

Tallinna Ülikool  
Informaatika Instituut

# Veebilehtede korrektse välimuse tagamine CSS 3 vahenditega erinevatel seadmetel

Bakalaureusetöö

Autor: Helen Kuut

Juhendaja: Andrus Rinde

Autor: ..... „ ..... „2013

Juhendaja: ..... „ ..... „2013

Instituudi direktor: ..... „ ..... „2013

Tallinn 2013

# Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

# Sisukord

<b>SISSEJUHATUS .....</b>	<b>4</b>
<b>1 SEADMETE JA TEHNOLOOGIATE AJALUGU .....</b>	<b>6</b>
<b>2 ERINEVAD SEADMED VEEBILEHE VAATAMISEKS .....</b>	<b>9</b>
2.1 ARVUTIEKRAAN.....	9
2.2 TAHVELARVUTITE EKRAANID .....	11
2.3 NUTITELEFONIDE EKRAANID.....	12
2.4 VEEBILEHITSEJATE RESOLUTSIOON PIHUSEADMETEL .....	13
2.5 PRINTIMINE.....	14
<b>3 HTML 5 .....</b>	<b>16</b>
3.1 UUED HTML ELEMENDID .....	16
<b>4 CSS 3.....</b>	<b>20</b>
4.1 CSS 3 UUED MOODULID JA FUNKTSIOONID .....	20
4.2 MEEDIATÜÜBID JA PÄRINGUD.....	24
<b>5 VEEBILEHE KUJUNDUS ERINEVATE SEADMETE JAOKS .....</b>	<b>28</b>
5.1 ERITI KÕRGE RESOLUTSIOONIGA EKRAANID .....	28
5.2 PRINTIMINE.....	30
5.3 TAUSTAPILT .....	31
5.4 KIRJA SUURUS.....	32
5.5 KIRJA STIIL JA KAUNISTUSED.....	33
5.6 NAVIGATSIOONIVAHENDID .....	33
5.7 PILDID.....	34
5.8 VIDEOD JA HELIKLIPID.....	35
<b>6 AUTORI KOGEMUS VEEBILEHTE KUJUNDADES .....</b>	<b>37</b>
<b>KOKKUVÕTE.....</b>	<b>39</b>
<b>SUMMARY .....</b>	<b>40</b>
<b>KASUTATUD KIRJANDUS .....</b>	<b>42</b>
<b>LISA 1 .....</b>	<b>47</b>
INDEX.HTML.....	47
STIIL.CSS.....	50

# Sissejuhatus

Praegusel ajal areneb internetimaailm üha kiiremini edasi. Veeb ja veebipõhised teenused muutuvad järjest populaarsemaks. Luuakse aina uusi tehnoloogiaid veebilehtede rikastamiseks. Üha enam kasutatakse veebiteenuseid ja vaadatakse veebilehti erinevate mobiilsete seadmete vahendusel ning enamusel veebilehtedest pole erinevate seadmetega veel arvestatud ja paljudel juhtudel ei osata seda teha. Veebikujundaja peab aina enam arvestama erinevate seadmete ning ekraanisuurustega, et veebileht iga kasutaja jaoks kasutatav välja näeks.

Enamjaolt kujundatakse tänapäeval veebilehti CSS-i (*Cascading Style Sheets*) abiga. Pikka aega – aastast 1998 – kasutati veebilehtede kujundamiseks CSS 2-te, mille vähesed võimalused hakkasid veebikujundajate ideid ja võimalusi piirama. Nüüd viimaste aastatega on populaarsust kogunud CSS-i 3. versioon. See annab disainerile võimaluse veebi paremini ja ilusamalt kujundada. Lisaks veebilehe välimuse määravatele vahenditele on uuel CSS-il võimalused, mis teevad nii kujundamise kui ka kasutamise palju lihtsamaks, näiteks veebilehele mitme erineva taustapildi lisamine. Varem pidi kujundaja Javascript'ilt abi otsima, kui klient midagi huvitavat oma veebileheküljele lisada tahtis, või tuli pilditöötlus programmi poolt, et efektse teksti asemel pilti kasutada. Nüüd on enamus huvitavaid efekte võimalik CSS 3-ga ära lahendada. (Ritso, 2011) Tehnilisest küljest lisandub pidevalt uusi seadmeid erinevate ekraanisuurustega. Veebilehti ei vaadata enam vaid laua- või sülearvutitega, lisandunud on ka tahvelarvutid ja nutitelefonid. Suuremad võimalused ja erinevad seadmed aga tekitavad ka uusi probleeme, näiteks kuidas tagada, et loodud veebilehed erinevatel seadmetel ja alati head välja näeksid.

Hoolimata sellest, et HTML 5 ja CSS3 on alles arendusjärgus, on nad juba praegu parimateks vahenditeks erinevatele seadmetele veebilehtede kujundamiseks. HTML 5 ise on küll pidevalt uuenemas ja paranemas, kuid juba praeguseks on see saavutanud populaarsuse, kuna võrreldes varasemate versioonidega, on ka sellel uusi võimalusi muuta veebileht huvitavamaks. Veel rohkem muutub aga CSS 3, mille erinevad moodulid on erinevate veebilehitsejate poolt erinevalt toetatud. Mõned funktsioonid vajavad prefiksita – *-moz-*, *-khtml-*, *-o-* – kasutamist, mõni töötabki hetkel vaid ühel veebilehitsejal.

Erinevate seadmete iseärasust arvestavat või kirjeldavat kokkuvõtlikkus materjali pole leida ning mitmeid iseärasusi seoses erinevate seadmetega ei mainita peaaegu kusagil, seepärast otsustas autor sellisel teemal bakalaureusetöö kirjutada ning tutvustada, kuidas HTML 5 ja CSS 3 vahenditega luua igas olukorras töötav veebileht. Paljud veebilehed ei arvesta erinevate seadmetega – näiteks meie enda kooli leht [www.tlu.ee](http://www.tlu.ee) – ning neid on pihuseadmetel väga ebamugav sirvida. Mõned ettevõtted on probleemi lahendanud aplikatsioonide näol – näiteks Delfi tegi aplikatsiooni, et nutitefonidel oleks mugav uudiseid lugeda – kuid siiski võiksid kõik veebikujundajad tänapäeval arvestada erinevate seadmete ekraanidega ning paari CSS-i on palju kergem veebilehele juurde lisada kui aplikatsiooni koostada.

Selle bakalaureusetöö eesmärgiks on erinevate seadmete jaoks vajaliku veebilehe kujunduse loomise vahenditest ülevaate andmine, nende võimaluste testimine ning veebiloojatele, näiteks Tallinna Ülikooli informaatika instituudi tudengitele, kes suvepraktika käigus erinevaid veebirakendusi loovad, soovitude andmine, kasutades uut CSS 3-e ja HTML 5-te, kuna need sisaldavad palju uusi võimalusi ning on vabad paljudest varasematest piirangutest. Autor uurib välja lühidalt, millised on nende uued võimalused ning mis on kõige mõistlikum viis veebilehe kujunduseks, et see oleks kaasaegsetel seadmetel loetav. Samuti annab autor ülevaate praegusel ajal veebilehede vaatamiseks kasutatavatest seadmetest ning millised ekraani resolutsioonid populaarsemad on, et veebilehe tegijad saaksid nende suurustega arvestada. Lisaks koostab autor näidisveebilehe ning katsetab erinevate seadmete jaoks CSS kujundust, et veebileht näeks korrektne välja nii suurtel monitoridel, tavalistel sülearvutitel, tahvelarvutitel, nutitefonides kui ka välja printimisel.

Autori poolt loodud katselist veebilehte on võimalik näha aadressil:

[http://www.tlu.ee/~kuut1990/Asjad/.Baka\\_veeb/](http://www.tlu.ee/~kuut1990/Asjad/.Baka_veeb/)

# 1 Seadmete ja tehnoloogiate ajalugu

Veebi idee avalikustati aastal 1989 (Wikipedia, World Wide Web, 2012), esimene veebileht ja –lehitseja tuli välja 1990. aastal (Wikipedia, Tab (GUI), 2012) ning veebi nimi aastal 1991. (Who invented the Internet?, 2012) Sellest ajast on veeb ja sellega koos ka HTML populaarsust kogunud. HTML sai populaarseks keeleks 1990. aastate keskpaiku. Töö HTML spetsifikatsiooni väljatöötamiseks võttis enda kanda organisatsioon nimetusega W3C (*World Wide Web Consortium*). Tema ülesandeks oli spetsifikatsiooni koostamine, mis rahuldaks kaasaegseid keele võimalusi ja arvestaks veebilehitsejaid tootvate kompaniide mitmesuguseid ettepanekuid. (Mis on HTML, 2007)

Interneti meeletu menu tõi kaasa veebilehtede populaarsuse, tehnoloogia arenes, ekraanid kasvasid ja muutusid värvilisemaks, veebilehed, mis esialgu olid lausa tekstipõhised, muutusid värviliseks. Siis tekkis aga vajadus veebidisaini järele. Esialgu lisati kujundusvahendid HTML keelde, kõige enam sisaldas neid HTML 3.2, mis tuli välja 1997. aasta jaanuaris. (Wikipedia, HTML, 2012) Nii paigutatakse praegugi paljud HTML-i kujunduselemendid tabelisse, mis pole oma ülesehituselt mõeldud disainilahenduste loomiseks. See muutis hüpertekstidokumendid mahult suurteks ja nende kuvamise aeglaseks. Kõige suurem probleem oli aga veebilehe kujunduse muutmine – kõik HTML-i lehed tuli eraldi kujundada ning selline tegevus võis pikapeale tüütuks ja aeganõudvaks muutuda. Seetõttu tuli sama aasta detsembris välja uuendus – HTML 4.0 *Strict* - ning tuli leida uus lahendus, milleks sai CSS, mis on juba eksisteerinud 1994 aastast (kujunduslaadid – *style sheets* – on eksisteerinud erineval kujul juba 1980ndatest aastatest), kuid ametlikult kasutusel olnud aastast 1996. (Wikipedia, HTML, 2012) (Wikipedia, Cascading Style Sheets, 2012) CSS lahendas kergelt kujunduse ja kuvamiskiirusega seotud probleemid. Veebiarenduse meeletu populaarsus aga sundis ka HTML-i edasi arenema. Praeguseks on olemas HTML-i viies põhiredaktsioon.

Veebilehe loomisel on ka oluline teada erinevatest seadmetest, nende ekraanidest – millise suuruse ja resolutsiooniga (punktihedus, mida mõõdetakse pikselite arvuga tolli kohta) ekraane kasutatakse tänapäeval kõige enam. (Taurus, kuupäev puudub) Ajaloo vältel on kasutatud erinevate mõõtude ja resolutsioonidega monitore ning praegune suur ekraanide valik teeb veebidisaineri elu raskemaks. HTML 1 ja 2-ga veebilehtede loojad said ja pidid

arvestama vaid VGA ekraanidega – 800×600 pikslit ja 16 värvi (hiljem 256 värvi). HTML 3.2 kasutamise ajaks olid juba SVGA ekraanid - 800×600 pikslit ja 16 bit värvid (hiljem 24 bit). Veebikujundajad, kes kasutasid HTML 4 versiooni pidid arvestama juba 1024×768 piksliste ekraanidega (2000ndate aastateni), kuni ilmusid ekraanid suurustega 1280×1024 ja suuremad variandid. Tänapäeval on erinevad ka monitoride ekraanikujud: *portrait* ehk spetsiaalseid portree tüüpi, millel on kõrgus suurem kui laius, ja *landscape* monitorid, millel on laius suurem kui kõrgus. *Portrait* monitore kasutatakse näiteks kirjastustes, *landscape* monitore aga igapäevatoos ning need on ka levinumad. (Vikipeedia, Kuvar, 2012)

Tehnoloogia arenedes, luuakse üha uusi seadmeid, nagu näiteks tahvelarvuti. See huvitav arvuti on populaarseks muutunud just tema kaasaskantavuse tõttu. Kuna ta on nii õhuke ja kerge, siis sobib ta ideaalselt mobiilse eluviisiga inimestele. Lisaks on teda puutetundliku ekraani ja aku kestvuse tõttu mugav kasutada. Esimese tahvelarvuti kontseptsiooni mõtles välja Xerox'i töötaja Alan Kay juba 1968. aastal ja esitles esimest prototüüpi 1971. aastal. Kay nimetas seda "Dynabook"-iks. Esimene müüdüd tahvelarvuti oli Apple'i MessagePad 1993. aastal, mida tihti nimetatakse Newton'iks. MessagePad ei saavutanud suurt müügi edu. (Vikipeedia, Tahvelarvuti, 2012)

2000. aasta juunis tutvustas Bill Gates kõige esimest prototüüpi PC-tüüpi tahvelarvutist, mis oli osa .Net strateegia esitlusest. Kuigi see prototüüp oli märkimisväärne osa sellest esitlusest, ei ilmunud meedias sellel teemal siiski mitte ühtegi artiklit. Sama aasta novembris esitles Gates tahvelarvutit uuesti, mis oli sama riistvaraga. Siis esitles ta seda kui „pliiatsiga arvuti kasutamist“. Aasta hiljem tuli Gates välja jällegi uue prototüübiga, mille eesmärk oli näidata maailmale, et pliiatsiga saab kirjutada arvutisse tinti kulutamata. (Vikipeedia, Tahvelarvuti, 2012) Tahvelarvutid muutusid aga populaarselt iPad'i välja tulekuga 2010. aasta aprillis. (Wikipedia T. F., 2012)

Tänapäeval on võimalik internetis surfata ka telefoni vahendusel, seepärast on autorile oluline siin bakalaureusetöös anda pisike ülevaade ka nutitelefonide arengust.

