

Henri Backlund

Typografian digitalisaatio

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

11.4.2017

Tekijä Otsikko	Henri Backlund Typografian digitalisaatio
Sivumäärä Aika	26 sivua 11.4.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaaja	Tutkijaopettaja Merja Bouters
<p>Insinööriyön tarkoituksena oli selvittää kirjoituksen kehityskulkua nuolenpääkirjoituksesta aina modernien tietokoneiden ilmestymiseen asti, ja sitä kuinka teknologiset edistysaskeleet ovat vaikuttaneet siihen. Teknologian merkitys typografian kehityksessä oli insinööriyön keskiössä, ja sitä tutkittiin perehtymällä merkittävimpiin painettuun ja digitaaliseen tekstiin vaikuttaneisiin keksintöihin.</p> <p>Kehitysympäristöt voidaan jakaa kolmeen osaan: mekaaninen ja sähköinen kirjapaino, käyttäjärjestelmätypografia ja verkkofontit. Kaksi viimeistä muodostavat yhdessä käsitteen digitaalinen typografia.</p> <p>Typografian käsitteet joutuvat muutoksen alaisuuteen uudessa kehitysympäristössä. Muutoksen täytyy tapahtua julkisessa tilassa, jossa käytäntöjen toimivuus voidaan helposti todentaa kulutustottumuksia seuraavan datan avulla. Verkkoselaimessa näkyvä teksti sisällyttää koko typografian kehityskaaren yhteen paikkaan. Kaikkien maailman kielten ja merkkien saatavuus sekä muokkaustyökalujen lähes rajaton mahdollistavuus voivat luoda jotain aivan uutta, joka on kuitenkin suurimmalta osin vanhaa.</p> <p>Insinööriyötä voidaan käyttää apuna verkkokehitystyössä, sillä siinä on esitetty erikoisfonttien toimintaperiaate ja sen käyttämisessä esiintyviä ongelmatilanteita.</p>	
Avainsanat	Typografia, fontti, css

Author Title	Henri Backlund Digitalization of typography
Number of Pages Date	26 pages 11 April 2017
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Merja Bauters, Researching Lecturer
<p>The topic of the thesis is to examine how writing has evolved from the days of Cuneiform script to the modern day and to clarify the importance of technological breakthroughs amongst it. The core of the thesis is in untangling the technological perspective from the human effect and declaring the encompassing scheme on how they interact.</p> <p>The development of typography is investigated in three stages: mechanical- and electrical printing, computer typography and web fonts. Two of the latter comprise the notion of digital typography.</p> <p>Concepts of typography are tested within the ever changing digital frame. Consumer habits are also affected by the declaration of new technological concepts. The public domain has to be the place for change and where the new concepts can be viewed and verified through behavioral studies. A text that is viewed with a browser comprises the entire trajectory of typography. Availability of all languages and symbols within the digital frame can only produce something new that has already been done in the past.</p> <p>For the web developers the thesis enlightens the basic principles on how to implement tailored fonts to web page and troubleshoots the most common unstable circumstances.</p>	
Keywords	Typography, font, css

Sisälllys

Lyhenteet

1	Johdanto	1
2	Typografia	2
2.1	Kirjoituksen kehittyminen	2
2.2	Kirjapaino	3
2.3	Elektroniset painokoneet	6
2.4	Digitaalinen tekstinkäsittely	7
2.5	Mittayksiköt	9
3	Web-fonttien teknologiset ratkaisut	10
3.1	Unicode	10
3.2	Fonttiformaatit	12
4	Web-standardien asettamat vaatimukset	18
4.1	Muuntuva fontti	19
4.2	Erikoisfontin käyttö	21
5	Yhteenveto	22
	Lähteet	24

Lyhenteet

ACR	Abstract character repertoire
CCS	Coded character set
CEF	Character encoding form
CES	Character encoding scheme
TES	Transfer encoding syntax
UTF-8	Unicode transformation format
ASCII	American standard code for information interchange
RIP	Raster image processor
FUnit	TrueType font unit
CVT	Control value table
TTF	TrueType font
EOT	Embedded OpenType
WOFF	Web open font format
CSS	Cascading style sheet
HTML	Hypertext markup language
DOM	Document object model
CSSOM	Cascading style sheet object model

1 Johdanto

Kirjoituksen kehittyminen on alkanut inhimillisestä tarpeesta muistaa asioita, joilla on keskeinen merkitys arkielämässä selviytymisen kannalta. Muistin merkitys on lajikehityksen keskiössä, ja se on kulkenut saman evoluutiopolun kirjoitustaidon kanssa. Puhe- ja kirjoitustaito on tuhansien vuosien kehitystyön tuotos, joten on hyvä ymmärtää muutosten hitaus, sillä pitkään kypsyneitä yleisesti hyväksytyjä käytänteitä ei hetkessä voida muuttaa. Kaikkea kirjoitustaitoon ja sen ilmenemiseen liittyvää on alettu kuvata sanalla typografia. Typografiaa voidaan käsitellä tieteellisestä ja taiteellisesta näkökulmasta. Molempiin näkökulmiin tulee sisällyttää humanisevoluutiivinen ulottuvuus.

Insinööriyössä selvitetään typografian kehittymistä aina ensimmäisistä savitauluhin kaiverretuista kirjoitusmerkeistä moderneihin tietokoneisiin asti. Näkökulmakseni olen valinnut teknologiaa seuraavan kehityspolun, ja pyrin osoittamaan, kuinka insinööri aina keksii ratkaisun inhimilliseen tarpeeseen. Toinen keskeinen osa on uuden kehittyminen vanhan päälle, ja se havainnollistuu jo keskiajalla kehitetyssä kirjapainossa, jonka yhteydessä kehitettyjä sivun jäsennysmittoja hyödynnetään edelleen digitaalisessa tekstinkäsittelyssä.

Aikaa alkuvuosista nykypäivään asti voidaan pitää typografian kehittymisen merkittävimpanä ajanjaksona. Teknologisten edistysaskeleiden myötä on syntynyt ajanjaksoja, joissa graafikoille on syntynyt uusi mahdollisuuksien ympäristö toteuttaa taiteellisuuttaan. Toteutusympäristön jatkuva uusiutuminen jatkuu edelleen tietokoneissa ja mobiililaitteissa. Tietokoneiden merkittävimpiä saavutuksia kirjoittamisen edistämisen kannalta ovat tehokkuus, tarkkuus ja kapasiteetti. Typografian kannalta on ymmärrettävä perinteiden merkitys, johon sisältyy vuosisatojen aikana kehittynyt lukutottumus. Digitaalisessa ympäristössä kirjainleikkausten muokausmahdollisuudet ovat lähes rajattomat. Fonttsuunnittelijan tulee ensisijaisesti varmistaa kirjainleikkauksen toimivuus yleisen lukutottumuksen ympäristössä, sillä hyvä fontti ei kiinnitä keskivertolukijan huomiota.

Kehitys ei kuitenkaan ole aina mennyt eteenpäin kirjoituksen esittämisen suhteen. Näyttöteknologian tarkkuus alkaa uusimpien teräväpiirtonäyttöjen yleistymisen myötä tarjota fonttikehittäjille varteenotettavia variaatiomahdollisuuksia suunnittelutyössä.

Kehittyneimmät painokoneet pystyvät edelleen tuottamaan monta kertaa tarkempaa painojälkeä. Näytöissä tarkkuus liittyy tietysti myös niiden fyysiseen kokoon pikselilukumäärän lisäksi. Painetusta tekstistä siirryttiin tietokoneiden näytölle ja siellä viimeisin kehitysympäristö on verkkoselain. Selain pakottaa fonttikehittäjät taas uuteen ympäristöön, jossa on omat muuttujansa. Selaimessa typografian mielekkyys ei enää rajoitu visuaaliseen puoleen, vaan mukaan on tullut datakapasiteetin muuttuja. Mobiililaitella tapahtuva nopea verkkoselailu on pakottanut kehittäjät optimoimaan sivuston latausajan minimiin, jossa fonttiedosto on osa kokonaisuutta.

2 Typografia

2.1 Kirjoituksen kehittyminen

Kirjoitusta on esiintynyt merkkien muodossa jo kauan ennen nykyisten aakkosten keksimistä. Sumerilaisessa korkeakulttuurissa kehitettiin ensimmäinen verojärjestelmä. Verotuksessa kerättyjen varojen seuraaminen synnytti tarpeen taltioida varainvaihtoon liittyvät tapahtumat myöhempää tarkastelua varten. Kulttuurissa kehitettiin nuolenpääkirjoitus, jonka kirjoitusmerkit painettiin savesta valmistettuihin tauluihin. Aluksi kuvat käsittivät vain konkreettisia asioita kaupankäyntiin tai verotukseen liittyen. Vähitellen ne alkoivat kuvata myös vertauskuvauksellisia asioita, esimerkiksi kuva kädestä saattoi ilmentää verbiä ”antaa”. Kuvien ketjuttamisesta syntyi idea sanakirjoituksesta, jota egyptiläiset kehittivät vieläkin pidemmälle. (1.)

Aakkosia ei tule ymmärtää ylivertaisena edistysaskeleena kirjoittamisen evoluutiossa, vaan yhtenä puheen toisintaa edistävänä keksintönä. Eri siviiliisaatioissa kehitetyt kirjoitusmerkit tai kuvat eivät olleet universaaleja, eli kuvista tai merkeistä johdetut äänteet ja merkitykset eivät korreloineet keskenään. Samankaltaisia kuviin ja merkkeihin perustuvia kirjoituksia on löydetty Kiinasta, Mesopotamiasta ja Egyptistä. Ensimmäiset aakkokset on kirjoitettu Egyptissä yli neljä tuhatta vuotta sitten. Nämä aakkokset perustuivat konsonantteihin, jossa vokaaleja ei kirjoitettu ollenkaan. Tämä johtuu egyptin kielen sanarakenteesta, jonka äänteet korostavat konsonantteja. Vokaalien sisällyttäminen kirjoitukseen tulee esiin kreikkalaisessa kulttuurissa vuoden 750 eaa. paikkeilla. Kreikkalaiset paransivat foinikialaisilta omaksumiaan merkkejä parantamalla niiden muotoa. Merkit saivat tasapainoisemman ulkomuodon geometristen

perusmuotojen avulla. Toinen merkittävä saavutus oli vasemmalta oikealle etenevän kirjoitussuunnan vakiinnuttaminen.

