited toimisid brauserites või kas veebilehitseja üldse andis teatud võimalust kasutada. HTML5-e uued elemendid, mida katsetati on siiski veel natuke ebakindlad, kui toetuda vaid nendele, kasutamata abivahendeid. Elemendid SVG ja *FontFace* on ainukesed, mis tehtud testides läbiti kõigi brauserite poolt. Kuid, kui mitte kasutada absoluutselt kõiki võimalusi, mida ettevõtetud uus süntaks annab, vaid valida teatud objektist ainult töötavad omadused, nagu seda saab teha *Canvase* ja vormide abil, on võimalik neid integreerida tänapäeva veebilehtedele. Kindlasti peab läbi viima erinevad testid, et olla veenduda lahenduse töövalmiduses. Täiesti kindlat, igas brauseris töötavat lahendust HTML5 üksinda ei paku, nagu selgus tööst. Samuti on töös näha, et erinevad veebilehitsejad interpreteerivad erinevaid võimalusi erinevalt ning ükski veebilehitseja ei toeta kõiki võimalusi. Eelõeldu kehtib ka CSS3-e kohta. Kõige paremini pidasid tehtud katsetustel vastu brauserid Google Chrome 16.0.912.77 ja Opera 11.61. Kõige halvemini läbis testid Apple Safari 5.1.2.

Kindlasti on selle valdkonna testimise võimalused lõpmatud, kuna süntaks areneb alles. Samuti annab see töö kõneainet teiste vahendite uurimiseks, mis aitavad HTML5-te ja CSS3-e mõista kõikidel veebilehitsejatel ühtemoodi. Loodetavasti ühilduvad brauserid järjest rohkem HTML5-e ja CSS3-e nõuete ja võimalustega, selleks on vajadus igatahes olemas. Senimaani peaks aga kindluse mõttes järgima eelmises peatükis toodud juhiseid. Kindlasti annab see töö võrdlemise materjali tulevikus tehtavatele taoliste uuringutele.

## Kasutatud kirjandus

1. W3C. (1992, November 13). *Tags used in html*. Saadud URL-lt <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>
2. W3C. (2001, Juuni 20). *The original proposal of the www, htmlized*. Saadud URL-lt <http://www.w3.org/History/1989/proposal.html>
3. Bidgoli, H. (2004). *The Internet encyclopedia: 1. köide*. NJ: John Wiley and Sons.
4. Bidgoli, H. (2004). *The internet encyclopedia*. John Wiley & Sons Inc. Saadud URL-lt <http://books.google.com/books/feeds/volumes?q=9780471222019>
5. Bidgoli, O. (2005). *The internet encyclopedia*. John Wiley & Sons Inc. Saadud URL-lt <http://books.google.com/books/feeds/volumes?q=0471222011>
6. W3C. (2011, Oktoober 6). *A vocabulary and associated apis for html and xhtml*. Saadud URL-lt <http://dev.w3.org/html5/spec/>
7. Woodford, H., Moschovitis, H. J. P., Poole, H. W., Group, M., & Lambert, L. (1969). *The internet: a historical encyclopedia*. Saadud URL-lt <http://books.google.com/books/feeds/volumes?q=1851096590>
8. Mavrody, S. (2010). *Serge'ys html5 & css3 quick reference*. (EBook Edition ed.). Saadud URL-lt <http://books.google.com/books/feeds/volumes?q=0983386706>
9. Teague, J. C. (2010). *css3: visual quickstart guide*. (5 ed.). Peachpit Press. Saadud URL-lt <http://books.google.com/books/feeds/volumes?q=0131389920>
10. *w3schools.com*. (2011). Saadud URL-lt <http://www.w3schools.com/html5/>
11. *www.tutorialspoint.com*. (2011). Saadud URL-lt [http://www.tutorialspoint.com/html5/html5\\_svg.htm](http://www.tutorialspoint.com/html5/html5_svg.htm)
12. Williamson, J. (2010). *Html5 first look* [DVD].
13. Upton, J. (2011, November 29). *Windows internet explorer testing center*. Saadud URL-lt <http://samples.msdn.microsoft.com/ietestcenter/>
14. Mozilla Developer Network. (2011, Aprill 22). *Mdn*. Saadud URL-lt <https://developer.mozilla.org/en>
15. Dixit, S. (2010, Juuni 16). *How to use the w3c geolocation api*. Saadud URL-lt

- <http://dev.opera.com/articles/view/how-to-use-the-w3c-geolocation-api/>
16. Tittel, E., & Noble, J. (2008). *Html, xhtml & css for dummies*. (6th ed.). Hobken NJ: Wiley Publishing
  17. AGI Creative Team. , & Osborn, J. (2011). *Html5 digital classroom*. (1 ed.). Indianapolis: Wiley.
  18. Kessin, Z. (2011). *Programming html5 applications: Building powerful cross-platform environments in javascript*. (1 ed.). O'Reilly Media.
  19. Sanders, B. (2010). *Smashing html 5*. (p. 316). John Wiley & Sons.
  20. Bechynský, S. (2011, November). *Html5, part I – markup*. Slaidid esitatud konverentsil „HTML5 laager“, Tallinn.
  21. Vallaste, H. (2007, Aprill 30). *e-teatmik: It ja sidetehnika seletav sõnaraamat*. Saadud URL-It <http://vallaste.ee/>
  22. James, J. (2008, August 27). *Html 5 editor ian hickson discusses features, pain points, adoption rate, and more*. Saadud URL-It <http://www.techrepublic.com/blog/programming-and-development/html-5-editor-ian-hickson-discusses-features-pain-points-adoption-rate-and-more/718>