Esimene nutitelefoni oli Simon (*personal communicator*), mille konstrueeris IBM 1992. aastal. (The First Smartphone, 2012) Simon töötas DOS operatsioonisüsteemi peal. (Well, You May Not Know My Name Is Simon, 2012) Lisaks mobiiltelefoni võimekusele sisaldas see telefon ka kalendrit, aadressiraamatut, kella vastavalt ajatsoonidele,

kalkulaatorit, märkmikku, e-maili, faksi (*fax*) saatmise ja vastuvõtmise võimalust ning mänge. (Vikipeedia, Nutifon, 2012) Simon'i ekraani suurus oli 3" ja resolutsioon 160×293 px. (The First Smartphone, 2012) Tänapäevaste standardite järgi oleks Simon madalama klassi toode, kuid omal ajal oli see uskumatult uudne. (Vikipeedia, Nutifon, 2012)

Algselt ei mõeldudki, et telefonidel on tavapärane võimalus veebilehti vaadata, kasutusel oli WAP (*Wireless Application Protocol*, standard, mis võimaldab WAP-teenust toetava mobiiltelefoniga saada ühendus mobiil-märgistuskeeles kirjutatud internetilehekülgedega ning kasutada seelsel info põhinevaid teenuseid). (Vikipeedia, Traadita rakenduste protokoll, 2012) Esimene mobiiltelefon, millel oli võimalik kasutada WAP veebilehitsejat oli Nokia 7110. See tuli turule oktoobris aastal 1999. Nokia 71 seeriat nimetati esimesteks meedia telefonideks, mis sisaldasid WAP-i ja tekstisõnumile võimalusi. (Wikipedia, Nokia 7110, 2012) Nüüdseks on aga võimalus telefonides samamoodi internetti kasutada nagu arvutiteski. Esimene telefon, mis sisaldas HTML veebilehitsejat oli Palm Treo 600 ning see tuli turule aastal 2003. (Reed, 2010) Selle veebilehitseja nimi oli Blazer (Gade, 2003) ning see võimaldas näha veebilehti sellisena nagu ka tänapäeva nutitefonid. Sellel oli QWERTY klaviatuur ning sisaldas ka SD-mälukaarti. (Wikipedia, Treo 600, 2012) Palm Treo töötas Palm operatsioonisüsteemiga ning see on meie praeguste nutitefonide eelkäija. (Reed, 2010)



## 2 Erinevad seadmed veebilehe vaatamiseks

Tänapäeval on võimalik internetis surfata väga erinevate seadmete abil. Varasemast ajast on juba tuntud lauarvuti ning sülearvuti. Nüüdseks on sinna lisandunud veel tahvelarvuti ja nutitelefoni, mis lisavad veebi disainile uusi nõudmisi. Selles peatükis annab autor ülevaate nendest seadmetest ning ekraanisuurustest, mille teadmine võib veebilehe tegemisel vajalik olla.

### 2.1 Arvutiekraan

Monitor (kuvar, videoterminal, ekraan) on arvuti väljundseade, mis muudab analoog- või digitaalinfo pildiks. See on üks tähtsamaid arvuti komponente kasutajasuunalise väljundseadmena. (Vikipeedia, Kuvar, 2012) Arvuti graafikakaardi abil kuvatakse monitorile videot ja muud graafilist informatsiooni. Need on väga sarnased televiisoritega, kuid tavaliselt näitavad nad palju kõrgema resolutsiooniga pilti. (Fisher, 2012)

Personaalarvutite juurde lisatakse tänapäeval tavaliselt kas:

- Vedelkristallkuvar ehk LCD.
- Plasmakuvar – eelkõige on need levinud suurte ekraanimõõtmetega (30 või rohkem tolli) teleritena. (Vikipeedia, Plasmateler, 2012)
- ja/või OLED-monitor ehk orgaaniline valgusdiod.

(Types Of Monitors, 2011)

Monitore eristatakse peamiselt nende suuruse ja tehniliste parameetrite järgi. Veebilehtede seisukohalt on oluline ära märkida järgmisi tehnilisi parameetreid:

- Lahutusvõimet ehk resolutsiooni (*screen resolution*) väljendatakse ekraanile mahtuvate pikslite arvuna, mis on ekraani lühema külje ja pikema külje pikslite korrutis, näiteks 800×600, 1024×768, 1152×864, 1440×900, 1920×1200. Seega

mida suurem on lahutusvõime, seda teravam on pilt ja seda suuremat ala saab tööpinnast näha.

- Suurust väljendatakse ekraani diagonaali pikkusega tollides. Levinumad mõõdud on vahemikus 15-24 tolli.

(Vikipeedia, Kuvar, 2012)

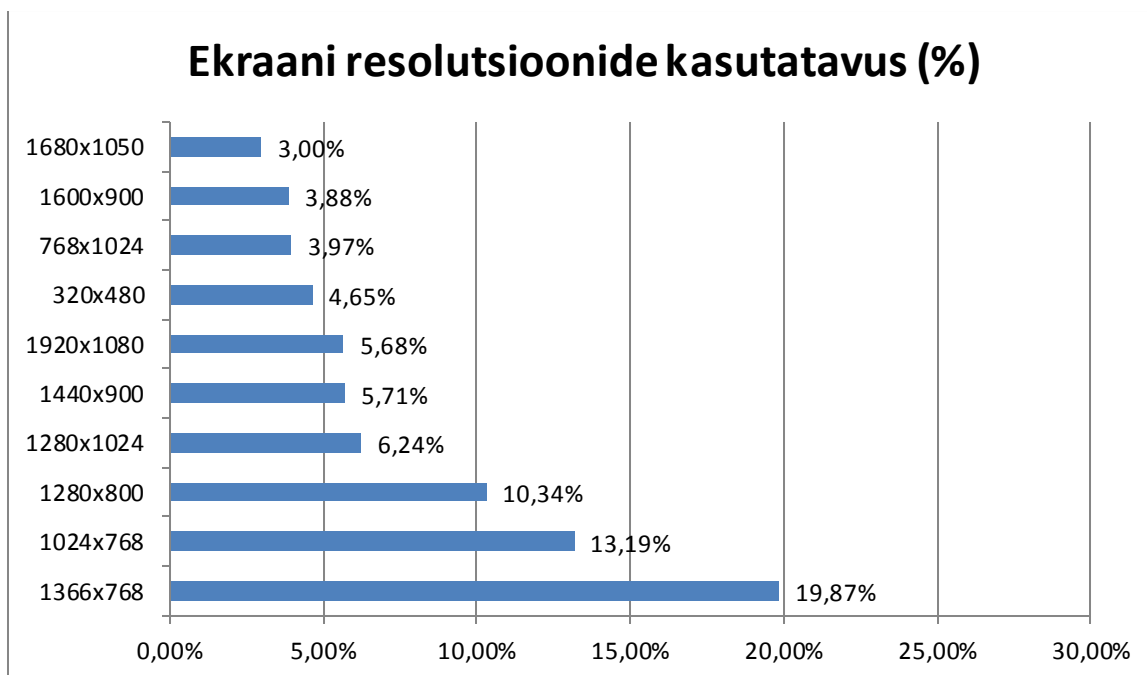
Suuremate monitoride hinnalangusega on läinud kasutajad aegapidi üle aina suurematele lahutusvõimetele. Alates 21. sajandi algusest on kasutusele võetud aina suuremaid laiekraan monitore, viimasel ajal on eriti kasvanud 1440×900 resolutsiooni populaarsus. Viimaste aastate jooksul on jõudsasti kasvanud ka laiekraan monitoride (16:9 ja 16:10) kasutajaskond ning HD-videode populaarsus soodustab ka kõrgemate resolutsiooni populariseerimist (1920×1080). (Vikipeedia, Lahutusvõime, 2012)

- Kuvasuhe (*aspect ratio*) näitab ekraani laiuse ja kõrguse suhet. Kui ekraan on ruut, siis on suhe 1:1. Praegu on enamjaolt kasutusel suhe 16:9, kuid pikka aega oli kasutusel suhe 4:3. (Vikipeedia, Kuvar, 2012) Nutitelefonide puhul kasutatakse suhteid 3:2, 16:9 või 5:3.

Sülearvutite ekraanide puhul jääb ekraani diagonaali pikkus üldiselt 10,6" ja 20" vahele ning võimalik on valida LCD, LED või OLED ekraanide vahel. (Vikipeedia, Sülearvuti, 2012) Populaarsemad ekraaniresolutsioonid on 1366×768 px, 1024×768 px ja 1400×1050 px. Selliseid suuruseid kasutatakse nii 10" kuni 16" ekraanide juures. Suurematel ekraanidel on ekraani resolutsioon ka suurem. (Notebook Screen Resolution, 2012) Lisaks kasutatakse tänapäeval sülearvutite puhul ka väga palju laiekraane, näiteks Asus Eee PC.

Veebitegija peab peamiselt arvestama just ekraani laiusega, sest horisontaalsuuna fikseerimine teeb veebilehe kasutamise mõnevõrra mugavamaks. Kõrgus on sõltuv materjali kogusest, mida sellel veebilehel kuvatakse. Kuigi igasugune kerimine on kasutamise mugavuse seisukohalt halb, on vertikaalsuunas kerimine kasutajate jaoks vastuvõetav ning horisontaalsuunas on see väga ebameeldiv.

Järgnev joonis näitab, kui paljude erinevate ekraanidega peab veebilehe tegija arvestama ainult arvuteid silmas pidades (vt. Joonis 1).



**Joonis 1. Enamkasutusel olevad ekraani suurused 2012. aasta märtsis. (Global Web Stats, 2012)**

Praegusel ajal on suurem osa (19,87%) ekraanidest suurusega 1366×768 px. Seega peaks veebikujundaja arvestama, et enamus kasutajaid vaatavad veebilehti sellise suurusega ekraanidel ning seetõttu peaks veebilehe kujundamisel esmaselt selle resolutsiooniga arvestama. Samas ei tohiks veebilehe kujundamisel ignoreerida ka sellest nii väiksemate kui suuremate ekraanide resolutsioone, kuna ka sellise suurusega ekraane kasutatakse piisavalt palju. Suured ekraaniresolutsioonid on muutunud aina populaarsemateks, kuna nendega on veebilehti mugavam sirvida, kuid ka 1024×768 px resolutsioon ei jää esimesest kohast väga kaugele.

Veebitegijatel soovitan veebiotsingut kasutada, et viimaseid andmeid populaarsetest ekraanisuurustest teada, kuigi aidata võib ka lehekülg: [www.neowin.net](http://www.neowin.net).

## 2.2 Tahvelarvutite ekraanid

Tahvelarvuti (*tablet computer*) on kaasaskantav, mobiilne personaalarvuti, puutetundliku ekraaniga, ja mis kasutab traadita internetiga ühendust. Tahvelarvuti on väiksem kui sülearvuti, kuid suurem nutitelefoni ja enamjaolt kasutakse seda internetis surfamiseks. Tahvelarvuti juhtimiseks kasutakse puutepliatsit, sõrme puudet, virtuaalklaviatuuri või harvemini ka füüsilist klaviatuuri. Tahvelarvutiteks on nimetatud ka selliseid sülearvuteid,

mille ekraani saab ümber pöörata selliselt, et moodustub tahvelarvuti. (Vikipeedia, Tahvelarvuti, 2012) Tuntuim tahvelarvuti on tänapäeval iPad, mida on praeguseks müüdnud üle 14 miljoni eksemplari. (Wikipedia T. F., iPad, 2012)

Kuna tegemist on puuteekraanidega, mis peab veebikujundaja arvestama asjaoluga, et igal kasutajal oleks sõrmega mugav erinevate lehekülgede vahel navigeerida. Ühesõnaga navigatsiooni puhul peab olema piisavalt palju ruumi kahe erineva lingi vahel, et kasutaja ei vajutaks kogemata näiteks kahte nuppu korraga.

Tahvelarvuteid on olemas igasuguses suuruses alates 7" kuni 13"-ni. Selleks, et niigi väikesel ekraanil oleks hea internetis ringi vaadata, on ekraani resolutsioonid erinevatel tahvelarvutitel suhteliselt sarnased. Väikseim levinud resolutsioon on praegu 480×800 px, populaarseim suurus aga 1024×600 px kuni 1280×800 px. Näiteks praegu väga levinud Samsung Galaxy Note 10.1'l on resolutsioon 1280×800 px ning iPad 3-1 ja 4-1 2048×1536 px. (Tablet PC Comparison, 2012) Nendes suurustes – 1280×800 px ja 1024×768 px – on igasuguste tollimõõtudega tahvelarvuteid ja seetõttu peaks veebilehe tegija arvestama nende resolutsioonidega ning mitte väga keskenduma pisematele.