Etruskit omaksuivat kreikkalaisten aakkoset, ja heidän aikanaan aakkosia oli 26. Italiaa asuttaneiden etruskien jälkeen kirjoitusmerkkien kehitystyö siirtyi roomalaisille, jotka kehittivät latinalaisen aakkoston lähelle nykyistä muotoaan noin 200 vuotta eaa. Roomalaisten kirjaimet jalostuivat myöhemmin kapitaaliseen asuun, ja niitä käytettiin muunmuassa julkisten rakennusten julkisivuissa, triumfikaarissa ja hautakivissä. Kirjoitusvälineiden kehittyminen edesauttoi kirjoituksen kehittymistä kapitaaliaakkosista pienaakkosiin. Arkipäiväistynyt nopeatahtinen kirjoitustyyli ruokokynällä oli mahdollista vain käyttämällä pieniä ja yksinkertaisia kirjaimia, joten juhlavat taltalla veistetyt kapitaaliaakkoset täydentyivät pienillä kirjaimilla. Euroopassa 300–800-luvulla kehittyi lukuisia kirjoittamisen muotoja, kuten unsiaalit ja puoliunsiaalit. Pienaakkosten lopullinen muoto oli nähtävissä jo 800-luvulla kehitetyissä minuskeleissa. Tasaterällä 30 asteen kulmassa kirjoitetuissa minuskeleissa oli nähtävissä paksuusvaihteluita, ja niiden tyyli siirtyi myöhemmin humanististen painokirjainten muotopiirteeksi. Minuskelikirjoituksessa sanat erotettiin sanaväleillä ja virkkeet aloitettiin suuraakkosilla. Myöhemmin 1400-luvulla käytössä oli erilaisia goottilaisia kirjakkeita, ja ne korvattiin humanistianttiikvoilla, kun kirjapaino sai alkunsa. (2, s. 25–26; 3, s 83.)

2.2 Kirjapaino

Kirjoituksen koostaminen Gutenbergin menetelmällä perustuu kirjaimiin, jotka on valettu lyijyseoksesta valmistettuihin muotteihin. Muotit asetetaan riveihin tekstirivin muodostamista varten, jolloin muotin päässä kohokuviona esiintyvä kirjainmerkki kastetaan musteeseen ja painetaan paperia vasten. Jokaisen muotin leveys vastaa siihen valetun kirjaimen leveyttä, ja sanaväli saadaan aikaan lyhyemmällä kirjaisimella, joka ei kosketa mustetta. Myös sisennykset ja kappaleiden loppuun jäävät tyhjät välit pystyttiin toteuttamaan käyttämällä erikokoisia kohokuviottomia kirjaisimia. Digitaalisen tekstinkäsittelyn taustalla toimiva järjestelmä noudattaa samaa menetelmää, jossa kirjaimet ja tyhjät välit on mitoitettu pikseleihin. (4, s. 3–4.)

Digitaalisessa tekstinkäsittelyssä kirjaimet on siis sidottu pisteisiin, joiden ääriviivat ovat yhdenmukaisia Gutenbergin metallikirjaisimien painojäljen kanssa. Piste ja metallimuotin painopinta ovat käytännössä sama asia: ne määrittelevät painetun kirjainmusteen

jättämän jäljen ja painopinnan ympärillä jäävän tyhjän alueen eli marginaalin (5, s. 27). Marginaalin koko täytyy valita tarkoin, jotta saadaan visuaalisesti yhdenmukainen ilme. Graafinen tekstinkäsittely on ollut jo pitkälle kehittynyttä ennen Gutenbergin aikaa. Kalligrafia eli kaunokirjoitus (5, s. 21), ja sen evoluutio, asetti omat vaatimuksensa myös kirjapainolle. Ladonnassa täytyi huomioida tarkkaan kalligrafiassa kehittynyt kirjoituksen malli, joka muodosti oman standardinsa tekstin ulkoasulle. Tämä tarkoitti myös sitä, että kirjainmerkkien mittasuhteet tulee valita tekstin koon mukaan. Pienellä kirjainkokoalla ladotun tekstin lukukelpoisuus parantuu, kun kirjaimet on painettu hieman tavanomaista leveämmällä ja lihavoidummalla kirjaisimella. Kirjainmerkin suurempi koko mahdollistaa niiden hienovaraisempien säätöjen toteuttamisen, eli ne voivat olla ohuempia ja koristeellisempia. Kirjainmuotojen koristelulla ei ole tekstisivun yleisilmeen kannalta merkitystä silloin, kun käytössä on monenkokoisia kirjaimia. Gutenbergin kehittämää kirjapainoa seurasi muutaman vuosisadan jälkeen uusien keksintöjen esiinmarssi. Kirjapainon seuraava vaihe oli automatisaatio, jonka ensimmäisiä keksintöjä olivat kirjoitus- ja latomakone. (4, s. 5.)

Remington tunnetaan kirjoituskoneen ensimmäisenä sarjatuotannon aloittaneena valmistajana, jonka tuotanto käynnistyi 1870-luvulla. Kirjoituskoneen keskeisin osa on sivuttain liikkuva kirjoituspää. Jokainen painallus liikuttaa kirjoituspäätä vaakasuunnassa ja luo vakiomittaisen tyhjän tilan kirjainten väliin. Kirjoituspää saattoi kuitenkin jäädä jumiin, jos kirjoitus eteni mekaanista liikettä nopeammin. Tämän välttämiseksi qwerty-näppäinjärjestelmä oli paras ratkaisu ongelmaan. Tietokoneen qwerty-mallisen näppäimistön vakiintumisella ei siis ole mitään tekemistä kirjoitusergonomian tai tehokkuuden kanssa (6). Kirjoituspään horisontaalasti suorittama liike on tuttu myös sähkökirjoituskoneista, joissa painopää liikkuu sivuttain pystysuuntaan nähden. Sama idea on tietokoneiden tekstinkäsittelysovelluksissa: jokainen painallus liikuttaa kohdistinta eteenpäin suhteessa kirjainmerkin kokoon ja tyyliin. Tietokoneen tekstinkäsittelyohjelmassa lyönnin varaamaa liikettä voidaan asetusten avulla säätää halutun mittaiseksi. Ensimmäisessä kirjoituskoneessa tätä mahdollisuutta ei kuitenkaan ollut, vaan liike oli vakiomittainen riippumatta kirjaintyylistä tai -koosta.

Eurooppalaiset kielet perustuvat suurimmalta osin Latinalaisiin aakkosiin, joiden ominaispiirteisiin kuuluu vaihteleva leveys ja monimuotoisuus. Yksi typografian keskeisistä haasteista muodostuu juuri tästä perustasta: merkkien monimuotoisuus on kehittynyt pitkä ajan kuluessa pelkästään visuaalisten ehtojen mukaan. Tämän vuoksi päädyttiin suunnittelemaan kirjaintyyli niin sanotusti tekniikka edellä eli kehitettiin

kirjaintyyli, joka oli yhteensopiva kirjoituskoneen mekanismin kanssa. Tästä sai alkunsa *monospace*- kirjaintyyli. (4, s. 8.)

Courier-kirjaintyyli kuuluu *monospace*-luokkaan eli sen kirjaimiston mittasuhteet ovat aina yhteneviä toistensa kanssa, riippumatta niiden koosta. Näin ollen ne ovat yhteensopivia kirjoituskoneen mekanismin kanssa, joka käyttää vakiomittaista kirjainväliä. Nykyäänkin *monospace*-kirjaintyyliä on olemassa sopivia käyttökohteita, joissa kirjainten yksinkertainen esitysmuoto luetaan eduksi. Useimmat kirjaintyyliä käsittelevät numeroita *monospace*-tyylin mukaisesti. Se tekee laskutoimitukset ja taulukoinnin helpommaksi niiden asettuessa vertikaalisti samalle tasolle. Kehitys pakotti *monospace*in umpikajaan, ja laitevalmistajat alkoivat kehittää uudenlaista konetta. Ensimmäiset ladontamekanismit jakoivat kirjaimiston viiteen erikokoiseen leveysluokkaan. Monimuotoisten kirjaintyylien esittäminen vain viidellä eri leveydellä olisi kuitenkin rikkonut niiden mittasuhteet, joten laitevalmistajat päätyivät nostamaan leveysien lukumäärää. (7, s. 123.)

Monotype-latomakone on automatisoitu kirjoituskone, joka valaa tekstirivin kirjain kerrallaan. Kone käsittää kaksi osaa: näppäimistön ja kirjoituspään. Jokainen näppäinlyönti rei'itetään nauhaan, jota käytetään kirjoituspään ohjaamiseen. Paineistettua ilmaa puhalletaan rei'itettyjen kohtien lävitse, jotta kirjoituspää osaa valaa oikeat kirjaimet. Nauhaan tallentuu kahdenlaista informaatiota: mikä merkki oli kyseessä ja kuinka leveitä kirjainvälien tulee olla. Leveydet on jaettu 18 eri kokoon, mikä oli merkittävä edistysaskel edeltäjiinsä verrattuna. Samoihin aikoihin kehitettiin myös toinen automatisoitu ladontakone, joka oli nimeltään *Linotype*. Sen periaatteena oli koostaa tekstiä rivi kerrallaan, ja myös sitä ohjattiin näppäimistön avulla.

Kivipainomenetelmä eli litografia alkoi yleistyä painomediassa 1900-luvun puolivälissä, sillä sen kustannukset olivat merkittävästi edullisemmat muihin painometelmiin verrattuna. Menetelmä on samankaltainen valokuvan kehityksen kanssa, jossa sivun kuva kastetaan emulsioon levitettyä painolevyä vasten. Offset-litografian yleistyminen johti lopulta valoladonnan kehittämiseen. (7, s. 124.)