## 2.3 Nutitelefonide ekraanid

Nutitelefoni on mobiiltelefon, mis pakub personaalarvutile sarnaseid kasutusomadusi. Nutitelefoni peetakse mobiiltelefoni, millel on terviklik operatsioonisüsteemi tarkvara ja mis võimaldab standardiseeritud kasutuskeskkonda ning platvormi erinevate rakenduste arendajate tarbeks. (Vikipeedia, Nutifon, 2012) Tänapäeva nutitelefoni on meediamängija, pisike digitaalne kaamera, GPS (*Global Positioning System*) navigatsioon, kõrgresolutsiooniga puutetundlik ekraan, veebilehitseja ning kiire andmeedastus Wi-Fi või mobiilse interneti kaudu. (Wikipedia, Smartphone, 2012) Teisisõnu on nutitelefoni miniatuurne arvuti, millel on lisaks ka telefoni võimekus. (Vikipeedia, Nutifon, 2012)

Nutitelefonide ekraanid varieeruvad nii ekraani suuruste kui resolutsiooni poolest. Enamuse nutitelefonide ekraanide diagonaali pikkus jääb vahemikku 2" kuni 4,5". Eksisteerivad ka mõned 5" ekraani suurusega seadmed, millel töötab mobiilne operatsioonisüsteem ning millega on võimalik telefonikõnesid teha, nagu näiteks Dell

Streak, Samsung Galaxy S III, mille ekraani suurus on 4,8" ja praegune Samsung Galaxy Note, ekraani suurusega 10,1". (Wikipedia, Smartphone, 2012)

Tavalised nutitelefonide ekraani resolutsioonid varieeruvad 240×320 pikslist kuni 720×1280 pikslini, iPhone 5 resolutsioon on 640×1136 px. Paljudel Android'i tippudelitel on ekraani resolutsioon 480×800 px, 540×960 px või 720×1280 ning kuna Android telefonid on ka kõige rohkem kasutusel, siis on need ekraani resolutsioonid ka levinuimad. Näiteks iPhone'idel on ekraani suurus 640×960 px, iPhone 5-l 640×1136, ning Samsung Google Galaxy Nexus'el ja HTC Rezound'il 720×1280 px. (Wikipedia, Smartphone, 2012)

Nutitelefonide puhul peaks kindlasti arvestama asjaoluga, et nende ekraanide suuruse tõttu on veebilehti suhteliselt ebamugav sirvida. Kuna nutitefonid muutuvad aina populaarsemateks, siis peaksid veebilehe tegijad arvestama sellega, et ka nendel pisikestel ekraanidel oleks veebileht väga mugavalt vaadatav ja kasutatav.

## 2.4 Veebilehitsejate resolutsioon pihuseadmetel

Pihuseadmetega, näiteks nutitelefonidega, on suureks probleem see, et veebilehete laius nutitefonides ei olene telefoni ekraani resolutsioonist, vaid telefonides kasutatavate veebilehitsejate resolutsioonist. Ehk siis näiteks kui telefoni ekraani laius on 450 px, siis sama lai veebileht on Android'i veebilehitsejaga vaadates liiga kitsas, kuna veebilehitseja resolutsioon on hoopis 800 px – toimub suuruse teisendus, et 800 piksline veebileht mahuks 450 pikslisele ekraanile, mis muudab loomulikult ka pisut veebilehe kvaliteeti. Ühesõnaga veebilehte kujundades ei ole nii oluline kui lai või kitsas on nutitefon, vaid milline on sellel nutitefonil kasutatava veebilehitseja resolutsioon. Näiteks kui telefoni resolutsioon on 450 px, aga veebilehitseja resolutsioon 800 px, siis surutakse veebilehitseja resolutsioon 450 füüsilise piksli peale kokku (vt. Tabel 1).

Tabel 1. Veebilehitsejate laius erinevate operatsioonisüsteemidega. (Responsive Design, 2011) (Firtman, 2012)

Veebilehitseja	Operatsioonisüsteem	Laius (px)
IE 10 browser	Windows Phone 8	1024
Mobile Safari	iOS	980
Android browser	Android 1-2	800
Android browser	Android 3-4	980
Opera mobile	iOS, Android	850
IE mobile browser	Windows Phone 7	320

Sellest infost lähtuvalt peab autor tõdema, et veebilehe tegijatel tuleb pigem arvestada populaarsemate veebilehitsejate resolutsiooniga, kuna ükskõik, milline resolutsioon on pihuseadmel, veebilehte vaadates loeb ikkagi veebibrauseri laius. Seega peaks veebileht olema selgelt ja mugavalt vaadatav nii 980 pikslise kui ka 320 pikslise resolutsiooniga ning seetõttu tuleks veebilehele mitu kujundust ehk CSS-i teha.

## 2.5 Printimine

Veebilehe printimine on tavaliselt üsnagi tülikas tegevus. Soovitud sisule lisaks printitakse hulk infot, mida pole vaja ning millega pole paberi peal midagi teha. See annab printitule kehva välimuse ning tihtipeale on seda ebamugav lugeda. Varem on mitmetel rohket infot sisaldavatel veebilehtedel pakutud eraldi linki printerisõbraliku (*printer friendly*) versiooni jaoks, kus kõik printimiseks sobimatu on eemaldatud. Nüüd on võimalik printimise jaoks kujundada eraldi CSS. Kuna paljudel veebilehe elementidel pole paberkujul otstarvet – navigatsioon, vormid, teatud kujunduselemendid – siis on oluline määrata, et välja printitaks vaid oluline sisu. Samuti määratakse printitavatele elementidele sobiv stiil - paberile sobilikud kirjastiilid ja värvid. (Vikat, 2010)

Paberi mõõdud erinevad ekraanidest ning printimisel on punktihedus erinev. A4 paberi puhul kasutatakse 595×842 pikslit resolutsiooni. (Lynch & Horton, 2004) Autor toob välja ka teistes mõõtudes paberite jaoks veebilehe resolutsioone (vt.Tabel 2).

**Tabel 2. Veebilehe mõõtmed erinevat mõõtu paberil. (Dimensions Of A Series Paper Sizes)**

<b>Paberi nimi</b>	<b>Paberi suurus (mm)</b>	<b>Veebilehe resolutsioon paberil (px)</b>
<b>A0</b>	1189×841	3370×2384
<b>A1</b>	841×594	2384×1684
<b>A2</b>	594×420	1684×1191
<b>A3</b>	420×297	1191×842
<b>A4</b>	297×210	842×595
<b>A5</b>	210×148	595×420
<b>A6</b>	148×105	420×298
<b>A7</b>	105×74	298×210
<b>A8</b>	74×52	210×147

## 3 HTML 5

HTML 5 teerajaja on alates 2004. aastast olnud WHATWG (*The Web Hypertext Application Technology Working Group*), mis on koostöö põhiliste veebilehitsejate avaldajate vahel. HTML 5 võttis 2007. aastal töö alguspunktina omaks uus W3C töörühm. HTML 5 varasem nimetus oli "*Web Applications 1.0*". HTML 5 uusi elemente on võimalik kasutada aga alates aprillist 2010. (Wikipedia, HTML5, 2012)

HTML 5 on pidevas arengus ja peaks saama W3C soovitusel aastal 2014. HTML 5 pakub arvukalt uusi elemente ehk märgendeid ja atribuute, mis iseloomustavad tänapäeva veebikülgede tüüpilist kasutust. (Wikipedia, HTML5, 2012) Hulk uusi elemente on loodud just eesmärgiga lihtsustada veebilehe struktuuri loomist ja ka selle kujundamist. Mõni uus element on olemuselt sarnane `<div>` ja `<span>` märgenditele, kuid mõni nagu näiteks `<nav>` (veebikülje navigeerimisala) ja `<footer>` (jalus) omavad veebilehel paigutamise tähendust. Selliste märgendite eesmärk on hõlbustada otsingumootori indekseerimist ja veebilehtede näitamist pisikeste ekraanidega seadmete poolt. Muud elemendid lisavad standardiseeritud liidese abil uusi võimalusi, nagu näiteks `<audio>` ja `<video>` elemendid. Välja on jäetud näiteks puhtalt esitluseks kasutatavad elemendid `<center>` ja `<font>`, mille otstarvet saab saavutada kaskaadlaadistikuga. (Wikipedia, HTML5, 2012)

HTML 5 on sihipäraselt loodud sellisena, et ta oleks loetav ka veebilehitsejatega, mis teda ei tunnista. (Munner, 2010) Vanemad veebilehitsejad ignoreerivad uusi elemente, muu osa töötab ikka nagu varem. Mõeldakse üha enam sellele, et asjad toimiksid erinevates olukordades, erineva tarkvara ja seadmetega. HTML 5 arendajad pakuvad lisavõimalusi tänastele veebilehtede autoritele. (Munner, 2010)

### 3.1 Uued HTML elemendid

Veebilehe struktuuri loomiseks on nüüd uued elemendid, mis asendavad vana tuttavat `<div>`-i ning võimaldavad lehe loogilisemat ülesehitust ja paindlikumat kujundust. Nüüd on võimalus kasutada erinevaid elemente, mis jagavad lehekülje loogilisteks osadeks nagu *section*, *article*, *aside*, *hgroup*, *header*, *footer* ja *nav*, mis loovad struktuuri ja hoiavad



kirjutajaid eemale hulgalisest `<div>` kasutamisest. Vajamineva elemendi puudumisel võib selle siiski alati abiks võtta. (Pieters, 2012)

Järgnevalt annab autor lühidalt ülevaate HTML 5 olulisematest uutest elementidest. Neid elemente võib julgesti kasutada kõikidele seadmetele veebilehte kujundades.

Uus element `<section>` esindab üldist dokumenti või rakenduste osa. Seda elementi peaks kasutama juhul, kui kasutaja ei oska või ei soovi mingile osale dokumendile pealkirja panna (vt. Koodinäide 1). Seda ei tohiks kindlasti segamini ajada elemendiga `<div>`. (Lawson, 2009) `<section>` element on mõeldud vaid lehekülje loogilisteks osadeks jagamiseks.

Neid elemente võib ka julgelt kõikide seadmete veebilehe kujunduse puhul kasutada, nad ei sega ka printimise puhul.

```
<section>
  <article>Artikli sisu...</article>
</section>
```

**Koodinäide 1. Näide elementide *section* ja *article* kasutusest.**

Elemendi `<aside>` kasutusvõimalused või –soovitused on palju muutunud. Enne soovitati seda kasutada, viidates seoses olevatele linkidele ja sõnaseletustele. Nüüd on oluline see, kui seda elementi kasutatakse elemendi `<article>` sees, siis peab selle sisu nimelt olema seotud artikliga, kui aga seda kasutatakse väljaspool elemendi `<article>` siis peab elemendi `<aside>` sisu puudutama veebilehte (vt. Koodinäide 2). (Robinson, 2009)

Autor seda küll lõplikult oma näidisveebilehel ei kasutanud, kuid siiski sobib see kasutamiseks kõikidel seadmetel, kuna see annab veebilehel võimaluse viidata vajalikule osale tekstist.

```

<p>Jutt siin on uuest HTML5 keelest, mille W3C peagi
standardiks kuulutab.</p>
<aside>
  <h4>W3C</h4>
  World Wide Web Consortium.
</aside>
<article>
<p>Täpsustav jutt elemendist aside.</p>
  <aside>
    Uue elemendi aside definitsioon.
  </aside>
</article>

```

#### Koodinäide 2. Näide elemendi *aside* kasutusest.

Uut elementi `<hgroup>` kasutatakse pealkirjade ja nende alapealkirjade grupeerimiseks. See element võib grupeerimiseks sisaldada vaid `<h1>`-`<h6>` elemente ning seda võib kasutada alapealkirjade, alternatiivsete pealkirjade või märksõnade puhul. `<hgroup>`-i kasutatakse juhul, kui dokument sisaldab järjest rohkem kui ühte pealkirja (vt. Koodinäide 3). Selle elemendiga grupeerides varjatakse dokumendi liigendusvaatest kõik teised pealkirjad, peale kõrgeima astme oma. `<hgroup>`-i kasutamine parandab vaid HTML-i koodi välimust. (Clark, 2010)

```

<hgroup>
  <h1>Pealkiri</h1>
  <h2>Alapealkiri</h2>
</hgroup>

```

#### Koodinäide 3. Näide elemendi *hgroup* kasutusest.

Päise element `<header>` sisaldab tavaliselt sektsiooni pealkirja (`<h1>`-`<h6>` või `<hgroup>` elemente), kuid see võib sisaldada ka sisukorda, lehekülje logosid või otsinguvõimalusi (vt. Koodinäide 4). (Clark, The header element, 2009) Elementi `<footer>` kasutatakse sarnaselt elementidega `<header>` või `<nav>`. Seda kasutades tuleb aga meeles pidada, et `<footer>`-it ei tohi kasutada sisu sektsioonideks jagamisel ning see ei sobi ka uue rubriigi algusesse. See element peaks sisaldama informatsiooni lehekülje või dokumendi koostaja, teistele dokumentidele viidete, autoriõiguse ja muu sarnase kohta. Elementi `<footer>` võib kasutada veebilehel ka mitu korda, näiteks artikli jaluse ja veebilehe jaluse defineerimiseks (vt. Koodinäide 4). (Osborne, The Footer Element Update, 2009)

Jalust soovivad enamus veebikujundajad oma veebilehe lõppu lisada, seega see element on väga kasulik kasutamiseks kõikide seadmete puhul. Kahjuks siiski varasemad Internet Exploreri versioonid seda veel ei toeta ning alternatiiviks tuleb ikkagi <div> elementi kasutada. Päis lisatakse lehekülje ülaosasse ning sellega on samasugune probleem, et Internet Exploreri varasemad versioonid seda elementi veel ei toeta.

Need mõlemad elemendid on hea võimalus veebilehe päise ja jaluse defineerimiseks ning neid kasutades muutuks HTML-i lugemine lihtsamaks.

```
<header>
  <h1>See on lehekülje nimi.</h1>
  <p>See on lisainfo lehekülje sisu kohta.</p>
</header>

<article>
  Artikli sisu.
  <footer>Artikli jaluse sisu.</footer>
</article>
<p>Lehekülje sisu</p>
<footer>Lehekülje jaluse sisu.</footer>
```

#### **Koodinäide 4. Näide elementide *header* ja *footer* kasutamisest.**

Menüülinkide navigeerimiseks kasutatakse uut elementi <nav>, mis lubab kasutajal viiteid kokku grupeerida ning tulemuseks on rohkem struktureeritud märgendus, mis võib aidata HTML koodi arusaamisel (vt. Koodinäide 5). (Leadbetter, 2009) Kahjuks varasemad Internet Exploreri versioonid seda elementi veel ei toeta ning tuleb leida alternatiiv – näiteks <div> elemendi kasutamine.

Navigatsiooni elementi kasutatakse peamiselt nende veebilehe kujunduste puhul, mida vaadatakse elektrooniliselt – ehk näiteks arvutid või pihuseadmed. Printimise puhul võimaldab see navigatsioonielemendid lehelt eemaldada - <nav> peidetakse.

```
<nav>
  <a href="esimene.html">Esimene</a>
  <a href="teine.html">Teine</a>
  <a href="kolmas.html">Kolmas</a>
</nav>
```

#### **Koodinäide 5. Näide *nav* elemendi kasutusest,**

## 4 CSS 3

CSS3 on praegu alles arendusjärgus, aga on osaliselt kasutatav juba aastast 2008. Suurim muutus CSS 3 puhul on moodulite käibelevõtt. Iga mooduli arendamisega tegeleb erinev töörühm ning moodulid valmivad ja saavad W3C soovitusi teistest sõltumatult. Eeliseks on see, et CSS-i spetsifikatsiooni saab valmis ja kinnitatud tunduvalt kiiremini. See lubab veebilehitsejal toetada spetsifikatsiooni osasid, toetades vaid vajalikke moduleid, näiteks tekstilugeja (tarkvara, mis vaegnägijatele ekraanil paistvat teksti ette loeb) ei pea hõlmama moduleid, mis ainult defineerivad, kuidas elementi visuaalselt näidata. Aga isegi kui see ainult hõlmaks foneetilist moodulit, oleks see siiski CSS 3 standardile vastav. (Kyrnin) Erinevate moodulite tööjärge saab jälgida aadressil: [www.w3.org/Style/CSS/current-working](http://www.w3.org/Style/CSS/current-working) ning seda, millist funktsionaalsust erinevad veebilehitsejad toetavad, on hea vaadata aadressil: [www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp). CSS 3 on alles arendamisel ning see uueneb igapäevaselt.

Kuna CSS3 on alles arendamisel, siis paljude omaduste määramisel tuleb kasutada erinevate veebilehitsejate jaoks spetsiaalseid eesliiteid (*prefix*). Autor märgib ära, milliseid eesliiteid, milliste veebilehitsejatega kasutatakse:

- -ms-\* – Microsoft'i tooted, nagu Internet Explorer;
- -moz-\* – Mozilla Firefox;
- -webkit-\* – Safari 3 ja vanemad ning Chrome;
- -khtml-\* – kuni Safari 2 ning Konqueror;
- -o-\* – Opera.

(List of css vendor prefixes?, 2011)

### 4.1 CSS 3 uued moodulid ja funktsioonid

**Piirjooned** (*borders*) – defineerivad näiteks pildi või tabeli piirjoonte kujunduse. (Sharma, 2009)

- *border-image* – defineerib, kas piirjoonena kasutatakse pilti (vt. Joonis 2). See pildist raami tekitamine lahendab varasema probleemi ilusate raamide ja eriti just nende nurkade kasutamisel. Varem pidi seda hulga *div*-ide abil tegema – iga *div*-i taustaks või sisuks tuli panna erinev pilt.



Joonis 2. Pilt, mida on võimalik *border-image*'na kasutada.

- *border-radius* – defineerib piirjoone ümarnurga suuruse (vt. Joonis 3). Kahjuks ei toimi see moodul vanemate veebilehitsejate puhul, sest see on täiesti uus asi, aga siiski üks esimesi, mis CSS 3 puhul tööle hakkas ja tunduks sai.



Joonis 3. Näited *border-radius*'e kohta.

- *box-shadow* – defineerib kasti varju (vt. Joonis 4). Seda funktsiooni kasutas autor ka näite veebilehel, kuigi see vanade veebilehitsejate puhul, nagu Interneti Explorer 8 või 7, ei toimi.



Joonis 4. Näide *box-shadow* kohta.

**Taustad (*backgrounds*)** – defineerivad veebilehe või veebilehe osa tausta kujunduse. Varem sai veebilehe kujunduseks määrata vaid taustapilti või värvi. Printimise puhul pole soovitatav kasutada, kuna värvilisel taustal on teksti paberilt raske lugeda. (Sharma, 2009)

- *background-origin* – defineerib, millise elemendi küljes taustapilt peaks olema.
- *background-clip* – defineerib tausta värvitud ala.

- *background-size* – defineerib taustapildi suuruse.
- *multiple backgrounds* – võimaldab lisada veebilehele mitu taustapilti, mis on ka üks olulisim uuendus, mida CSS 3 pakub.

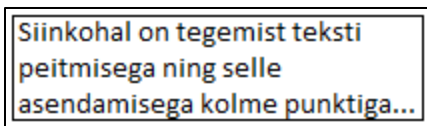
**Tekstiefektid (*text effects*)** – defineerivad teksti kujunduse. (Sharma, 2009)

- *text-shadow* – defineerib teksti varju (vt. Joonis 5). Varem pidi teksti pildina salvestama, et piltitötlusprogrammiga, näiteks Adobe Photoshop'iga, varju lisada.



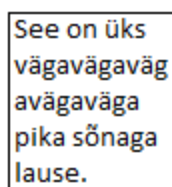
**Joonis 5. Näide *text-shadow* kohta.**

- *text-overflow* – defineerib, mis juhtub, kui tekst läheb üle elemendi serva, mis teda sisaldab. Hea võimalus kasutada kitsastel ekraanidel, et liigne tekst niigi pisikesel ekraanil ei segaks (vt. Joonis 6).



**Joonis 6. Näide *text-overflow* kohta.**

- *word-wrap* – defineerib, kuidas pikka sõna rea lõpus poolitatakse. Samuti hea kasutada kitsastel ekraanidel, kuna alati ei pruugi pikad sõnad nende puhul ühele reale ära mahtuda, kuigi see ei jälgi poolitusreegleid ning ei lisa rea lõppu poolitusmärki (vt. Joonis 7).



**Joonis 7. Näide *word-wrap*'i kohta.**

- *text-justify* – määrab teksti rööpjoonduse.

**Kasutajaliides (*user-interface*)** – võimalused, mida näiteks printimisel ei kasutata, kuid pihuseadmetes on jälle väga mugav kasutada. Siin on mõned kasutajaliidesega seotud funktsioonid, mida autor oma veebilehel ei kasutanud, kuna need toimivad vaid väheste veebilehitsejate puhul: (Sharma, 2009)

- *box-sizing (content-box, border-box)* – defineerib näiteks teksti ümbritseva kasti suuruse või piirjoone jämeduse. CSS 2.1 lisas raamjoone paksuse ja polsterduse (*padding*) ploki suurusele, CSS 3 võimaldab plokkide suuruse arvutamise meetodit muuta – paksus ja polsterdus võidakse arvestada ploki mõõtmete sisse (vt. Koodinäide 6).

```
div {  
box-sizing: border-box  
}
```

**Koodinäide 6. Näide *border-box*'i kohta.**

- *resize* – defineerib, kas kasutaja saab elemendi suurust muuta. See funktsioon toimib hetkel vaid Mozilla Firefox'i, Chrome'i ja Safari puhul.

**Tavalised kasti mudelid (*basic box model*)** – defineerivad kerimisribade olemasolu. Varem ei saanud horisontaalset ja vertikaalselt kerimisriba eraldi määrata. (Sharma, 2009)

- *overflow-x, overflow-y* – defineerivad kerimisribade nähtavuse. Printimise puhul on soovitatav kerimisribad ära peita, kuna paberil pole nendega midagi teha.

**Lisamoodulid (*other modules*)** – defineerivad erinevate funktsioonide, näiteks veergude, kujunduse. Jällegi head uued võimalused veebilehe kujundamisel pihuseadme jaoks. (Sharma, 2009)

- *media queries* – võimaldavad määrata erinevaid kujundusi vastavalt ekraani suurusele või resolutsioonile. Kuna need funktsioonid on käesoleva bakalaureusetöö jaoks olulised, siis nendest lähemalt järgnevates peatükkides.
- *multi-column layout* – võimaldab tekstil mitmesse veergu jaotada defineerides veeru laiuse või veergude arvu. Hea kasutada erineva laiusega ekraanide puhul nii, et laiema ekraaniga on veebilehel näiteks kolm veergu ja kitsama ekraaniga kaks.

- *web fonts* – võimaldab fonte veebi laadida ning peaaegu mistahes fonte veebilehel kasutada. Varem oli kasutatavate fontide hulk piiratud, ehiskirjade kasutamisel tuli sageli tekst pildina veebilehele panna või pidid veebitegijad kasutama kindlaid fonte (*web-safe fonts*), mis iga veebilehitseja ja operatsioonisüsteemi puhul töötasid.

## 4.2 Meediatüübid ja päringud

Alates CSS 2-st on stiililehtede üks tähtsaid omadusi võimalus luua kujundust erinevatele seadmetele – ekraanile, trükisele, kõnesüntesaatorile, pimedate kirjale, pihuseadmetele. (Vikat, 2010) Nii ei pea disainer arvestama ekraanile kuvatava veebilehe samaaegset sobimist näiteks prindiks või programmeerija lisama eraldi printimismiskavasid. Olemas on kümme erinevat CSS meediatüüpi:

- *screen* – mõeldud arvutiekraanidel esitamiseks;
- *print* – lehekülgedeks jaotatud materjali esitamiseks ning printimiseks;
- *handheld* – lehtede kuvamiseks pihuseadmetes (pihuarvutid, mobiiltelefonid), mis on tüüpiliselt väikse ekraani ja aeglase ühendusega;
- *aural* – ekraanilugemise programmi jaoks, mis kasutab audiot teksti lugemisel;
- *braille* – kombitavatel pimedate kirja genereerivatel seadmetel;
- *embossed* – pimedate kirja lehekülgede kaupa trükkivatel seadmetel;
- *projection* – mõeldud projektori kasutamisel, kui veebileht projitseeritakse näiteks seinale;
- *speech* – kõnesüntesaatorid;
- *tty* – ettemääratud suurusega sümbolirudustikud nagu terminalide ning teletaibi ekraanid, kehtib selliste brauserite nagu *lynx* puhul;
- *tv* – mõeldud teleritüüpi seadmetele, millel on kehv resolutsioon (nagu vanad SD-telerid), värvid, piiratud kerimisvõimalused ja heli;
- *all* – kõik meediatüübid.