2.3 Elektroniset painokoneet

Toisen maailmansodan jälkeen painoteollisuudessa siirryttiin uuteen aikakauteen. Käsipainomenetelmät muuttuivat vanhanaikaisiksi, kun valoladontakoneet alkoivat yleistyä painoteollisuudessa. Painoteollisuuden tarpeisiin kehitettiin kahdenlaisia valoladontakoneita: näytöllä varustettuja, joita käytetään isojen otsikoiden luomiseen, ja näppäimistön avulla toimivia, joilla leipäteksti kirjoitetaan. Näytölliset koneet toimivat siten, että valo johdetaan linssin läpi merkkikokoelman sisältävälle filminegatiiville ja siitä toisen linssin läpi valokuvapaperille. Valokuvapaperi sijaitsee valokuvaemulsiota sisältävässä altaassa, jossa valon läpi suodatetut merkit kehittyvät. Ensimmäiset laitteet olivat mekaanisesti toimivia ja painotilanteessa koneenkäyttäjä joutuu yksitellen pyörittämään filminegatiivista esiin tarvittavat merkit. Merkin kohdalla painetaan valaisunappia, ja merkki alkaa kehittyä paperille. Koneenkäyttäjä pystyy tarkkailemaan kehittyvää paperia samalla, kun uusia kirjaimia työstetään. Kirjainväliä pystytään säätämään kehitystyön aikana, ja se oli merkittävä kehitysaskel edeltäviin painomenetelmiin verrattuna. Lisäksi alkuperäisfontin kokoa pystyttiin tarvittaessa säätämään molempiin suuntiin. Tämä oli selkeä etu metallipalikoihin sidottuun fonttikokoon verrattuna. Filminegatiivia pystytään myös uusiokäyttämään lähes loputtomasti. Edistykselliset säätömahdollisuudet ja painotyön nopeutuminen vakiinnuttivat nopeasti näyttökoneiden aseman otsikoiden koostamisessa. (7, s. 123.)

Näppäimistöllä varustetut valoladontakoneet esiteltiin 1950-luvulla. Niitä on kahdenlaisia: optisia ja kuvaskannauslaitteita. Laitteet ovat käytännössä samanlaisia, ja niiden ero liittyy paperin valaisemistekniikkaan. Varhaisimpia painokoneita ohjattiin rei'itetyn paperin avulla, joka sisälsi näppäimistön lyönnit. Myöhemmin ne korvattiin magneettinauhoilla ja sen jälkeen levykkeillä. Optisen järjestelmän merkistö on tallennettu filmikiekolle, ja sitä operoidaan näppäimistön avulla. Kiekkopyörä kiertää sisällä yli tuhat kertaa minuutissa, ja tietokone ohjaa linssiä, jonka läpi merkkejä valotetaan. Tietokoneistumisen myötä mittasuhteita pystyttiin säätämään tarkemmin kuin koskaan aikaisemmin, ja painotyöstä oli tullut merkittävästi kustannustehokkaampaa. Uuden sukupolven elektroniset painokoneet korvasivat kuvaskannauslaitteet. Elektronisten painokoneiden merkkidata on tallennettu tietokoneen muistiin, josta se projektoidaan katodiputkinäytölle ja edelleen linssin läpi valokuvapaperille. Elektronisen tallennusmahdollisuuden myötä näytöiltä pystyttiin ottamaan ruutukaappauksia, mikä mahdollisti saman tekstipaletin uusiokäytön.

Fyysisten painomenetelmistä eriytyminen jatkui edelleen, kun tietokoneiden yleistyminen aloitti digitaalisen tekstinkäsittelyn aikakauden. (7, s. 125–127.)

2.4 Digitaalinen tekstinkäsittely

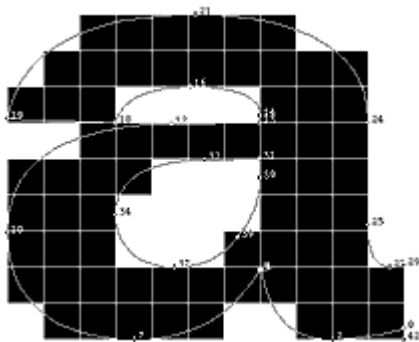
Typografinen ammattitaito on laaja-alainen käsite. Toteutuskehyksen muuttuminen perinteikkäästä käsityötaidosta digitaalisen aikakauden määrittelemään suuntaan tulee ymmärtää typografian monimuotoisuuden ilmentämisen uutena mahdollisuutena, sillä hyvä typografia on enemmän tiedettä kuin taidetta (8, s. 53). Typografian perinteen täytyy kuitenkin säilyä ennallaan fyysisen toteutusmuodon siirtyessä digitaaliseen ympäristöön, sillä lopputuotteen täytyy edelleen vastata yleiseen lukutottumuskäsitykseen. Digitaalinen ympäristö tarjoaa lähes rajattomat mahdollisuudet tekstimuotoilulle, ja sitä seurannut muuntelu on osittain johtanut hyvien vanhojen sääntöjen unohtamiseen. Uusien fonttityylien runsas variointi on toisinaan johtanut esteettisyyden ja luettavuuden laadun heikkenemiseen. Suunnittelijat voivat luoda uudenlaisia yhdistelmiä valtavan valikoiman ja muokkaamisen helppouden tarjoamassa ympäristössä. Perinteikäs pysty- ja vaakatilaan jaettu typografia on näyttötekniikan kehittymisen myötä muuttunut kolmiulotteiseksi, ja tilailluusiosta on tullut osa typografiaa. (8, s. 87.)

Mekaanisten painokirjasimien digitaalinen mallintaminen on vaatinut vanhojen määritelmien päivittämistä. Glyyfi eli kirjoitusmerkin esitysmuoto on sidottu pikseliin, jonka korkeutta ei ole määritetty, ja leveys on sidottu kirjaimen kokoon. Jokaiselle merkille on asetettu peruslinja, jonka päälle niiden ääriiviivat asettuvat. Joidenkin merkkien kaaret ulottuvat kuitenkin hienokseltaan tämän linjan yli. Glyyfikaaren ja pikselin alarajan risteämiskohtaa kutsutaan glyyfin alkupisteeksi. Matemaattisessa esitysmuodossa glyyfin alkupiste on sidottu koordinaatteihin. Alkion leveys määritellään siirtymävektorin avulla, jonka alku- ja loppupiste yhdistää kahden peräkkäisen pikselin alkupisteet. Glyyfien ja pikselireunan väliin jää valkoinen tila, joka typografiassa määritellään sommittelun elementiksi. (5, s. 11; 8, s. 15.)

Yksi sivunkuvantamiseen suunnitellun *PostScript*-ohjelmointikielen merkittävimpiä ominaisuuksia on laiteriippumattomuus. Sitä voidaan käyttää millä tahansa laitteella ja käyttöjärjestelmällä. *PostScript* käsittelee tekstin skaalautuvina vektoreina, joita käyttäjä voi muuttaa haluttuun kokoon. Vektoreita edeltävänä aikana tietokoneiden graafinen

käyttöjärjestelmä käsitteli kuvat ja tekstin bittikartan avulla, jossa jokaiseen kuva-alkioon kohdistettu bitti käsittää yhden kuvapisteen. Vektorina eli matemaattisena ääriviivaesityksenä kuvattu kirjain piirtyy paperille rasterikuvan tavoin. PostScriptillä kirjoitetun sivun muuntaminen tulostettavaan muotoon oli monimutkainen prosessi, ja sen käsittelemiseen tarvittiin erillinen tietokone. Tätä kutsutaan tiedoston muuntoprosessoriksi, englanniksi *raster image processor (RIP)* (5, s. 35, 38). Rasterikuva piirtyy näytölle tai paperille vasemmalta oikealle etenevän säteen ansiosta. Kuvan tarkkuus on sidottu tulostus- tai kuvantamisvälineen pikselitarkkuuteen eli siihen kuinka monta pistettä se pystyy esittämään tuumaa kohden (*dpi, dots per inch*). Nykyisissä tietokoneissa on edelleen oma sisäänrakennettu RIP-prosessori, joka hoitaa kuvantamisen laskutoimitukset. (4, s. 13–14.)

Rasterikuva piirtyy näytöllä olevaan pikseleistä koostuvaan ruudukkoon. Piirtoprosessin keskeisin tekijä on merkin ääriviivojen sovittaminen ruudukkoon siten, että kuvan sisäpuolelle rajoitetut pikselit saavat värin ja kuvasta ei tule vääristynyttä. Kuvatarkkuuden ongelma syntyy siitä, kun näytön pikselit ovat neliön muotoisia ja niissä yritetään toisintaa virheettömän muotoista vektorikuvaa. Näytön resoluution kasvattaminen parantaa kuvan laatua, sillä pikselit määräytyvät pienemmiksi niiden lukumäärän kasvaessa ja näin niiden avulla voidaan toisintaa paremmin kaarista muodostuvaa vektorikuvaa. Kuva 1 havainnollistaa, kuinka tasapintainen vektorikuva vääristyy pieniresoluutioisessa näytössä.



Kuva 1. Matalaresoluutioisen kuvan aiheuttama vääristymä suurennettuna (9).

Keskeinen ongelma fonttien esittämisessä tietokoneen näytöllä liittyy painomedian laitteisiin suunniteltuihin fontteihin. Latomakoneet pystyvät tuottamaan 2 000 dpi tasoista printtijälkeä, kun taas tietokonenäyttöjen pikselitiheys 15 tuuman 4K-näytöllä jää hieman alle 300:aan. Verkkokehittäjien fonttivaihtoehdot ovat pitkään olleet rajoitettuja

muutamaan fonttityyliin, kuten (*Verdana, Georgia ja Arial*). Ne on suunniteltu näyttämään hyvältä pieniresoluutioisissa näytöissä. Pikselöitymisen ehkäisemiseksi on kehitetty fontin ulkoasua ennakoiva tekniikka. Varhaisimmat ohjelmointikielet, kuten PostScript ja *TrueType*, sisältävät ohjeen, jonka avulla ne hienosäätävät ruudukkoon asettuvan rasterikuvan mahdollisimman lähelle ideaalia (10). Näyttötekniologia on laatuun nähden edelleen merkittävästi jäljessä kehittyneimpiin latomakoneisiin verrattuna. Ongelman ratkaisemiseksi on kehitetty *anti-aliasing*-tekniikka, jota kutsutaan myös *font smoothing*-nimellä. Sen yksinkertaisena tarkoituksena on tehdä näytöllä olevasta tekstistä tasaisempaa. Merkit koostuvat mustista ja harmaista pikseleistä, joissa harmaapikselit sijoitetaan merkkien reunoille luomaan tasaisempi vaikutelma. Toisinaan harmaapikselien lisääminen voi saada tekstin reunan näyttämään sumealta. (4, s. 18–19; 11.)