(Lie, Celik, Glazman, & Kesteren, 2010)



Iga meediatüübi jaoks lingitakse veebilehega eraldi CSS fail (vt. Koodinäide 7). Millise meediatüübi jaoks konkreetne CSS mõeldud on, määratakse atribuudiga „media“, mille erinevad võimalikud väärtused on toodud eespool loetelus.

```
<link rel="stylesheet" type="text/css" href="style.css"
media="handheld" />
```

#### Koodinäide 7. Meediatüübi määramine HTML failis.

Kõikide sama tüüpi seadmete ekraanid ei ole ühesuursed. Sellele on mõelnud ka CSS3 arendajad, kes töötavad *media query* (meediapäring) mooduli kallal. See võimaldab küsida meedia erinevaid parameetreid ning seega, näiteks vastavalt ekraani suurusele, resolutsioonile või seadme kaldenurgale, muuta CSS kujundust. Võimalikud parameetrid:

- *width* – akna laius;
- *height* – akna kõrgus;
- *device-width* – seadme ekraani laius;
- *device-height* – seadme ekraani kõrgus;
- *orientation* – seadme asend;
- *aspect-ratio* – akna külgede suhtarv;
- *device-aspect-ratio* – seadme ekraani suhtarv;
- *color* – seadme värvussügavus, kirjeldab väljundseadme värvikomponendi bittide arvu;
- *color-index* – värvi indeks, kirjeldab väljundseadme värvitabeli kirjete arvu;
- *monochrome* – ühevärvilised;
- *resolution* – resolutsiooni järgi;
- *scan* – mõeldud teleritele ja vastava tehnoloogia kasutamisele (*progressive scan*, *interlace scan*);
- *grid* – *grid*-põhised seadmed, kasutatakse, et saada päringut, kas väljundseade on võrgustik või raster (*bitmap*).

(Lie, Celik, Glazman, & Kesteren, 2010)

Enamus parameetreid võimalik laiendada miinimum ja maksimum parameetritega, nagu näiteks `max-width` või `min-device-height`.

Meediapäring lisatakse meediatüübile, kasutades selleks lihtsaid loogikalauseid. Need on ka seadme ekraani suuruse tuvastamisel abiks. Neid lauseid kasutades saab määrata, et ühte CSS-i kasutaksid näiteks erinevate laiustega ekraanid (saab kirja panna miinimum ja maksimum lehe laiuse). Kasutatavad loogikalauseid on järgmised:

- Kasutada saab ka tuntud loogikaoperaatoreid *and*, *or* ja *not* (vt. Koodinäide 8).

```
<link rel="stylesheet" href="32_screen_999.css"
type="text/css" media="screen and (max-width:999px) or (max-
height:666px) not (min-width:100px)"/>
```

**Koodinäide 8. Näide loogikalauseite *and*, *or* ja *not* kasutamisest.**

- *only* – ainult, lisatakse meediatüüpi ette kui peab olema täidetud ainult valitud tingimus (vt. Koodinäide 9).

```
<link rel="stylesheet" href="32_screen_999.css"
type="text/css" media="only screen and (max-width:999px)"/>
```

**Koodinäide 9. Näide loogikalause *only* kasutamisest.**

(Metshein, CSS3 - Meedia tüübid ja päringud, 2012)

Üldiselt toimib meediatüüpide osa hästi, probleem nende defineerimisel seisneb aga selles, et paljud pihuseadmed või nende operatsioonisüsteemid ei toeta handheld meediatüüpi – need näitavad veebilehti kogemata screen meediatüübina. Sel juhul peaksid veebilehe koostajad arvestama sellega, et ka mõni pisema ekraaniga seade võib näidata veebilehte sellisena, nagu tavalisel arvutiekraanil. Selleks leidis autor lahenduse kasutada meediatüübi asemel parameetreid (täpsemalt ekraani või veebilehitseja laiust), et pihuseadmetele „öelda“, millist lehe kujundust nad peavad arvestama (vt. Koodinäide 10).

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

<link rel="stylesheet" href="mob.css" type="text/css"
media="(max-width:850px)"/>

<link rel="stylesheet" href="suur.css" type="text/css"
media="(min-width:1551px)"/>
```

**Koodinäide 10. Näide, kuidas vältida meediatüübi *handheld* kasutamist.**

# 5 Veebilehe kujundus erinevate seadmete jaoks

Otsesid ettekirjutisi, millised peavad veebilehed kindlasti erinevatel seadmetel välja nägema, ei ole, küll aga on konkreetset tehnilised piirangud, mida tuleb arvestada (näiteks ekraani laius). Iga veebikujundaja saab ise välja mõelda, milline on tema arvates parim lahendus ning milline peaks tema arvates veebileht välja nägema. Kõige tähtsam on seega oma veebileht kujundada lauaarvutite, sülearvutite, tahvelarvutite, nutitelefoni ja printimise jaoks sobivaks ning selleks ongi hea kasutada CSS 3-e, HTML 5-te ja nende võimalusi.

Nagu ka eelmisest peatükist teada sai, tuleb erinevate CSS-ide defineerimiseks meediatüüpe ja –päringuid kasutada. Mõningad CSS omadused on aga mõeldud vaid teatud meedia jaoks – näiteks `page-break-before` kehtib ainult printimise puhul, kus on leheküljed, või jagatud meedia puhul. Samas on mitmeid omadusi, mis on küll kõigil meediatüüpidel olemas, kuid nõuavad erinevaid väärtusi – näiteks `font-size` – ekraanil kasutatakse tavaliselt suuremat kirjasuurust kui paberil. Kõiki omadusi ei saagi iga seadme puhul kasutada ning edasi toobki autor välja soovitusi, kuidas erinevatel seadmetel veebilehte on soovitatav kujundada.

## 5.1 Eriti kõrge resolutsiooniga ekraanid

Praeguseks on ettevõtte Apple tootnud ka eriti kõrge ekraaniresolutsiooniga (*retina display*) arvuteid (vt. Joonis 8). Halb on see, et veebilehitsejad ei oska veel sellise võimalusega arvestada ning paljud neist ei näe seetõttu *retina display*'ga vaadates head välja. Suurimaks probleemiks osutuvad pildid, enamjaolt väikesed .jpg logod veebilehtede päistes või ekraanipildid, mis nende pikslite vähesuse tõttu võivad ebakvaliteetsed välja näha. (Johnson, 2012)

Üldiselt on *retina display* hea, kuna enamjaolt näeb siiski õigesti kujundatud veebileht sellisel ekraanil parem välja (vt. Joonis 8).



Joonis 8. Näide *retina display*-ga (paremal) ekraanist. (Seeing with a critical 'i' - Things that make iPhone 4 great, 2010)

Loomulikult on võimalik kindlustada, et veebileht näeks kindlalt igasuguse resolutsiooniga ekraanil hea välja. *Retina display* puhul tuleks sel juhul jälgida järgmisi reegleid:

- Kasutada tuleks kindlasti *@font-face* kirjastiile. See kindlustab alati, et veebilehe tekst on täpne ning iga seadmega kergesti loetav.
- Kasutada tuleks meediapäringuid. Need aitavad kindla seadme jaoks vajadusel pilti koomale tõmmata.
- Kasutada tuleks CSS *Image Sprite*'i. See lubab võtta kasutusele palju väiksemaid pilte ning kombineerib need üheks suureks pildiks kokku, kasutades *background-position* omadust, et veebilehitsejale öelda, kus milline pilt asuma peab.

(Mansfield, 2012)

Nende lihtsate ja lühikeste soovitusete abil on võimalik kergelt oma veebileht kujundada ka kõrgresolutsiooniga ekraaniga korrektseks.

Autor oma töös *retina display* jaoks veebilehte ei kujundanud, kuna selle testimise võimalus puudus. Sellegipoolest soovib autor tulevikus veebikujundajatel ka kõrgresolutsiooniga ekraanide peale mõtlema hakata, kuna tehnika areneb aina edasi ning võib uskuda, et lähitulevikus on sellised ekraanid veel rohkem levinud.

## 5.2 Printimine

Veebilehte printides tuleks veebilehe sisu paberi veeristega sobivaks kohandada. Juhul kui veebileht on lai, siis võib selle suuruse muutmine nii, et see lehele ära mahuks, muuta teksti loetamatult väikeseks. Oluline polegi printeri resolutsioon, vaid paberi laius.

A4 paberi puhul kasutatakse 595×842 pikslist resolutsiooni – äärtest antakse tavaliselt printitule vaba ruumi ning sellega peaks kindlasti veebilehe kujundaja arvestama. 595 px on lehe laius püstises (*portrait*) asendis veebilehe puhul ning 842 px vertikaalasendis (*landscape*) veebilehe puhul. See vaba ruum oleneb konkreetse printeri seadetest ning seetõttu on väga raske veebilehe laiust printimise jaoks paika panna. A4 paberi puhul soovitatakse siiski kasutada väiksemat laiust kui 560 px. (Lynch & Horton, 2004) Sellisel juhul on garanteeritud, et veebileht näeb printimisel, olenemata printeri seadetest, hea välja. (What is the best width for a fixed-width page layout and why?, 2011)

Printides on mõnikord vajadus ka veebilehte paberil horisontaalasendis (*landscape*) näha, mis siis, et veebilehed on enamasti üles ehitatud püstises (*portrait*) asendis. Püstisel (*portait*) juhul peaks veebilehe laius A4 paberil olema vähem kui 560 px. Kui aga disaineril on vajadus näidata veebilehte teistpidi (*landscape*), siis peaks selle laius A4 paberil olema umbes 800 px (vaata täpsemalt: Tabel 2). Selleks, et veebileht kuvataks *landscape* asendis, tuleks CSS koodi kirjutada paar rida juurde (vt. Koodinäide 11).

```
@page land {size: landscape} või lihtsalt @page {size: landscape}
body {page: land}
```

**Koodinäide 11. Näide, kuidas veebilehte *landscape* asendis näidata.**

Eelnev näide on Opera kohta ja üldjuhul on selliseid lahendused alati veebilehitsejapõhised. Näiteks Safari, Chrome'i ja Mozilla Firefox'i puhul toimub materjali pööramine -90 kraadi võrra (vt. Koodinäide 12).

```
.page {
-webkit-transform: rotate(-90deg);
-moz-transform: rotate(-90deg);
filter:progid:DXImageTransform.Microsoft.BasicImage(rotation=
3) }
```

**Koodinäide 12. Näide, kuidas erinevate veebilehitsejate puhul veebilehekülge *landscape* asendis näidata. (Landscape printing from HTML, 2009)**

Lisaks on võimalus kirjutada HTML-i päisesse, mis asendis näeb veebileht välja erinevate meediatüüpide puhul (vt. Koodinäide 13). Enamjaolt kasutatakse seda printimise puhul, kuid on võimalik ka teiste seadmetega kasutada.

```
media=print and (orientation:landscape)
```

**Koodinäide 13. Näide, kuidas defineerida, et veebileht on erinevate meediatüüpide puhul *landscape* asendis. (Strange offset in the CSS element position after changing orientation on iPad, 2011)**

HTML-i päises võib lehe orientatsiooni ka teisiti defineerida – lehekülje sisu (*@page*) orientatsiooni defineerides (vt. Koodinäide 14).

```
@page landscape {size: LANDSCAPE}
TABLE {PAGE: landscape}
```

**Koodinäide 14. Näide, kuidas defineerida HTML-i päises veebilehe *landscape* asend.**

## 5.3 Taustapilt

Taustapilt võiks üldjuhul tagasihoidlik olla, seda nii arvutite kui ka pihuseadmete puhul. Väga värvilise taustapildi või mitme taustapildi pealt on ebamugav teksti lugeda ning see tõmbab tähelepanu olulisest informatsioonist eemale. Mitu taustapilti näiteks nurgaornamentidena on parem ning pigem kaunistab lehte ega häri silma. Kui veebileht tundub liiga kirjuks minema, siis on muidugi ka hea lahendus taustapilt üldse ära kaotada või see lihtsalt ühtlase värvitooniga asendada.

Paberile printimise korral peaks aga tausta värvi ja taustapildi üldse eemaldama, kuna värvilist teksti on värviliselt taustalt raske lugeda, rääkimata värvilise printeri lisakuludest. Traditsiooniliselt on paberilt lugemiseks kõige mugavam valge taust (vt. Koodinäide 15).

```
body {
background-color:#FFF;
background:url (pildid/taust.jpg);
background-position:top;
background-size:auto
}
```

**Koodinäide 15. Näide tausta värvi ja pildi defineerimisest CSS-is.**

## 5.4 Kirja suurus

Arvutite puhul kasutatakse enamjaolt teksti jaoks kirja suurust 16, mis on ka vaikesead (default). See on piisavalt suur, et lugeda nii väiksel kui ka suuremal ekraanil ning piisavalt väike suures koguses teksti lugemiseks.

Pihuseadmetel veebilehe vaatamiseks peaks kiri olema palju suurem, neil on pisem ekraan ning seega ka teksti halvem näha. Nende puhul võiks kirja suurus olla vähemalt 22 (vt. Koodinäide 16), kui mitte rohkemgi. Menüülingid peaksid loomulikult veelgi suuremad olema – vähemalt 30 – et kasutajal oleks kerge kiiresti erinevate lehekülgede vahel valida.