2.5 Mittayksiköt

Typografiassa määritellyt mittayksiköt ovat pica ja sen osatekijä piste. Pisteen koko on vakiintunut PostScript-formaatin julkistamisen myötä, ja nykyään sen standardisoitu mitta on $1/72$ tuumaa eli 0,353 mm. Mitta koostuu kahdestatoista osasta, mikä tarkoittaa sen jakaantuvan tasan kuuteen picaan tuumalle. Pistekokomääritelmä vakiintui jo Gutenbergin aikana, jolloin pisteen koko määräytyi metallikirjaisinpalkan korkeudesta. Digitaalisen paperin mittasuhteiden kuvaaminen on sidottu pica-pisteeseen. Suuremmat kokonaisuudet, kuten sivuelementtien paikat ja marginaalit, kuvataan pisteinä, ja pienemmät osat on jaettu pisteen osatekijöihin.

Tekstinkäsittelysovellusten mittayksikkökäytänteisiin ovat vakiintuneet metrijärjestelmän mitat, vaikka pisteen koko on verrannollinen tuumaan. Desimaalijärjestelmän osayksiköt ovat huomattavasti hienojakoisempia tuuman osasiin nähden, ja tietokoneiden on helpompi käsitellä desimaalijärjestelmään perustuvaa dataa. Perinteisiä välimittoja ovat ohuke (*thin space*), puolikas (*en space*) ja neliö (*em space*). Välimitat olivat käytössä metalliladonnassa ja sen jälkeen valoladonnassa. Julkaisuohjelmissa käytössä oleva puolikas on $\frac{1}{2}$ neliötä ja ohuke on $\frac{1}{8}$ neliötä. Sanaväli sijoittuu puolikkaan ja ohukkeen väliin, ja se on vähintään $\frac{1}{4}$ mutta enintään $\frac{1}{3}$ neliötä. Puolikasta käytetään useimmiten taulukoissa, sillä se on yhtenevä taulukoissa käytettävän versaalinumeroitten leveyden kanssa. Ohuketta voidaan käyttää pitkissä numerosarjoissa, sillä niiden jäsentäminen tavallisella sanavälillä luo hajanaisen vaikutelman. (2, s. 121; 12.)

Merkkien suhteellista kokoa kuvataan em-mittayksiköllä, jossa yksi em vastaa yhtä pistettä. Mittayksikön nimi juontuu m-kirjaimesta, joka oli usein leveydeltään yhden pisteen verran metallikirjaisimien aikakautena. Em-yksikön määritelmä on kuitenkin muuttunut vuosien myötä kuvaamaan myös fontin korkeutta, sillä kaikissa merkistöissä ei ole m-kirjainta. *CSS (Cascading style sheet)* -tyyleissä em-yksikkö on yleismittaväline kaikille etäisyyksille, joita sivuelementtien välillä halutaan kuvata. Em-yksikkö hakee suhteellisen kokonsa ylemmän tason elementistä. Silloin kun fontille annetaan koko em-yksikkönä ja tyylietiedostossa ei ole määritelty verrattavaa fonttikokoa, sitä verrataan selaimen oletusarvoiseen fonttikokoon, joka on useimmiten 16 pikseliä. Kehittäjän on järkevää antaa alimman tason fonttikoolle jokin arvo, jotta em-yksikkönä määritellyt koot säilyttävät samat mittasuhteet selaimesta riippumatta. Näin voidaan määritellä responsiivinen, ja kaikissa näyttöko'oissa oikein skaalautuva fontti.

Monotype-latomakoneen aikakautena kirjainleikkausten suunnittelu noudatti suhteellisia mittoja, jolloin glyyfin ääriviivat mahdutettiin 18 osaan jakautuvaan pisteeseen. Digitaaliset fontit esitetään sovelluksissa, jossa piste jakautuu yli tuhanteen osaan. TrueType- ja OpenType-formaatit tarjoavat fonttikehittäjälle pisteen, joka jakautuu 32 000 osaan. Kirjainleikkauksen suunnittelun pullonkaulaksi onkin muodostunut näyttöteknologia, sillä kehitystyökalut tarjoavat lähes rajoittamattomat mahdollisuudet. Kirjainparien väliin asettava tyhjä valkoinen tila on muiden mittasuhteiden tavoin sidottu em-yksikköön, ja typografiassa sitä kutsutaan välistykseksi. Fonttiformaattien taulukoissa on jokaiselle merkille määrätty data, josta tekstinkäsittelyohjelma voi lukea kirjainparille tarvittavan välistysmitan. (4, s. 21–25.)

3 Web-fonttien teknologiset ratkaisut

3.1 Unicode

Eri kansojen kirjoitusjärjestelmiä on standardisoitu kansainvälisten *ISO* (International organization for standardization) -standardien avulla jo vuosia ennen kuin uusi ISO-komitea aloitti hankkeen, jonka tarkoitus oli yhdistää kaikki merkistöt yhdeksi yleiskäyttöiseksi merkkikokoelmaksi (14). Vuonna 1984 aloitetun kehitystyön nimike oli *ISO 10646*, ja se suunniteltiin yhteensopivaksi aikaisemmin julkaistun *ISO 2022* standardin kanssa. Yksi keskeisimmistä päämääristä oli yksiviivaistaa edeltävän *ISO 2022:n* yksi- ja kaksibittinen perusta kaksibittiseksi järjestelmäksi. Teknologiyhtiöt

Apple ja Xerox kehittivät samaan aikaan omaa vastaavanlaista koodistoaan nimeltään Unicode. Myöhemmin molemmat kehittäjäryhmät päätyivät yhdistämään osaamisensa, ja Unicode julkaistiin vuonna 1993. Useimmat käyttöjärjestelmät käyttävät Unicoden merkistöä, ja merkittävä osa verkkosivuista hyödyntää sitä. Unicode on muutakin kuin merkkiketju: sen koodi voi sisältää ohjeen, joka auttaa sovellusta tulkitsemaan sen sisällön oikein. Numerot voivat esiintyä eri kielissä symboleina, joten niiden tulkinta vaatii merkityksen sitomisen useampaan alityyppiin. Lisäksi muinaisten kielten sisältämät glyyfit voivat kerrallaan ilmaista useaa merkkiä, ja oikean vastaavuuden tavoittamiseksi glyyfiä tuleekin tarkastella asiayhteyden kautta. (13, s. 53–57.)

Universaalisuuden periaate asettaa merkistölle tiettyjä haasteita. Merkin ja glyyfin välistä eroa on pohdittu filosofiankin avulla, ja tarkkaan määritelmään ei ole vielä kukaan päästy. Ongelmaa on lähestytty teknisestä näkökulmasta, ja sitä on yritetty selkeyttää poissulkevalla menetelmällä, eli merkin tai glyyfin monimerkityksellisyys on sidottu kontekstiin. Filosofisten kysymysten lisäksi on olemassa joukko teknisiä kysymyksiä liittyen glyyfien esitystapaan eri ympäristöissä. Tietoliikenteessä tiedonsiirtoa kuvataan protokollapinin avulla, joka etenee matalasta (fyysinen) korkeaan (sovellus) tasoon. Samankaltainen tasojärjestelmä on myös Unicodessa, jossa merkistön esitystapa on eriytetty viiteen kerrokseen. Järjestyksessä matalasta korkeaan ne esitetään seuraavasti:

- *Käsitteellinen merkkikokoelma* ACR (Abstract character repertoire) sisältää ainoastaan kuvauksen merkistöstä ilman minkäänlaista indikaattoria merkkien sijainnista Unicode-taulukossa.
- *Koodattu merkkikokoelma* CCR (Coded character repertoire) käsittää merkkien lisäksi niiden sijainnin taulukossa.
- *Lomake merkistön koodaukseen* CEF (Character encoding form) mahdollistaa merkkien palauttamisen tietokoneella esitettävään muotoon niihin sidotun taulukkosijainnin (koodipiste) avulla.
- *Merkistön koodauskaavio* CES (Character encoding scheme) esittää merkistön tavuina.

- *Kuljetussyntaksi* TES(Transfer encoding syntax) muuntaa merkistön kohdelaitteen vaatimaan muotoon. (13, s. 62–63.)

UTF-8 (Unicode transformation format) on kaikkein yleisin selaimessa käytetty merkkikooderi, koska XML-merkkikieli käyttää oletusarvoisesti sen merkistöä. Se pystyy toistamaan minkä tahansa Unicoden sisältämän merkin. Siinä yhdistyvät samanaikaisesti järjestelmätasot CEF ja CES, koska se koostuu kokonaan tavuista. Käytössä on neljä tavua ja jokainen voidaan osoittaa yhtä koodipistettä kohti. Koodipisteen lukuarvon suuruus korreloi suoraan käytettyjen bittien lukumäärän kanssa, eli pieniarvoisen pisteen koodaukseen käytetään vähemmän bittejä. Käytännössä lomake muuntaa numerot binäärimuotoon, jotka Unicode on muuttanut numeromuotoon. Ennen UTF-8:aa internetin käytetyin merkkikooderi oli *ASCII* (American standard code for information interchange). UTF-8 on suunniteltu tämän jatkumoksi eli kaikki vanhat merkit toimivat uudessa kooderissa. UTF-8 vaatii merkkijonon halkaisemisen osiin juuri oikeasta kohdasta, jotta halutut merkit saadaan esiin. Merkkijonon katkaiseminen juuri ennen välittäjätavua palauttaa väärät merkit. Tällaisessa tapauksessa sovellus voi hylätä saadut merkit tai jättää välittäjätavun huomioimatta ja aloittaa jonon tulkitsemisen ensimmäisestä tavusta lähtien. (13, s. 65.)

Unicoden kehittäjät päivittävät jatkuvasti UTF-8:aa vastaamaan nykykielen uusimpia merkkejä. Viimeisin päivitys toi 7 500 uutta merkkiä kokoelmaan. Uudistuksessa tuli muunmuassa kuusi uutta kieltä ja 72 emoji-merkkiä. (14.)

3.2 Fonttiformaatit

Typografiassa puhutaan samanaikaisesti fonteista ja kirjainleikkauksesta, vaikka niiden välillä on selvä ero. Fontti tulee ranskan kielen sanasta *fondre* ('valaa'), sillä sen alkuperä juontuu kirjapainon alkuvuosiin asti, jolloin kirjaisimet valettiin lyijyseoksesta. Alkuaan sanan merkitys tarkoittikin vain yhdellä koolla valettua merkistöä. Nykyään sen merkitys on laajentunut sisältämään jonkin kirjainleikkauksen kaikki mahdolliset koot. Kirjainleikkaus voidaan ymmärtää eräänlaisena esteettisyyden itseisarvona, sillä se ilmentää ainoastaan johonkin tyyliin sidotun merkistön visuaalisen asun. *"Fontti on se mitä käytetään ja kirjasintyyppi se mitä nähdään"* (2, s. 14–15).

Useimpien aakkosellisten fonttien merkistöt ovat standardisoituja, eli niiden merkkikokoelmat ovat lähes identtisiä keskenään. Fonttikoko sisältää jokaisen merkin käsittämisen alkion leveyden, joka ilmaistaan *em*-mittayksikön murto-osina. Tekstinkäsittelyohjelmat hyödyntävät fonttiin sidottua alkion leveyttä rivipituuden ja muiden mittasuhteiden määrittämiseksi. Fontti sisältää samaan fonttiperheeseen lukeutuvat kirjainmuodot, kuten *italic*, *bold* ja *bold italic*. Fonttiin kuuluu myös välistystaulukko, joka sisältää kirjainparien keskinäiset ja rivin koostamiseen vaadittavat leveystiedot. Fonttiformaatit oli alun perin sidottu tietokoneessa olevaan käyttöjärjestelmään. Adoben ja Microsoftin kehittämä *OpenType*-formaatti on ensimmäinen fontti, jonka koodaus on samanaikaisesti yhteensopiva Applen ja Microsoftin käyttöjärjestelmien kanssa. (4, s. 51–52.)

PostScript

PostScript-ohjelmointikielellä kirjoitettu fontti sisältää kahdenlaista dataa, jota voidaan lukea painolaitteissa ja tietokoneen näytöllä. Datan tulkitseminen vaatii oman prosessorin, joka integroidaan painokoneeseen valmistusvaiheessa. Sisältö muutetaan matemaattiseen muotoon, jonka prosessori muuntaa sopivaksi tulostamista varten. Datan sisältämät glyyfit koostuvat Bezier-käyristä ja viivoista. Tietokoneissa on sisäänrakennettuna oma tulkitin, joka muuntaa datan kuvantamiseen sopivaksi. OpenType-formaatin julkistamisen jälkeen PostScriptin käyttö on vähentynyt merkittävästi ja Adobe on muuntanut sen merkkikokoelmia OpenType-formaattiin. Formaatti on edelleen laajalti käytössä sen aikaisemman suosion ansiosta, ja useimmat sovellukset ja laitteet ovat edelleen yhteensopivia sen kanssa. (13, s. 635–636.)

TrueType

Adoben kehittämä PostScript-formaatti oli 1980-luvun loppupuolella sivukuvauskielten standardi. Microsoft ja Apple päättivät yhteisvoimin aloittaa uudenlaisen fonttiformaatin kehitystyön horjuttaakseen Adoben monopoliasemaa. Yhtiöt sopivat työnjaosta siten, että Apple vastaa fonteista ja Microsoft pitää huolen grafiikkasuorittimen ohjelmoinnista. Microsoftin kehittämää ensimmäistä versiota nimeltään *TrueImage* ei koskaan julkaistu. Applella oli samaan aikaan vastaava projekti, jossa suomalainen insinööri Sampo Kaasila toimi ohjelmistokehittäjänä. Projekti nimettiin uudestaan TrueType-nimellä, ja se julkaistiin vuonna 1991. TrueType oli huomattavasti kehittyneempi edeltäjäänsä PostScriptiin verrattuna. Keskeisin uudistus oli fonttiin ohjelmoitu lisäke, jonka avulla

merkkien ääriviivat säilyttävät terävyyden eri resoluutioissa. Erityisesti pienikokoisten merkkien esittäminen näytöllä sai merkittävän parannuksen uudesta teknologiasta. TrueType-fontti on jaettu kahteen osaan: fonttiedosto ja sen esittämiseen suunniteltu rasteri. (13, s. 705; 4, s. 53.)

Fontin ääriviivat on tallennettu fonttiedostoon koordinaatteina, jotka muodostavat merkkien ääriviivat. Esitettäväksi tuleva glyyfi skaalataan laitekohtaiseen kokoon, ja samalla alkuperäiset ääriivakoordinaatit muunnetaan pikselikoordinaatistoon sopivaksi. Sovituksen jälkeen glyyfi skannataan, jotta saadaan selville, mitkä pikselit osuvat ääriviivoihin. Tämän jälkeen glyyfi muunnetaan rasterille sopivaan muotoon. Ääriviivat koostuvat suorista viivoista ja kaarista. Kaaret määritellään *Bezier-spline* pisteinä, jotka voivat sijaita kaarella tai sen ulkopuolella. Molemmat esitysmuodot lukeutuvat kaaren määritelmään. Suorat ääriviivat määritellään kahden perättäisen suoralla sijaitsevan pisteen perusteella. TrueTypessä on oma fontin kokoa kuvaava yksikkö (*FUnit*), joka on em-mittayksikköön perustuvan neliön pienin osamäärä. Em-neliö sisältää useimmiten kaikki merkin osat. Yksittäisten merkkien viiva tai kaari voi kuitenkin ulottua neliön ulkopuolelle. TrueType pystyy näyttämään kaikki merkit, vaikka niiden osat eivät mahtuisikaan neliön sisäpuolelle. Glyyfin kaarella sijaitseva piste säilyttää koordinaattiinsa eri pistekokoon muunnettaessa, sillä molemmat ovat suhteellisia em-neliöön. Näin ollen alkuperäiseen merkkiin tehdyt muutokset voidaan todentaa heti riippumatta siitä, mihin resoluutioon merkkiä ollaan skaalaamassa. (15.)

Yksi skaalausoperaation keskeisimmistä piirteistä on huolehtia glyyfin mittasuhteiden säilymisestä uuteen resoluutioon muunnettaessa. Skaalausohjeen tulee sisältää värin saavien pikseleiden koordinaatit, jotta se rasteroituu oikein pikseliruudukkoon. Rasteroitu glyyfi joudutaan toisinaan sovittamaan ruudukkoon uudestaan vääristämällä sen mittasuhteita. Sovittamisprosessi (*grid-fitting*) täyttää glyyfin ääriviivat pikseleillä fonttiedoston sisältämän ohjeen avulla. Fonttia voidaan käyttää ilman ohjettakin, ja yleensä se skaalautuu vääristymättä. Fontin esittämiseen vaikuttavat kohdelaitteen resoluutio, merkin pistekoko ja kirjainleikkaus. Ohjeen tarpeellisuuteen vaikuttaa sen käyttökohde. (16.)

Merkkien esittäminen pieniresoluutioisissa näytöissä pienellä fontilla vaatii ohjeistuksen sisällyttämisen fonttiin. Ohjeessa on merkkikohtaisesti määritelty tarvittavat tiedot skaalauksen varten, mikäli tulkitseminen katsoo tarpeelliseksi vääristää merkin

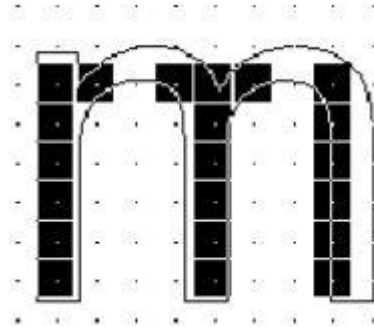
kohdelaitteeseen sopivaksi. Ohjeistuksen toteuttamiseen on kehitetty tulkitsin, joka nimensä mukaisesti tulkitsee merkkiin sisällytetyn ohjeen.

Ohjeet on järjestetty pinoon, josta ne käsitellään yksitellen. Tulkitsin hakee ohjeen erillisestä ohjeistuskanavasta, jossa operaatiokäskyt on järjestetty pinoon. Käsky sisältää yhden tavun verran dataa, ja se voi olla pakattu yksi- tai kaksitavuiseksi. Kaksitavuinen käsky tarkoittaa sanaa, ja sen esittämisessä tulkitsin asettaa korkeampiarvoisen tavun ennen pienempää. Käskyn suorittamiseen vaikuttaa pinossa olevan komennon laatu. *Push*-komento ottaa ohjeen suoraan kanavasta. Muut pinossa olevat komennot ottavat tarvitsemansa datan pinosta. Ohjeet liikkuvat pinossa ”viimeinen sisään ja ensimmäinen ulos” -periaatteella. Ohjeen suorittamisesta saatu tuotos työnnetään pinon ensimmäiseksi, jossa siitä on potentiaalisesti eniten hyötyä seuraavalle ohjeelle. Pinon ohjaamiseen tarvittavat operaattorit on sisällytetty ohjeeseen. Ohje on suoritettu sitten, kun pino on tyhjä. Pino toimii eräänlaisena väliaikaismuistina, eli pinossa olevaa tietoa voidaan pitää tallessa ja muokata sen sisältöä.

TrueType-fontti koostuu taulukoista, ja ohje voidaan sijoittaa sen eri sijainteihin. Ohje voidaan sijoittaa fonttiohjelmaan (*font program*), ohjearvotaulukkoon (*CVT, control value table*) tai glyyfidataan. Glyyfiin sijoitettu ohje muuttaa vain yhtä merkkiä, kun kahdesta ensimmäisestä löytyvä ohje vaikuttaa koko merkistöön. Fonttiohjelmaan lukeutuva ohje suoritetaan vain ensimmäisellä lukukerralla. Sen avulla luodaan funktioiden ja ohjeiden kuvauksia, jotka sijoitetaan fonttiedostoon yleiskäyttöä varten.