```
body {
font:22px Trebuchet, "Trebuchet MS", Arial;
color:#FFF
}
```

**Koodinäide 16. Näide veebilehel teksti kirjasuuruse määramisest CSS-is.**

Paberil on aga teksti kõige mugavam lugeda ning seetõttu võib kasutada väiksemat kirja kui ekraanil, traditsiooniliseks kirja suuruseks valitud 12. Tavaliselt on dokumentidel kiri sellise suurusega ning see on osutunud ka kõige õigemaks valikuks.

Pihuseadmete puhul soovib autor kasutada veidi suuremat teksti pealkirjadel – kui arvuti ekraanide puhul kasutas autor *h1* suuruseks 40 px, siis pihuseadmete puhul 60 px. Printimise puhul tundus aga piisav juba 30 pikslise suurusega *h1* pealkiri. Need suurusi soovib autor ka teistel veebikujundajatel kasutada, kuigi pealkirjade suurusi valides tuleb kindlasti arvestada veebilehe laiusga, et pealkirja tekst „üle ääre“ ei läheks.



## 5.5 Kirja stiil ja kaunistused

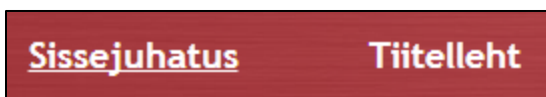
Arvutite puhul on kirjastiilidel vaid üks oluline nõue – loetavus. Üldiselt on ükskõik, millisel kujundatud või ilustatud näiteks veebilehe pealkiri on peaasi, et see loetav oleks.

Pihuseadmetel on aga liigselt kaunistatud teksti veelgi raskem lugeda, kuna ekraan on juba niigi väike. Seega nutitelefonide või tahvelarvutite puhul oleks soovitatav teksti mitte üleliia kaunistada – näiteks kalligraafilised (*calligraphic*) fondid. Printimise puhul võib veidike vabamalt ümber käia, kuna paberilt on niikuinii mugavam teksti lugeda kui arvuti ekraanilt.

Üldiselt soovitab autor kasutada siiski *web safe* fonte, kuna neid on lihtne ja kindel kasutada, need on testitud ning ei ole üleliigsete kaunistustega.

## 5.6 Navigatsioonivahendid

Navigeerimiseks kasutatakse enamasti tekstilinke, mis võivad ka menüüks koondatud olla. Menüülingid võiksid olla arvutite, pihuseadmete ja paberi puhul erinevad. Veebilehede kujundamisel arvutitele võiks menüülinkidel olla hiirega üleliikumise (*hover*) efekt ning aktiivsel menüülingil mingi erinevus, näiteks allajoonitus (*underline*), mis on ka standardnõue mitte ainult aktiivsetele linkidele, vaid ka tekstilinkide puhul, et kasutaja saaks kiiresti aru, millisel lehel ta viibib (vt. Joonis 9). Hoolimata reeglitest kõik lingid alla joonida, võiks tekstilingi ja menüülingi veebilehel eristamiseks need erinevalt kujundada – näiteks ühe teha teist värvi või rasvaselt (*bold*).



Joonis 9. Aktiivse menüülingi, mis on allajoonitud, näide.

Pihuseadmetel hiirega üleliikumise efekti pole vaja, kuna nende puhul hiirt ei kasutata. Küll aga võiks kasutaja siiski kuidagi aru saada, millisel lehel kasutaja viibib. Autor soovitab selle jaoks näiteks igale lehele pealkiri panna vastavalt lehe sisule – see oleks isegi soovitatav kõikide seadmete puhul. Samuti peaksid menüülingid olema niivõrd suured, et igaüks saab kergelt neile ka oma põidlaga pihta. Autor valis enda veebilehele menüülinkide suurused katse-eksitusmeetodi teel.

Juhul kui menüüks kasutatakse ikoone või pilte, siis nende suuruse valikuks on olemas ja praktilisi soovitusi:

- HTC toodetel (Arrive, Desire, Evo, Inspire, Sensation, Thunderbolt) soovitatakse kasutada nupu suurusi 60×60 px kuni 72×72 px.
- Apple iPhone 3G puhul soovitatakse nupu suuruseks kasutada 45×45 px, iPhone 4G puhul kasutada 91×01 px ja iPadi puhul soovitatakse nupu suurust 37×37 px.
- Motorola toodete (Defy, Droid, Droid Incredible) puhul soovitatakse nupu suurusena kasutada 70×70 px kuni 74×74 px.
- Samsungi toodete (Infuse, Galaxy S, Galaxy S II) puhul soovitatakse nupu suurusi 58×58 px kuni 65×65 px.
- Tahvelarvutitel, mille ekraani suurus on 7 tolli, soovitatakse miinimum nupu suuruseks 30×30 px. (Wallace, 2011)

(Adam, 2012)

Printimise puhul peaks menüülingid täiesti ära kaotama, kuna paberil pole menüüst mingit kasu, see võtab asjatult ruumi ning raiskab paberi pinda ning samuti tuleks tekstilingid rahulikumaks muuta ja neilt igasuguse kaunistuse (näiteks allajoonituse) eemaldada, see muudab lugemise ladusamaks (vt. Koodinäide 17).

```
nav {  
display:none;  
text-decoration:none  
}
```

#### **Koodinäide 17. Näide menüü peitmisest ja teksti kaunistuste ära võtmisest CSS-is.**

Autor kasutas pihuseadmetel navigaationivahendi linkideks miinimumina 220 pikslist laiust ning veel pisemate (kuni 320 px laiusega veebilehitsejad) seadmete puhul 188 pikslist laiust, kuna ekraan on sellisel juhul juba liiga väike, et laiemaid menüülinke kasutada.

## **5.7 Pildid**

Veebilehte vaadates on tihtipeale kasutatud illustreerimiseks pilte. Arvutis pole see tavaliselt probleemiks, kuna ekraan on suur ja internetiühendus piisavalt kiire.

Pihuseadmetel soovitatakse ebaolulised pildid ära keelata, kuna need võtavad mõttetult ruumi ja viivad kasutaja tähelepanu tekstist eemale(vt. Koodinäide 18).

```
.pilt {  
visibility:hidden  
}
```

**Koodinäide 18. Näide, kuidas ebaolulisi pilte CSS-is ära keelata.**

Juhul kui veebitegija soovib pihuseadmetel ka pildid alles jätta, siis soovitab autor need pisemaks teha, olenevalt veebilehitseja resolutsioonist, et pildid ekraanile ära mahuksid ning liialt palju tähelepanu endale ei tõmbaks. Liiga suure pildi korral, kui see enda alla terve ekraani võtab, peab kasutaja jällegi kerima, et tekstini jõuda.

Autor pani näiteks pihuseadmete puhul pildi laiuseks 150 px, kui arvutiekraanide puhul on pildi laius 220 px. Pildi laius peab kõigi seadmete puhul kindlasti veebilehe laiusest väiksem olema, et korrektne välja näeks ja mugav sirvida oleks.

Printimisel pole oluline pilte peita ega ka nende suurust muuta, kuna need pole lehe peal ebamugavad sirvida.

## 5.8 Videod ja heliklipid

Uus HTML 5 võimaldab ka videote ja heliklippide lihtsat lisamist veebilehele. Arvutites veebilehti vaadates teeb see võimalus veebilehed huvitavamaks ja külgetõmbavamaks.

Pihuseadmetes aga ei pruugi need alati töötada, näiteks vajaliku pistikprogrammi (plug-in) või koodeki (codec) puudumise tõttu, seega soovitab autor need välja jätta (vt. Koodinäide 19) või lisada mitu erinevat sama sisuga heliklippi, millest siis vastavalt meediatüübile sobivat näidatakse. Android toetab järgmiseid meediatüüpe: aac, flac, mp3, midi, pcm/wave, vorbis, avc, mpeg (Android Supported Media Formats, 2012) ning iPhone ja iPad toetavad järgmiseid meediatüüpe: aac, aiff, mp3, wav, audible, apple lossless. (Koyan, kuupäev puudub)

```
video {  
display: none  
}  
audio {  
display: none  
}
```

**Koodinäide 19. Näide veebilehel videote ja heliklippide peitmisest CSS-is.**

Printimisel tuleks video- ja heliklipid ära jätta, kuna paberil pole nendega midagi peale hakata.

## 6 Autori kogemus veebilehte kujundades

Näidisveebilehe loomisel võttis autor eesmärgiks kujundada leht, mis näeb korrektne välja ning on mugav kasutada erinevatel seadmetel. Kõigepealt tuli välja mõelda, milliseid seadmeid võimalik sihtgrupp kasutab, millele vastavaid kujundusi luua ning see on ka üks esimesi asju, mida iga autor uurima peab. Ehk siis tuli välja mõelda mitu erinevat CSS-i tuleb kirjutada ning kui suurte resolutsioonidega ekraanidele iga CSS mõeldud on. Siinkohal koostas autor kõigepealt kujunduse ühe kõige tavalisema ja enam kasutusel olevatele ekraani resolutsioonidele – lausega 1024 ja 1366 px. Veebilehe lauseks sai võetud 900 px, kuna pisemate ekraanide puhul tuli niikuinii lisa CSS teha, ning HTML-i päises sai ära märgitud, et see CSS läheb kasutusele arvutiekraanide või nende seadmete puhul, mille laius on 851-1500 px. Erinevaid laiuseid katsetades tundus see vahemik kõige mõistlikum, et veebileht mugav kasutada oleks. See oli kujundamise kõige lihtsam osa.

Järgmisena tuli autoril valida, millise resolutsiooniga pihuseadmetele teha järgmist kujundust. Kuna autoril endal on nutitelefoni (Sony Ericsson Xperia Ray), siis sai see peamiseks testimise seadmeks. Kõigepealt tegi autor veebilehe sellise lausega, nagu nutitelefoni ekraaniresolutsioon (450 px). Siin aga tekkisid probleemid, kuna sellises lauses veebileht nägi nutitelefoni ekraanil millegipärast väga pisike välja. Pärast pikka uurimist avastas autor, et tegelikult tuleks arvestada hoopis veebilehitseja resolutsiooniga, mitte telefoni ekraani omaga (vt. Tabel 1). Selle teadmisega sai CSS endale lõpuks õige lause paika – 780 px – ning HTML-i päises sai defineeritud, et see kujundus läheb kasutusele kuni 851 px lausega veebilehitsejate puhul. Neid laiuseid, millest siin peatükis juttu, soovib autor kasutada ka teistel veebikujundajatel, kuna need osutusid kõige paremateks valikuteks erinevate seadmete puhul. Olgugi, et Android 4 puhul on veebilehitseja laius suurem kui 850 px, otsustas autor lause sellise võtta, kuna muidu näeks veebileht tavaekraanidel ebakorrektna välja.

Probleemne koht oli seda CSS-i kirjutades ka menüü, mille tekst tuli teha piisavalt suureks, et ka kohmakad sõrmed linkidele pihta saaksid (lingi suuruseks sai 45 px). Samuti soovis autor, et menüülingid oleksid pihuseadmetes veebilehe keskel ühes tulbas. Menüü perfektselt sättimiseks läks küll aega, kuid otseselt see keeruline ei olnud.

Edasi otsustas autor, et ülipisikestele ekraaniresolutsioonidele tuleb ka omaette CSS kujundada. Seega tuli valida veebilehele CSS-is jälle uus laius, milleks sai 300 px. Selle kujunduse puhul tuli menüülinkide kirja suurust vähemaks võtta 36 pikslile, et ekraanile üldse muu tekst ära mahuks. Ülejäänud tekst tuli aga veidi suuremaks teha, et veebilehte mugavam lugeda oleks. See CSS sai HTML-i päises defineeritud kuni 320 px laiusega (et äärtesse veidi tühja ruumi jääks) veebilehitsejatele.

Keerulisemad osad valmis, tuli kontrollida, kas veebileht ka laiekraan monitori puhul korrektne välja näeb. Seal tundus aga veebileht kuidagi liiga kokkupressitud ning autor otsustas koostada veel ühe CSS-i eriti suurte ekraaniresolutsioonide – 1551 px või suuremad – jaoks. Nende puhul sai veebilehe laiuks julgelt 1500 px võetud. Teksti ja menüülinkide suurus jäi sellel kujundusel samaks, mis tavaliste arvutiekraanide puhul. Seda kujundust oli autoril kõige lihtsalt koostada, kuna kõige esimeses CSS-is oli kõik juba põhimõtteliselt ette ära tehtud.

Viimaseks jäi teha vaid CSS kujundus printimise jaoks. Veebilehe laiuks sai võetud soovitude järgi vähem kui 560 px (täpsemalt 540 px). Teksti tegi autor väiksemaks – 12 px – ning menüüelemendid said peidetud. Samuti sai tausta värv ja pilt ning igasugused hiirega üleliikumise efektid ära võetud. Ühesõnaga autor tegi printimise jaoks kujunduse just nii, nagu viendas peatükis soovitatud.

Veebilehe sisu valis autor selle järgi, et kuna tegemist on bakalaureusetöoga ja bakalaureusetöö vormistusjuhend tuleb niikuinii läbi lugeda, siis tundus sobiv seda ka oma näite veebilehel kajastada.

Veebilehte on võimalik näha aadressil: [http://www.tlu.ee/~kuut1990/Asjad/.Baka\\_veeb/](http://www.tlu.ee/~kuut1990/Asjad/.Baka_veeb/)

# Kokkuvõte

Käesoleva bakalaureusetöö eesmärkideks oli anda ülevaade erinevate seadmete jaoks vajaliku kujunduse loomise vahenditest, nende võimaluste testimine ning veebiloojatele soovitude andmine, kasutades uut CSS 3-e ja HTML 5-te. Autor andis kirjanduse põhjal lühikese ülevaate HTML 5-st, CSS 3-st ning erinevate seadmete ajaloost. Kirjeldas, millised seadmed on olemas ning millised ekraani resolutsioonid populaarsemad on, et veebilehe tegijad saaksid nende suurustega arvestada. Samuti tutvustas autor mida uut töid HTML 5 ja CSS 3, mida nad võimaldavad ning mis on kõige mõistlikum viis veebilehe kujunduseks, et see oleks praeguse aja seadmetel loetav, kuna kokkuvõtlikku materjali erinevatele seadmetele veebilehe tegemiseks pole.

Üheks eesmärgiks oli veel koostada veebileht, mis näeks korrektne välja nii suurtel monitoridel, tavalistel sülearvutitel, tahvelarvutitel, nutitelefonides kui ka välja printimisel. Autor katsetas erinevate seadmete jaoks erinevaid CSS kujundusi, et leida sobivaimat.

Autor õppis seda bakalaureusetööd koostades väga palju uut erinevatest veebi kujundamis võimalustest ning oskab edaspidi soovitada teistele paremaid lahendusi, et veebileht korrektne välja näeks ja mugav kasutada oleks. Kuigi sellist veebilehte on väga keeruline luua ning ette jäävad igasugused erinevad probleemid, siis autor peab tõdema, et kindlasti on võimalik koostada sellist veebilehte, mis suurel enamusel seadmetest korrektne oleks. Autori eesmärgid said täidetud ning nendel seadmetel, millega ta veebilehte kontrollida sai, nägi see korrektne välja.

See bakalaureusetöö võiks olla abiks neile, kes soovivad tutvuda uute HTML 5 ja CSS 3-ga ning alles alustavad veebilehe loomist. Loomulikult on võimalik seda bakalaureusetööd veel edasi spetsiifilisemaks arendada, eriti kuna nii HTML 5 kui ka CSS 3 on alles arendamisel ning muutusi tuleb kogu aeg juurde. Autor usub, et sellest tööst võib abi olla mitte ainult algajatele, vaid ka kogenud veebikujundajatele.

# Summary

The purpose of this bachelor's thesis is to give an overview of website design for different devices and testing these possibilities while using new CSS 3 and HTML 5. Author also wanted to offer suggestions how should a website look like to be correct and comfortable to use with different devices such as usual computers, tablet computers, smartphones and in print.

Moreover, this work introduces the devices that are in use nowadays and which screen resolutions are the most popular among them, so that web designers can take that information into consideration. Author also made a web page example to test different designs on different devices and to figure out how should a web page look like on them to be almost the best.

In the first chapter, the author gives an overview of the history of CSS 3 and HTML 5 and how has the web design developed along with important devices. There we, for example, read up on the first tablet computer or smartphone, called Simon.

The second chapter is divided into sections. There is information about different devices that will be important for the web page design. Mostly, the author talks about screen resolutions generally and specifically about the resolutions that are most common or becoming more and more popular.

In the third and fourth chapter, the author gives an overview of HTML 5 and CSS 3. There is information about what new possibilities they enable and how or where to use them. In the fourth chapter, there is information about media types and their usage and also some examples and suggestions. This is probably the most important chapter of the bachelor's thesis, as that's where it's defined how to tell a web page which CSS design to use.

The fifth chapter is also very important. In this part, the author gives recommendations on how should different elements look like on different devices. For example, how big or small should the text size or navigation pane be. In the sixth chapter, the author talks how her web page evolved and which suggestions did she take from the fifth chapter.

In conclusion, the author learned a lot during writing this bachelor's thesis and has a lot more knowledge about designing a web page for different devices. She gave a lot of



suggestions to other web designers how to make their web page better. She is confident to give others advice about finding better solutions, so that their web page would look correct and be comfortable to use. Author believes that this work could be helpful for all the web designers who want some answers about the appearance of the websites on different devices.

# Kasutatud Kirjandus

Adam. (23. Veebruar 2012. a.). *Minimum touchable area on mobile devices*. Allikas: PugetWorks: <http://pugetworks.com/blog/2012/02/minimum-touchable-area-on-mobile-devices/>

*Android Supported Media Formats*. (22. Detsember 2012. a.). Allikas: developers: <https://developer.android.com/guide/appendix/media-formats.html#core>

Clark, R. (16. Juuni 2009. a.). *The header element*. Allikas: html5 doctor: <http://html5doctor.com/the-header-element/>

Clark, R. (23. Märts 2010. a.). *The hgroup element*. Allikas: html5 Doctor: <http://html5doctor.com/the-hgroup-element/>

*Dimensions Of A Series Paper Sizes*. Allikas: Paper Sizes: <http://www.papersizes.org/a-paper-sizes.htm>

Firtman, M. (19. Oktoober 2012. a.). *Why All Web Devs Need To Prepare Their Sites for IE10 and Windows 8*. Allikas: DZone: <http://css.dzone.com/articles/ie10-meets-modern-mobile-html5>

Fisher, T. (2012). *Monitor*. Allikas: About.com PC Support: [http://pcsupport.about.com/od/componentprofiles/p/p\\_monitor.htm](http://pcsupport.about.com/od/componentprofiles/p/p_monitor.htm)

Gade, L. (28. Oktoober 2003. a.). *PalmOne Treo 600 Palm OS Smartphone from Sprint PCS*. Allikas: MobileTechReview: [http://www.mobiletechreview.com/treo\\_600.htm](http://www.mobiletechreview.com/treo_600.htm)

*Global Web Stats*. (30. Detsember 2012. a.). Allikas: W3Counter: <http://www.w3counter.com/globalstats.php>

Johnson, J. (9. Juuli 2012. a.). *Designing on a Retina Screen: My Thoughts on the Retina MacBook Pro*. Allikas: design shack: <http://designshack.net/articles/graphics/designing-on-a-retina-screen-my-thoughts-on-the-retina-macbook-pro/>

- Koyan, M. *What Audio Format Can the iPhone Play?* Allikas: SALON:  
<http://techtips.salon.com/audio-format-can-iphone-play-10160.html>
- Kyrnin, J. *What is CSS3?* Allikas: About.com Web Design/HTML:  
<http://webdesign.about.com/od/css3/a/aa061206.htm>
- Landscape printing from HTML.* (8. September 2009. a.). Allikas: Stackoverflow:  
<http://stackoverflow.com/questions/138422/landscape-printing-from-html>
- Lawson, B. (11. September 2009. a.). *The section element.* Allikas: html5 Doctor:  
<http://html5doctor.com/the-section-element/>
- Leadbetter, T. (15. Juuli 2009. a.). *Semantic navigation with the nav element.* Allikas:  
 html5 Doctor: <http://html5doctor.com/nav-element/>
- Lie, H. W., Celik, T., Glazman, D., & Kesteren, A. v. (27. Juuli 2010. a.). *Media Queries.*  
 Allikas: W3C: <http://www.w3.org/TR/css3-mediaqueries/>
- List of css vendor prefixes?* (23. Märts 2011. a.). Allikas: stackoverflow:  
<http://stackoverflow.com/questions/5411026/list-of-css-vendor-prefixes>
- Lynch, & Horton. (5. Märts 2004. a.). *Web Style Guide.* Allikas: Web Style Guide:  
<http://webstyleguide.com/wsg2/page/dimensions.html>
- Mansfield, J. (22. Märts 2012. a.). *Designing for the Hi-Res iPad Retina Display.* Allikas:  
 Brand Aid Design Company: <http://www.brandaidesignco.com/blog/designing-for-hi-res-ipad-retina-display/320/>
- Metshein, M. (21. Mai 2012. a.). *CSS3 - Meedia tüübid ja päringud.* Allikas: Metshein:  
<http://metshein.com/content/view/808/26/>
- Mis on HTML.* (19. Oktoober 2007. a.). Allikas: Mis on HTML:  
<http://skaut.planet.ee/html4/html4tund1/html4tund1.htm>
- Munner, M. (6. Mai 2010. a.). *HTML 5, mis see on?* Allikas: interneti.info:  
<http://www.interneti.info/artiklid/html5/>
- Notebook Screen Resolution.* (2012). Allikas: ScreenTek:  
<http://www.screentekinc.com/resource-center/resolution.shtml>

- Osborne, J. (29. September 2009. a.). *The Footer Element Update*. Allikas: html5 Doctor:  
<http://html5doctor.com/the-footer-element-update/>
- Pieters, S. (5. Oktoober 2012. a.). *HTML5 differences from HTML4*. Allikas: W3C:  
<http://dev.w3.org/html5/html4-differences/>
- Reed, B. (18. Juuni 2010. a.). *A Brief History of Smartphones*. Allikas: PCWorld:  
[http://www.pcworld.com/article/199243/a\\_brief\\_history\\_of\\_smartphones.html](http://www.pcworld.com/article/199243/a_brief_history_of_smartphones.html)
- Responsive Design*. (24. August 2011. a.). Allikas: Stanford University:  
<http://itservices.stanford.edu/service/web/mobile/developers/responsivedesign>
- Ritso, J. (2011). *Veebilooja uuend võimalused CSS3 abil*. Tallinn.
- Robinson, M. (28. Oktoober 2009. a.). *Aside Revisited*. Allikas: html5 Doctor:  
<http://html5doctor.com/aside-revisited/>
- Seeing with a critical 'i' - Things that make iPhone 4 great*. (12. Juuni 2010. a.). Allikas:  
GizmoWatch: <http://www.gizmowatch.com/entry/seeing-with-a-critical-i-things-that-make-iphone-4-great/>
- Sharma, R. (9. Veebruar 2009. a.). *Difference between CSS2 and CSS3*. Allikas:  
Tutorialfeed: <http://www.tutorialfeed.org/2009/02/difference-between-css2-and-css3.html>
- Strange offset in the CSS element position after changing orientation on iPad*. (24. November 2011. a.). Allikas: Stackoverflow:  
<http://stackoverflow.com/questions/8254774/strange-offset-in-the-css-element-position-after-changing-orientation-on-ipad>
- Tablet PC Comparison*. (November 2012. a.). Allikas: Tablet PC Comparison:  
<http://www.tabletpccomparison.net/>
- Taurus, K. *Rastergraafika*. Allikas: Arvutigraafika liigid:  
<http://www.cs.ut.ee/~kerstit/gimp/arvutigraafika.html>
- The First Smartphone*. (27. Märts 2012. a.). Allikas: All About Screens and Pixels:  
<http://allaboutscreenz.blogspot.com/2012/03/first-smartphone.html>

- Types Of Monitors.* (22. Oktoober 2011. a.). Allikas: HubPages: <http://hassam.hubpages.com/hub/Types-Of-Monitors#>
- Wallace, M. (12. November 2011. a.). *Touchable area of Screen Sizes.* Allikas: Information Architect: <http://mwallace.info/2011/11/touchable-area-of-screen-sizes/>
- Well, You May Not Know My Name Is Simon.* (23. Aprill 2012. a.). Allikas: Things Wide Shut: <http://www.thighswideshut.org/tag/ibm-simon/>
- What is the best width for a fixed-width page layout and why?* (2011). Allikas: Answers: [http://wiki.answers.com/Q/What\\_is\\_the\\_best\\_width\\_for\\_a\\_fixed-width\\_page\\_layout\\_and\\_why](http://wiki.answers.com/Q/What_is_the_best_width_for_a_fixed-width_page_layout_and_why)
- Who invented the Internet?* (2012). Allikas: Computer Hope: <http://www.computerhope.com/issues/ch001016.htm>
- Vikat, M. (2010). *CSS raamistiku kasutamise veebilehe kujunduse loomisel.* Tartu.
- Wikipedia, t. F. (18. Aprill 2012. a.). *Cascading Style Sheets.* Allikas: Wikipedia, the Free Encyclopedia: [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- Wikipedia, t. F. (12. Aprill 2012. a.). *HTML.* Allikas: Wikipedia, the Free Encyclopedia: <http://en.wikipedia.org/wiki/HTML>
- Wikipedia, t. F. (24. Aprill 2012. a.). *HTML5.* Allikas: Wikipedia, the Free Encyclopedia: <http://en.wikipedia.org/wiki/HTML5>
- Wikipedia, t. F. (16. Märts 2012. a.). *HTML5.* Allikas: Wikipedia, the Free Encyclopedia: <http://et.wikipedia.org/wiki/HTML5>
- Wikipedia, T. F. (23. Detsember 2012. a.). *iPad.* Allikas: Wikipedia, The Free Encyclopedia: <http://en.wikipedia.org/wiki/IPad>
- Wikipedia, t. F. (18. Märts 2012. a.). *Nokia 7110.* Allikas: Wikipedia, the Free Encyclopedia: [http://en.wikipedia.org/wiki/Nokia\\_7110](http://en.wikipedia.org/wiki/Nokia_7110)
- Wikipedia, t. F. (7. Detsember 2012. a.). *Smartphone.* Allikas: Wikipedia, the Free Encyclopedia: <http://en.wikipedia.org/wiki/Smartphone>

- Wikipedia, t. F. (15. September 2012. a.). *Tab (GUI)*. Allikas: Wikipedia, the Free Encyclopedia: [http://en.wikipedia.org/wiki/Tab\\_\(GUI\)](http://en.wikipedia.org/wiki/Tab_(GUI))
- Wikipedia, t. F. (23. Oktoober 2012. a.). *Treo 600*. Allikas: Wikipedia, the Free Encyclopedia: [http://en.wikipedia.org/wiki/Treo\\_600](http://en.wikipedia.org/wiki/Treo_600)
- Wikipedia, t. F. (7. November 2012. a.). *World Wide Web*. Allikas: Wikipedia, the Free Encyclopedia: [http://en.wikipedia.org/wiki/World\\_Wide\\_Web](http://en.wikipedia.org/wiki/World_Wide_Web)
- Vikipeedia, V. E. (3. Aprill 2012. a.). *Kuvar*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/Kuvar>
- Vikipeedia, V. E. (17. Jaanuar 2012. a.). *Lahutusvõime*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/Lahutusv%C3%B5ime>
- Vikipeedia, V. E. (27. Märts 2012. a.). *Nutifon*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/Nutifon>
- Vikipeedia, V. E. (27. Märts 2012. a.). *Plasmateled*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/Plasmateled>
- Vikipeedia, V. E. (2. Aprill 2012. a.). *Sülearvuti*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/S%C3%BClearvuti#Ekraan>
- Vikipeedia, V. E. (15. Märts 2012. a.). *Tahvelarvuti*. Allikas: Vikipeedia, Vaba Entsüklopeedia: <http://et.wikipedia.org/wiki/Tahvelarvuti>
- Vikipeedia, V. E. (20. Veebruar 2012. a.). *Traadita rakenduste protokoll*. Allikas: Vikipeedia, Vaba Entsüklopeedia: [http://et.wikipedia.org/wiki/Traadita\\_rakenduste\\_protokoll](http://et.wikipedia.org/wiki/Traadita_rakenduste_protokoll)

# Lisa 1

Autori poolt loodud veebilehe lähtekood:

## Index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>HTML 5 näiteleht Bakalaureusetöök</title>
  <link rel="stylesheet" href="stiil.css" type="text/css"
media="screen"/> Iga screen tüüpi seadmed võtavad selle CSS-ikülge.
  <link rel="stylesheet" href="mob.css" type="text/css"
media="(max-width:850px)"/> Pihuseadmed, maksimaalse veebilehitseja
laiusega 850 px, võtavad selle CSS-i külge.
  <link rel="stylesheet" href="pisi.css" type="text/css"
media="(max-width:320px)"/> Pihuseadmed, maksimaalse veebilehitseja
laiusega 320 px, võtavad selle CSS-i külge.
  <link rel="stylesheet" href="suur.css" type="text/css"
media="(min-width:1551px)"/> Suurte resolutsioonidega monitorid, mille
minimaalne laius on 1551 px, võtavad selle CSS-ikülge.
  <link rel="stylesheet" href="print.css" type="text/css"
media="print"/> Veebilehte printides võtab HTML selle CSS-ikülge.
</head>
<body>
<div class="sisu">
  <header> Uue päiseelemendi kasutamine.
  <h1>Bakalaureusetöö vormistusjuhend</h1>
  </header>
<nav>
```