CVT-taulukko sisältää sarjan komentoja, jotka suoritetaan aina, kun merkin pistekoko tai muoto muuttuu. Taulukolla pyritään säilyttämään merkistön yhtenäisyys fontin ohjeistuksen lomassa. CVT komentoja voidaan säilyttää merkkimuunnoksen jatkohyödyntämisen varalta. CVT-taulukon arvot ovat ilmaistu FUnit-yksikkönä. Merkin muuntaminen FUnitista pikselihin muuttaa samalla taulukon arvot uuteen yksikköön. Taulukkoa tulkitseva skannaaja hakee glyyfin ääriivakoordinaatit ja muuntaa sen pikseleiksi. Aluksi skannaaja sovittaa pikseliruudukon glyyfin päälle ja valitsee ne pikselit, joiden keskikohta osuu sen kaareen. Sen jälkeen sama toimenpide suoritetaan kääntäen, eli glyyfi sovitetaan pikseliruudukkoon ja valitaan pikselit keskikohtien mukaan. Kaikkia glyyfin ääriiviivojen sisään osuvia pikseleitä ei voida valita sen esitykseen. Poissulkevaa metodia käytetään pikselien karsimiseen silloin, kun kahden perättäisen ääriviivan sisäpuolelle jäävien pikselien keskikohtaa ei voida yhdistää

suoralla, joka pysyy ääriiviivojen sisällä. Pikselien poissulkemista hyödynnetään useimmiten monimutkaisten fonttien ja näyttöjen kohdalla, joiden pikselit ovat pienikokoisia. Kuva 2 esittää m-kirjain, jonka kahta pikseliä ei voida yhdistää suoralla, joka pysyy ääriiviivojen sisäpuolella. (16.)



Kuva 2. Kaksi pikseliä on suljettu poissulkemissäännön mukaisesti (16).

TrueTypessä on edeltäjiinsä nähden suurempi merkkikokoelma sen kaksi-tavuinen indeksointijärjestelmän vuoksi. PostScriptissä on käytössä yksitavuinen merkkien indeksointiin käytetty järjestelmä, joka enimmillään nimeää 256 merkkiä tunnisteeseen. Kaksitavuinen järjestelmä mahdollistaa 65 000 merkkisen kirjaston. Laaja merkkivalikoima on tuonut aasialaiset kirjainmerkit kaikkien ulottuville. Käytetyimmät selaimet ja niiden varhaisimmatkin versiot tukevat TrueType-formaattia. Opentype jatkokehitettiin TrueTypestä Adoben ja Microsoftin yhteistyönä. Siinä yhdistyvät PostScriptin ja TrueTypen ominaisuudet sekä mahdollisuus käyttää samaa fonttiedostoa Applen ja Microsoftin käyttöjärjestelmissä. Edeltäjiensä tavoin OpenType koostuu taulukoista, joissa on fontin ohjedataa ja sen merkistö. Lisänä on mahdollisuus parsia tekstiä, mikä onnistuu taulukko-dataa muokkaavalla pienoisohjelmalla. Merkkien ääriiviivoja voidaan muunnella uuden ominaisuuden ansiosta. (4, s. 53.)

Verkkoformaatit

Painolaitteisiin verrattuna digitaalisten fonttien suunnittelu ja käyttö on huomattavasti rajoittuneempaa pelkästään näyttötekniikan kehittymättömyyden takia. Tietokonenäyttöjen heikko resoluutio rajoittaa, mutta samalla se asettaa suunnittelulle selkeän kehyksen. Uusimmat pieninäyttöiset lukulaitteet ja tabletit tarjoavat parhaimmillaan 300 dpi:n tasoista kuvaa, joka mahdollistaa typografian suunnittelijalle

joitain vapauksia tekstin suunnittelussa tavallisiin tietokonenäyttöihin verrattuna. Laadun rajoitteet vähenevät huomattavasti, kun näyttötarkkuus asettuu 300 dpi:stä suurempaan kokoluokkaan.

Näyttöfonttien kirjainvälit on suunniteltu väljäksi näyttötekniologian asettaessa rajoitteensa. Ensimmäiset näytöille suunnitellut fontit kehitettiin näyttämään hyvältä erittäin heikkoresoluutioisissa näytöissä. Microsoftin tarjoamat Georgia- ja Verdana-fontti suunniteltiin näyttämään hyvältä pikseliruudukossa. Kirjainleikkausten kehitystyön keskeisimpiä tavoitteita oli varmistaa merkkien selkolukuisuus pienellä pistekoolla, joka on ominaista internetissä käytetyille tekstile. Fontit kehitettiin pikseliruudukolla, jotta pystyttiin varmistamaan niiden selkolukuisuudesta. Nykyään on kehitetty useita verkkoturvallisia fontteja, jotka toimivat varmasti kaikissa digitaalisissa näytöissä. Verkkoturvalliset fontit ovat sisäänrakennettuja nykyaikaisiin selaimiin, ja kehittäjän tarvitsee vain mainita CSS-tiedostossa, mitä fonttia haluaa käyttää.

Näytöllä esitettävää tekstiä voidaan selkeyttää asettamalla fonttikoko suureksi. Useimmiten internetsivustoilla näkyvä teksti on suurikokoista, jotta voidaan välttää kirjainleikkauksen ääriviivojen vääristymiseltä. Käytössä on myös anti-aliasing-tekniikka, jossa merkin ääriviivoja pehmennetään harmailla pikseleillä. Päätyöntekijän silmä voi joutua tarkentamaan katsetta 40 000 kertaa tavallisen työpäivän aikana. Silmien väsymien välttämiseksi fontin koko voidaan nostaa kynnyksen yli, jossa anti-alias-tekniikan hyödyt tulevat esiin. (4, s. 286–289.)

Tekstin esittämisessä on olemassa sääntöjä, jotka yhdessä ihmisen näköjärjestelmän kanssa ovat säilyneet muuttumattomina pitkiä aikoja. Lukeminen on kognitiivinen prosessi, jossa merkkien muodostama informaatio tulkitaan ja järjestetään osaksi kokonaisuutta, josta tiedon sisäistäminen muodostuu. Sisäistämisen tehokkuuteen vaikuttavat typografiset ja psykofysikaaliset muuttujat. Yksittäisten kirjainten havaittavuuteen ja tunnistettavuuteen vaikuttavat psykofysikaaliset tekijät, kuten kontrasti, polariteetti ja kirjainkoko. Sanojen tunnistettavuuteen ja lauseiden ymmärtämiseen vaikuttavat typografiset muuttujat, kuten kirjainlaiji, rivin pituus, viivojen voimakkuus ja välistys. Luettavuus koostuu siis psykofysikaalisista ja typografisista tekijöistä, jotka yhdessä kuvaavat tekstin merkityksen ymmärtämistä.

Luettavuuden tutkimiseksi on kehitetty kokeita, joissa mitataan lukunopeutta, sisällön ymmärtämistä ja yksittäisten sanojen ja merkkijonojen tunnistamista. Tekstin graafisen

ulkoasun ominaisuuksia tutkitaan seuraamalla silmänliikkeitä, ja samalla saadaan selville lukunopeus. Lukunopeuden lisäksi halutaan selvittää, kuinka hyvin lukija on sisäistänyt tekstin sisällön. Graafisen ulkoasun osuutta lukemisen rasittavuuteen pyritään selvittämään eri menetelmin. Usein lukijaa pyydetään arvioimaan silmien väsymistä seitsenportaisen asteikon avulla. (8, s. 125–127.)

Verkkoselainten neljä yleisintä fonttiformaattia ovat *EOT (Embedded OpenType)*, *TTF (TrueType font)*, *WOFF (Web open font format)* ja *WOFF2*. Formaattien yhteensopivuutta selaimen kanssa tulee tarkastella tapauskohtaisesti. Universaalia kaikissa uusissa ja vanhoissa selaimissa toimivaa formaattia ei ole vielä kehitetty. EOT-formaatti on kehitetty OpenTypen verkkofontiksi. Formaatti toimii ainoastaan Microsoftin selaimessa. Woff-formaatti on käytännössä vain pakkausformaatti, joka pakkaa fontin datan CSS-tyyleissä määriteltävällä *@fontface*-komennolla. Pakkausformaatin lisenssi kattaa yhteensopivuuden TrueType- ja OpenType-formaattien kanssa. Pakkaustiedostoon voidaan sisällyttää metadatan suunnittelijoita tai lisenssioikeuksia valvovia tahoja varten. Fonttikehittäjälle WOFF-formaatti tarjoaa ilmaisen jakelukanavan, joka edesauttaa fonttien saatavuutta. (17; 18; 19.)

4 Web-standardien asettamat vaatimukset

Näyttölaitteiden monenkirjaisuus on tämän päivän teknologinen standardi, ja niiden lisääntymistä voidaan pitää suoranaishana itseisarvona. Verkkosivustojen skaalautuminen eri näyttökokoihin on välttämättömyys, sillä kuluttajat ovat tottuneet siihen viimeisen kymmenen vuoden aikana. Responsiivinen suunnittelu yrittää vastata tähän tarpeeseen monen osatekijän kautta. Responsiivisesti suunnitellussa sivustossa näytölle asettava sisältö on mallinnettu pikseliruudukkoon, jonka elementit skaalautuvat symmetrisesti käytettävän laitteen mukaisesti. Laittekohtaisen verkkoselailun painopisteen kääntyessä pöytäkoneista mobiililaitteisiin täytyy kehittäjän optimoida myös sivuston latausaika. Latausaika on riippuvainen datan määrästä, jonka selain joutuu hakemaan palvelimelta. Suurimman osan verkkosivuston datasta kattavat kuvat, joiden osuus on keskimäärin 62 %. Sivustoilla käytettävien erikoisfonttien lisääntyessä kasvaa myös datan määrä, joten kehittäjän on syytä pohtia erillistä strategiaa fonttiedoston lataamiseen (18). Kuva 3 esittää Css-tiedostossa määritetyt suhteelliset fonttikoot eri näyttöko'oilte.

```

/* SMARTPHONES LANDSCAPE */
@media only screen and (min-width: 480px) {
  p {
    font-size: 0.5em;
  }
}

/* TABLETS PORTRAIT */
@media only screen and (min-width: 768px) {
  p {
    font-size: 0.7em;
  }
}

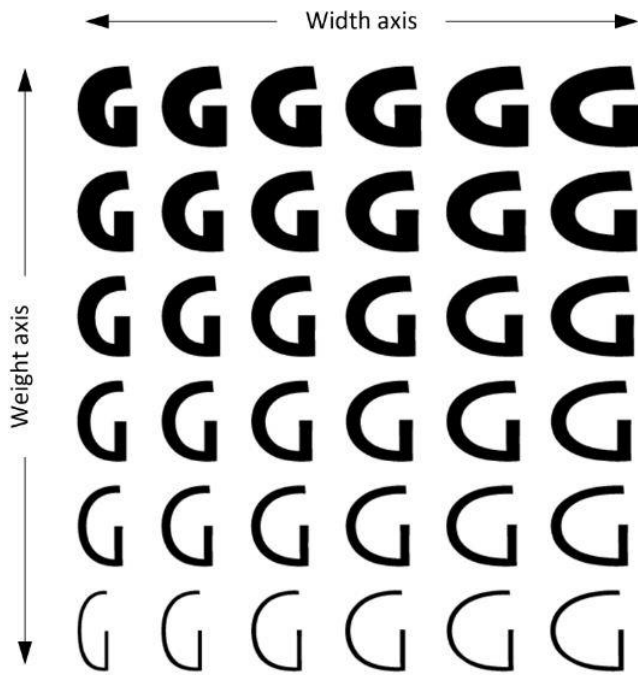
```