`<div class="menyy">` Kuna vanad Internet Exploreri versioonid `<nav>` elementi ei toeta, siis tuli kasutusele võtta ka `<div>` element.

```
    <a href="index.html" class="aktiivne"><span style="padding-left:30px; padding-right:30px">Sissejuhatus</span></a>
```

Selle elemendi küljes on ka klass nimega „aktiivne“, et aktiivset navigatsioonilinki teistest eristada.

```
    <a href="tiitel.html"><span style="padding-left:30px; padding-right:30px">Tiitelleht</span></a>
```

```
    <a href="nouded.html"><span style="padding-left:30px; padding-right:30px">Nõuded</span></a>
```

```
    <a href="joonised.html"><span style="padding-left:30px; padding-right:30px">Joonised</span></a>
```

```
    <a href="viited.html"><span style="padding-left:30px; padding-right:30px">Viitekirjed</span></a>
```

```
    <a href="kirjandus.html" class="yks"><span style="padding-left:30px; padding-right:30px">Kirjandus</span></a>
```

```
</div>
```

```
</nav>
```

```
<div class="tekst">
```

```
<section> Uue elemendi <section> kasutamine.
```

```
    <h2>Sissejuhatus</h2>
```

```
    <p>Käesoleva dokumendi eesmärgiks on anda soovitusi ja tutvustada reegleid, mille alusel üliõpilane saab vormindada seminaritööd ja bakalaureusetööd. Käesolevas dokumendis ei kirjeldata tehnilisi võtteid, mis peavad olema selgeks saanud Arvuti töövahendina aineist. Samuti ei kirjeldata töö sisulisi aspekte.</p>
```

```
    <p>Üliõpilastöö kohustuslikeks osadeks on (toodud järjekorras) :</p>
```

```
    <ul>
```

```
    <li>tiitelleht;</li>
```

```
    <li>autorideklaratsioon;</li>
```

```
    <li>sisukord;</li>
```



- <li>sissejuhatus;</li>
- <li>töö põhiosa;</li>
- <li>kokkuvõte;</li>
- <li>kasutatud kirjanduse loetelu;</li>
- <li>võõrkeelne resüme (võõrkeelsel tööl eestikeelne).</li>

<p>Tööl võivad olla lisad, lühendite loetelu ja võõrkeelsete terminite loetelu, mis paigutatakse kasutatud kirjanduse järele.</p>

<p>Esitamiseks tuleb töö korralikult köita. Bakalaureusetööle sobivad: liimköide ja rullköide; seminaritööde juures võib kasutada ka pehme kaanelisi augustamisega kiirköitjad. Köitmiseks ei sobi kõvakaanelised kiirköitjad, augustamiseta kiirköitjad, kirjaklambrid, kiletaskud, jms.</p>

<p>Esitada tuleb üks töö eksemplar paber kandjal, millele on lisatud CD või DVD töö terviktekstiga, rakenduse puhul ka programmi kood. Digitaalselt esitatud failid peavad olema üldkasutatavates formaatides. Sobivad vormingud on OpenDocument (odt/ods/odp), tavatekst ja keerulisemate väljundvormingute korral PDF. MS Office dokumendivormingute puhul on soovitatav kasutada vanemaid versioone (doc/xls/ppt).</p>

<p>Tööle lisatud CD või DVD plaadid jms tuleb kinnitada töö külge nii, et nad töö vaatamisel kindlalt töö küljes püsiksid (nt kasutades paberist või kilest ümbrikku). CD-le või DVD-le tuleb vähemalt kirjutada oma nimi ja töö peakiri.</p>

<p>Käesolev juhend pakub kujunduse poolest välja ühe võimaluse töö vormindamiseks. Siiski ei tuleks seda pidada ainsaks võimaluseks ja näiteks lõikude vahesid joonlauaga mõõtma hakata. Järgmist vajavad suurused on juhendi tekstis eraldi välja toodud.</p>

```

</section>
</div>
<footer> Uue jaluselemendi kasutamine, millega on sama lugu, mis navigatsiooniga, et
kuna vanemad Internet Exploreri veebilehitsejad seda ei toeta, pidi autor kasutama ka
<div> elementi.
<div class="jalus">
  <h3>Kontakt:</h3>
  Helen Kuut<br/>
  <u>e-post</u>:
  <address><a href="mailto:kuut1990@tlu.ee">
kuut1990@tlu.ee</a>
  </address>
  <u>msn</u>:
  <address><a href="mailto:helen.kuut@gmail.com">
helen.kuut@gmail.com</a>
  </address>
  <u>skype</u>: helen.kuut<br/>
  <u>telefon</u>: +372 56214418<br/>
</div>
</footer>

</div>
</body>
</html> HTML-i lõpp.

```

## Stiil.css

```

html {
  overflow-y:scroll; Defineerib kerimisribade olemasolu.
  overflow-x:auto}
body {
  font:16px Trebuchet, "Trebuchet MS", Arial;
  color:#FFF;

```

```

background-color:#FFF;
background:url (pildid/taust.jpg);
background-position:top;
background-size:auto;
margin:10px;
cursor:default}
nav {
font:18px Trebuchet, "Trebuchet MS", Arial;
width:900px; Navigatsioonimenüü laiuse defineerimine.
margin-top:25px;
text-align:center} Defineerib teksti keskjoonduse.
.menyy { Kuna varasemad Internet Exploreri versioonid <nav> elementi ei toeta,
kasutas autor ka <div> elementi, kus defineeris menüü klassi.
font:18px Trebuchet, "Trebuchet MS", Arial;
width:900px;
margin-top:25px;
text-align:center}
footer {
width:840px; Jaluse laiuse defineerimine.
margin:20px;
text-align:center}
.jalus { Kuna varasemad Internet Exploreri versioonid <footer> elementi ei toeta,
kasutas autor ka <div> elementi, kus defineeris jaluse klassi.
width:840px;
margin:20px;
text-align:center}
.sisu {
border-radius:15px;
box-shadow: 10px 10px 5px #888; Defineerib sisukasti taustavarju.
-moz-box-shadow: 10px 10px 5px #888; Defineerib sisukasti
taustavarju Mozilla Firefox'i puhul.
position:absolute;
width:900px;
left:50%;

```

```

margin:0 0 0 -450px;
min-height:400px;
background-color:rgb(176, 23, 31);
background-color:rgba(176, 23, 31, 0.7);
-ms-background-filter:"alpha(opacity=70)";           Defineerib
tausapildi läbipaistvuse Interneti Exploreri puhul.
-moz-background-opacity:0.7;   Defineerib taustapildi läbipaistvuse
Mozilla Firefox'i puhul.
-khtml-background-opacity:0.7} Defineerib taustapildi läbipaistvuse
Konquerori ja kuni Safari 2 puhul.
header { Uue päiseelemendi kasutamine.
width:900px} Defineerib päise laiuse.
.tekst {
width:860px; Defineerib tekstiosa laiuse.
margin:20px;
text-align:justify; Defineerib teksti rööpjoonduse.
-webkit-column-count:2; Defineerib, mitut tulpka kasutatakse Safari 3 ja
vanemate ning Chrome'i puhul
-moz-column-count:2; Defineerib, mitut tulpka kasutatakse Mozilla Firefox'i
puhul.
column-count:2}
table, th, td { Defineerib tabeli kujunduse.
border: 1px solid black;
width:280px}
table {
border-collapse:collapse} Defineerib, kas tabelil on ümber kast.
@font-face {
font-family: "Esimene font"; Defineerib kasutatava fondi nime ning
järgmised erinevad faililaiendid kindlustavad, et see font oleks kasutatav võimalikult
paljude veebilehitsejatega.
src: url("Barracuda.ttf");
src: url("Barracuda.eot");

```

src: local("☺"), Vajalik kasutamiseks, et veebilehitsejad ei defineeri ette antud fonti kogemata mõneks teiseks fondiks.

```
url("Barracuda.woff") format("woff"),
url("Barracuda.otf") format("opentype"),
url("Barracuda.svg#filename") format("svg");}
```

```
h1, h2, h3, h4, h5, h6, p, ul {margin:0; padding:10px 0 10px 0}
```

```
h1 {font-size:40px; text-shadow: 0.2em 0.2em #000; text-align:center} Defineerib elemendi <h1> kujunduse ning autor lisas ka teksti varju funktsiooni.
```

```
h2 {font-size:20px; color:#FFF; font-family: "Esimene font", Trebuchet, "Trebuchet MS", Arial; text-align:left} Defineerib, kui on tegemist elemendiga <h2>, siis kasutatakse kõigepealt eespool defineeritud fonti ning kui veebilehitseja seda ei toeta, siis kasutatakse järgnevat võimalust.
```

```
h3 {font-size:17px; color:#FFF}
```

```
p {text-align:justify}
```

```
ul {margin-left:20px}
```

```
a:link {text-decoration:none; color:#FFF; font-weight:bold}
```

```
a:visited {text-decoration:none; color:#FFF; font-weight:bold}
```

```
a:hover {text-decoration:underline; color:#FFF; font-weight:bold} Defineerib, et navigeerimislingist hiirega üle „sõites“ kaunistatakse teksti allajoondusega.
```

```
a:active {text-decoration:underline; color:#FFF; font-weight:bold} Defineerib, et aktiivsel navigeerimislingil oleks tekst allajoondusega.
```

Autor lisas selle <div> elemendina ka, et funktsioon toimiks kõikide veebilehitsejatega.

```
a.aktiivne:link{text-decoration:underline; color:#FFF; font-weight:bold}
```