Kuva 3. Rajapisteissä määritelty suhteellinen fonttikoko

Skaalautuvuuden avuksi on olemassa CSS-mediakyselyt. Niiden avulla voidaan määrittellä pikseli- tai em-yksikköön sidottuja rajapisteitä eri näyttöko'oilte, jotka asettavat voimaan tarvittavat mittasuhteet. Verkkosivustolla esitettävään tekstiin suhtaudutaan elementin tavoin, joten sen mittamuutokset ilmoitetaan myös mediakyselyissä. Useimpien selainten oletusarvoinen fonttikoko on 16 pikseliä eli 1 em. Rajapisteissä määritelty suhteellinen fonttikoko voidaan ilmoittaa em-yksikön kertoimena.

4.1 Muuntuva fontti

Muuntuva fontti on vielä kehitysvaiheessa oleva teknologia, jota Adobe, Microsoft, Apple ja Google kehittävät yhteistyössä. Uutuusteknologia tulee toimimaan OpenType-formaatissa fonttikehittäjän lisätyökaluna ja mahdollistamaan fonttiperheen kirjaintyyppien sisällyttämisen yhteen fonttitiedostoon. Kirjainleikkaus on jaettu yhteen tai useampaan akseliin, joita kehittäjä voi muunnella samanaikaisesti. Kirjaintyyppien sisällyttäminen yhteen fonttitiedostoon tuo esiin useita hyötyjä nykyisiin fontteihin verrattuna. Useiden fonttitiedostojen sijasta fonttissa on lukuisia taulukoita, joihin variaatiot on koodattu. Näin säästyy levytilaa ja selaimen tarvitsee ladata vain yksi tiedosto usean sijaan. Typografinen sommittelu eri näyttöko'oilte helpottuu, kun tekstin leveys- ja lihavuussäädöt voidaan integroida fonttikokoihin. Kuvassa 4 g-kirjaimen muunto-operaatiot on esitetty paino- ja leveysakseleissa. (20.)

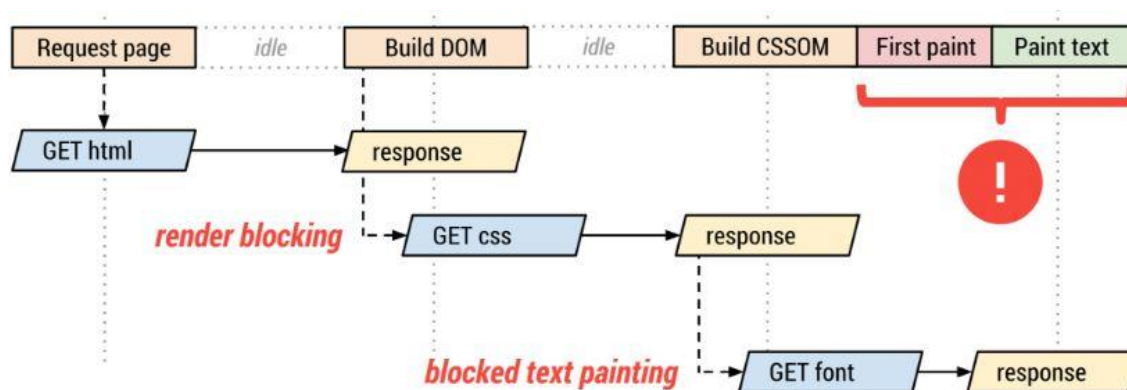


Kuva 4. Kirjainleikkauksen variaatio kuvattuna leveys- ja painoakseleissa (20).

Fonttivarიატიot sijaitsevat taulukossa, joka määrää, miten fontti esitetään loppukäyttäjälle. Variaatioakselin muuttujat on jaettu gradientin etenemissuunnan mukaisesti, jota kuvataan numeerisilla arvoilla. Näin kehittäjällä tai sovelluksella on suora linkki akselilla sijaitsevaan variaation instanssiin. Kehittäjällä on mahdollisuus nimetä mikä tahansa akselilla sijaitseva instanssi. Instanssin nimi voi olla esimerkiksi "Light" tai "Semibold", ja sama nimi voidaan antaa mille tahansa instanssille. Responsiivisuuden kannalta merkittävimmät variaatioakselit sisältävät kirjaimen paino- ja korkeussäätömahdollisuuden. Instanssia voidaan muokata myös tyyliattribuuttien avulla, joiden avulla voidaan hienosäätää yksittäisen merkin typografisia ominaisuuksia. Nämä ominaisuudet sijaitsevat omassa taulukossa, ja niihin on sidottu instanssin lisäkuvaukset, jotka voidaan palauttaa taulukkoon tallennetun merkkijonon avulla. Tyyliattribuuttien avulla voidaan usealla akselilla muokattu fontti interpoloida yhdeksi instanssiperheeksi, mikä mahdollistaa paremman yhteensopivuuden vanhemmissa sovelluksissa. Käyttöjärjestelmä pystyy tulkitsemaan vanhan sovelluksen tukemat variaatiot ja suodattamaan instanssiperheestä sille sopivan tyyliattribuutin. Instanssien muokkaaminen ketjuttaa muutokset muihinkin fonttitaulukoihin. Merkkien oletusarvoiset ääriarvot on tallennettu glyfitaulukkoon, ja niistä muokatut instanssit tallentuvat glyyfi variaatiotaulukkoon. Fonttivarიატიotaulukossa on jokaiselle akselille oma oletusarvoinen instanssi, joka mahdollistaa sen käyttämisen ilman kehittäjäkohtaisia variaatioita. (20.)

4.2 Erikoisfontin käyttö

Selaimissa on omat oletusarvoiset fontit, jotka on kehitetty näyttämään hyvältä näytöissä. Tästä huolimatta verkkosivustojen kehittäjät hyödyntävät laajalti erikoisfontteja. Tuoreen tutkimuksen mukaan 67 prosenttia verkkosivuista käyttää räätälöityä fonttia (21). Verkkoselain hakee internetsivuston sisällön palvelimelta *HTML (Hypertext markup language)* -pyynnön välityksellä. Selain rakentaa tiedoston muuntopolun *DOM (Document object model)*- ja *CSSOM (Cascading style sheet object model)* -puukuvion muodossa, jotta sille selviää, mitä fontteja esitetään (22). HTML-pyyntöä koostuva polku on esitetty kuvassa 5.



Kuva 5. Verkkosivuston datan hakeminen palvelimelta ja purku selaimessa (22).

Verkkosivuston sisältö rakentuu aikajärjestyksessä, ja sen kulku on esitetty seuraavasti:

1. Selain pyytää HTML-dokumentin.
2. Dokumenttia aletaan purkaa ja DOM alkaa muodostua.
3. Selain tunnistaa CSS, Javascript ja muut tiedostot ja aloittaa niiden purkamisen.
4. CSSOM rakentuu CSS- sisällön latautumisen jälkeen ja se yhdistetään DOM:iin. Tässä vaiheessa tiedetään, mitä fontteja sivustolla tarvitaan ja lähetetään pyyntö palvelimelle.
5. Selain alkaa piirtää sisältöä näytölle. Teksti piirretään kaikilla selaimilla viimeistään siinä vaiheessa, kun fontitiedosto on latautunut. (22.)

Selainten käytännöt esittää tekstiä poikkeavat selkeästi toisistaan. Safari esittää tekstin vasta, kun fonttiedosto on latautunut. Chrome ja Firefox odottavat kolme sekuntia fonttiedoston latautumista, minkä jälkeen teksti esitetään selaimen oletusarvofontilla, vaikka erikoisfontti ei olisikaan latautunut. Internet Explorer taas piirtää fontin heti ensimmäisten sivuelementtien joukossa, vaikka fontti ei olisikaan latautunut. Palvelimelta ladattava fontti haetaan CSS:n @font-face-komennolla. Verkkokehittäjän tulee pystyä esittämään suunnittelemansa sivusto yhtenevästi selaimesta riippumatta, joten selainten vaihtelevaan fonttikäsittelyyn täytyy puuttua. Tähän on olemassa CSS-komento *font-display*, joka määrittelee eri metodeja ladattavalle fontille. Metodien avulla kehittäjä pystyy räätälöimään selainkohtaisen strategian fontin esittämiseen. (23; 24.)

Selainten vaihtelevat käytännöt esittää tekstiä tuovat esiin kaksi ongelmallista ilmiötä: *FOIT (Flash of invisible text)* ja *FOUT (Flash of unstyled text)*. Kehittäjien joukossa vallitseva yhteinen näkemys on, että kaikkein huonoin vaihtoehto käyttäjystävällisyyden kannalta on FOIT eli se, että selain piirtää tekstin vasta, kun fonttiedosto on latautunut. Useimpien selainten käytäntö on käyttää FOUT-strategiaa eli piirtää teksti välittömästi näytölle käyttäen selaimen oletusarvofonttia tai CSS-tyyleissä määritellyn turvafontin avulla, joka on kirjainleikkaukseltaan mahdollisimman yhtenevä erikoisfontin kanssa. Fonttiedoston latauduttua teksti yleensä välähtää, kun selain vaihtaa fonttia. Ihmiset ovat kärsimättömiä odottamaan sivun latautumista. Mobiililaitteilla kärsimättömyys korostuu, ja tutkimuksen mukaan 53 % mobiilikäyttäjistä poistuu sivustolta, jos sisältö ei ole latautunut kolmessa sekunnissa (24). Mobiilisivustojen keskimääräinen latausaika on 14 sekuntia 4G-verkossa. Kolme keskeisintä latausaikaan vaikuttavaa tekijää ovat datan määrä, elementtien järjestys ja palvelinpyyntöjen lukumäärä. Mobiililaitteilla tapahtuva verkkoselailu on lisääntynyt tasaista tahtia vuodesta 2009 lähtien. Statcounterin tilaston mukaan mobiililaitteilla tapahtuva verkkoselailu ohitti pöytäkoneet vuoden 2016 viimeisellä neljänneksellä. (26; 27.)

5 Yhteenveto

Typografian kehittyminen jatkuu edelleen, ja teknologia muovaa sen toteutumisympäristöä. Kirjoituksen siirtyminen paperilta tietokoneen näytölle ja edelleen mobiililaitteisiin on asettanut uusia vaatimuksia typografialle. Typografiasta on sanottu sen olevan enemmän tiedettä, kuin taidetta ja insinööriyössä tehdyn selvityksen nojalla se voidaan helposti osoittaa todeksi. Tieteessä keksinnöt usein seuraavat toisiaan, ja

typografia on siitä malliesimerkki. Käsitys kirjoituksen ilmeen perusvaatimuksesta on opittua ja samalla myös ihmismieleen sisäänrakennettua, eli typografian olemassaolo paljastuu keskivertolukijalle viimeistään siinä vaiheessa kun teksti pistää silmään.

Graafinen ilmaisuvapaus on digitalisoitumisen myötä saanut näennäisen vapaan kehyyksen. Digitaalisuus tarjoaa lähes rajattoman teknisen vapauden fontin muokkaamiseen. Vapaudesta huolimatta typografia ei koskaan voi unohtaa menneisyyttään, eli on pysyttävä pitkään kypsyneissä normeissa. Normit koskevat suurimmalta osin lukulaitteita, jotka yrittävät mallintaa kirjan tarjoamaa lukukokemusta. Uusimmat lukulaitteet ovat jo kelpo laitteita teknisiltä ominaisuuksiltaan, ja niiden luonnonvaloa heijastavat bi-stabiilinäytöt eivät kuluta lukutilanteessa ollenkaan virtaa.

Merkittävin kokeiluympäristö typografialle ja siihen liittyville tottumuksille on tällä hetkellä verkkoselain. Selaimessa näkyvää tekstiä kutsutaan ennemminkin fontiksi kuin typografiaksi. Verkossa esiintyvillä fonteilla on uusi ja vapaampi ympäristö kokeilulle lukulaitteisiin verrattuna. Fonttikirjastojen helppo saatavuus ja muokkaamisen helppous mahdollistaa uusien vapaampien typografisten käytänteiden harjoittamisen. Fonttitekniikan täytyy pystyä vastaamaan verkkoselailun tuomiin haasteisiin. Fontti on dataa, jonka koon täytyy olla mahdollisimman pieni, ja muokausmahdollisuuksien pitää olla kattavat. Näihin kysymyksiin on tarjottu ratkaisuksi muunmuassa muuntuvaa fonttia, joka pienentää fonttiedoston kokoa ja tarjoaa kehittäjille paremmat muokkaustyökalut. Verkkoselailutottumukset pakottavat teknologiayhtiöt kehittämään kokonaisvaltaisempia ratkaisuita fontin esittämiseen sen ollessa sidottuna nopeusvaatimukseen. Yleisellä tasolla voidaan todeta, että digitaaliteknologiassa kuten fonteissakin selkein kehityssuunta on muuttua koko ajan pienemmäksi ja tehokkaammaksi.

Verkkokehittäjä voi hyödyntää insinööriyössä tehtyä selvitystä, jossa käydään läpi selain- ja erikoisfonttien toimintaperiaate ja niiden käytössä ilmeneviä ongelmatilanteita. Tämän lisäksi verkkokehittäjä voi hyödyntää insinööriyössä tehtyä katselmusta tulevaan, jossa selvitettiin muuntuvan fontin toimintaperiaate. Insinööriyössä ei selvitetty fonttien lataamiseen käytettävien Javascript-pohjaisten fonttilataajien hyödyntämistä.

Lähteet

- 1 Nuolenpäistä aakkosiin. 2003. Verkkodokumentti. Kansallisarkisto. <<https://www.kansalliskirjasto.fi/extra/verkkonayttelyt/kirjaimistot/sumerilaiset.html>>. 17.9.2003. Luettu 8.3.2017.
- 2 Itkonen, Markus. 2012. Typografian käsikirja. Helsinki; RPS-yhtiöt.
- 3 Fischer, Steven Roger. 2001. History of writing. London: Reaktion books.
- 4 Felici, James W. 2012. The complete manual of typography: A guide to setting perfect type. California: Peachpit.
- 5 Koskinen, Pertti. 2010. Graafisen alan sanasto. Helsinki: Libris.
- 6 Typewriters: History & Types. 2017. Verkkodokumentti. Study.com. <<http://study.com/academy/lesson/typewriters-history-types-quiz.html>>. 10.3.2017. Luettu 10.3.2017.
- 7 Carter, Rob, Day, Ben, Meggs, Philip, Maxa, Sandra & Sanders, Mark. 2015. Typographic design: Form and communication. New Jersey: John Wiley & Sons.
- 8 Brusila, Riitta, Hinkka, Jorma, Itkonen, Markus & Laarni Jari. 2002. Typografia: Kieltä vai visuaalisuutta. Porvoo: WS Bookwell Oy.
- 9 Webtype launched by partners of experts. Verkkodokumentti. TypeNetwork. <<https://fontbureau.typhenetwork.com/news/article/webtype-launched-partnership-experts>>. 12.3.2017. Luettu 12.3.2017.
- 10 10 About Us. 2017. Verkkodokumentti. . International organization for standardization. <<https://www.iso.org/about-us.html>>. 14.3.2017. Luettu 14.3.2017.
- 11 Font hinting. 2010. Verkkodokumentti. Typotheque. <<https://www.typotheque.com/articles/hinting>>. 17.5.2010. Luettu 13.3.2017.
- 12 International paper sizes & formats. 2017. Verkkodokumentti. i56.org. <http://www.papersizes.org/default-tsta.htm?utm_exp=76296969-1367.nCt5EE4yTruz4xJiDLa3oA.1&utm_referrer=http%3A%2F%2Fwww.papersizes.org%2Fdefault-tsta.htm>. 12.3.2017. Luettu 12.3.2017.
- 13 Yannis, Haralambous. 2007. Fonts & encodings. California: O'Reilly media.
- 14 Unicode 9.0.0. 2017. Verkkodokumentti. The unicode consortium. <<http://www.unicode.org/versions/Unicode9.0.0/>>. 21.6.2016. Luettu 15.3.2017.

- 15 Microsoft typography. 1997. Verkkodokumentti. Microsoft.
<<https://www.microsoft.com/typography/TrueTypeFonts.msp>>. 30.6.1997.
Luettu 20.3.2017.
- 16 TrueType fundamentals. 2017. Verkkodokumentti. Microsoft.
<<https://www.microsoft.com/typography/otspec/TTCH01.htm>>. 4.1.2017. Luettu
20.3.2017.
- 17 Empedded OpenType file format. 2008. Verkkodokumentti. World wide web
consortium. <<https://www.w3.org/Submission/EOT/>>. 5.3.2008. Luettu
23.3.2017.
- 18 Typography on the web. 2014. Verkkodokumentti. World wide web consortium.
<https://www.w3.org/wiki/Typography_on_the_Web>. 14.3.2014. Luettu
24.3.2017.
- 19 WOFF file format 1.0. 2012. Verkkodokumentti. World wide web consortium.
<<https://www.w3.org/TR/WOFF>>. 13.12.2012. Luettu 26.3.2017.
- 20 OpenType font variations overview. 2016. Verkkodokumentti. Microsoft.
<<https://www.microsoft.com/typography/otspec180/otvaroverview.htm>>.
17.10.2016. Luettu 30.3.2017.
- 21 Sites with custom fonts. 2017. Verkkodokumentti. HTTP archive.
<<http://httparchive.org/trends.php#perFonts>>. 1.4.2017. Luettu 1.4.2017.
- 22 Web font optimization. 2017. Verkkodokumentti. Google.
<[https://developers.google.com/web/fundamentals/performance/optimizing-
content-efficiency/webfont-optimization?hl=en](https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/webfont-optimization?hl=en)>. 9.2.2017. Luettu 2.4.2017.
- 23 Css font rendering controls module level 1. 2015. Verkkodokumentti. Github.
<<https://tabatkins.github.io/specs/css-font-display/#biblio-css-font-loading-3>>.
10.9.2015. Luettu 2.4.2017.
- 24 Loading web fonts with the web font loader. 2015. Verkkodokumentti. CSS –
tricks. <<https://css-tricks.com/loading-web-fonts-with-the-web-font-loader/>>.
16.11.2015. Luettu 3.4.2017.
- 25 The need for mobile speed: How mobile latency impacts publisher revenue.
2017. Verkkodokumentti. Google.
<<https://www.doubleclickbygoogle.com/articles/mobile-speed-matters/>>.
3.4.2017. Luettu 3.4.2017.
- 26 Mobile web browsing overtakes desktop font the first time. 2016.
Verkkodokumentti. Guardian.
<[https://www.theguardian.com/technology/2016/nov/02/mobile-web-browsing-
desktop-smartphones-tablets](https://www.theguardian.com/technology/2016/nov/02/mobile-web-browsing-desktop-smartphones-tablets)>. 2.11.2016. Luettu 4.4.2017.

- 27 The amazing em unit and other best practices. 2017. Verkkodokumentti. World wide web consortium. <<https://www.w3.org/WAI/GL/css2em.htm>>. 5.4.2017. Luettu 5.4.2017.