



Juho Salmela

# Verkkosivujen tiedonsiirron ja energiatehokkuuden optimointi ekologisesti kestäväen suunnittelun näkökulmasta

Metropolia Ammattikorkeakoulu

Muotoilija

Digitaalisen muotoilun tutkinto-ohjelma

Opinnäytetyö

14.4.2024

## Tiivistelmä

|                         |  |
|-------------------------|--|
| Tekijä(t):              | Juho Salmela   |
| Otsikko:                | Verkkosivujen tiedonsiirron ja energiatehokkuuden optimointi ekologisesti kestävästä suunnittelun näkökulmasta |
| Sivumäärä:              | 58 sivua + 5 liitettä  |
| Aika:                   | 14.4.2024  |
| Tutkinto:               | Muotoilija   |
| Tutkinto-ohjelma:       | Digitaalisen muotoilun tutkinto-ohjelma  |
| Suuntautumisvaihtoehto: | Digitaalinen muotoilu  |
| Ohjaaja(t):             | Lehtori Markus Norrena   |

---

Opinnäytetyö keskittyy digitaalisten palveluiden suunnitteluun käyttäjien tasa-arvon ja ekologisen kestävyysnäkökulmasta. Työssä kartoitetaan, miten verkkosivuja voidaan suunnitella niin, että ne latautuvat nopeasti ja samalla toimivat energiatehokkaasti. Energiatehokkuuden parantaminen on tärkeää ympäristön kannalta, sillä se vähentää hiilidioksidipäästöjä ja tuottaa kustannussäästöjä. Työssä tarkastellaan konkreettisesti, miten verkkosivujen osatekijöiden optimointi vaikuttaa energiankulutukseen ja tiedostokokoon.

Suunnittelun merkitys ekologisesti kestävästä verkkosivun luomisesta on keskeistä, sillä hyvä informaatioarkkitehtuuri ja käyttäjäkokemus voivat vähentää toteutuksen myöhempien käytäntöjen vaikutusta ympäristöhaittoihin. Verkkosivujen koon optimointi on myös erittäin olennaista tiedonsiirron kannalta.

Opinnäytetyössä suoritettiin kattava joukko testimittauksia tiedonsiirron yleiseksi analysoimiseksi ja verkkosivujen energiatehokkuuden arvioimiseksi. Testitulosten analysoinnissa havaittiin vaihteluita virrankulutuksessa ja sivujen tehokkuudessa mobiililaitteella.

Verkkosivujen suunnittelussa hyvä suorituskyky ja alhainen tehonkulutus ovat yhteydessä toisiinsa. Opinnäytetyö osoittaa, että kuvien optimoinnilla ja uusien formaattien käytöllä voidaan vähentää tiedonsiirtokapasiteetin tarvetta ja siten säästää energiaa.

Avainsanat: optimointi, verkkosivut, kestävä suunnittelu

---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author(s): Juho Salmela  
Title: Optimizing Website Data Transfer and Energy Efficiency from an Ecologically Sustainable Design Perspective  
Number of Pages: 58 pages + 5 appendices  
Date: 14 April 2024

Degree: Bachelor of Culture and Arts  
Degree Programme: Design  
Specialisation option: Digital Design  
Instructor(s): Markus Norrena, Senior Lecturer

---

The study focuses on the design of digital services from the perspectives of user equity and ecological sustainability. It explores how websites can be designed to load quickly while also operating efficiently in terms of energy consumption. Improving energy efficiency is crucial for the environment as it reduces carbon dioxide emissions and generates cost savings. The study specifically examines how optimising various elements of websites affects energy consumption and file size.

The significance of design in creating environmentally sustainable websites is paramount, as good information architecture and user experience can mitigate the impact of implementation practices on environmental harm. Optimising website asset file size is also crucial in terms of minimising data transmission.

A comprehensive set of test measurements was conducted in the study to analyse data transmission and evaluate the energy efficiency of websites. The tests utilised a mobile device to evaluate digital content's impact on power consumption and page efficiency. Results showed varying power consumption rates, with screen tests indicating differences between light and dark screens. Fonts, CSS files, images, audio, animations and video were optimised. Comparisons between SVG animation and AVC video highlighted SVG's efficiency. Overall, tests revealed significant variations in power consumption and efficiency across mobile versions.

In website design, good performance and low power consumption are interconnected. The study demonstrates that optimising images and utilising new formats can reduce the need for data transmission capacity and thus save energy.

Currently the file size and processing power need of websites is growing. As designers and developers become more aware of the energy consumption associated with digitalization, we can hopefully expect a shift in the current trend. Ensuring energy efficiency and minimising file sizes should become integral parts of the web service development process. The tools are already available; they just need to be utilised.

Keywords: Optimisation, websites, sustainable design

## Sisällys

|     |   |    |
|-----|---|----|
| 1   | Johdanto                                | 1  |
| 2   | Keskeisten termien määritelmiä          | 4  |
| 2.1 | Kestävä kehitys ja ekologinen kestävyys | 4  |
| 2.2 | Hidas, raskas, kevyt ja nopea           | 5  |
| 2.3 | Häviötön ja häviöllinen pakkaus         | 5  |
| 3   | Suunnittelu ja optimointi               | 5  |
| 3.1 | Suunnittelu                             | 5  |
| 3.2 | Datan siirto ja virrankulutus           | 6  |
| 3.3 | Verkkosivujen koko                      | 6  |
| 4   | Eri aineistotyyppien optimointi         | 8  |
| 4.1 | Näyttö ja värit                         | 8  |
| 4.2 | Kirjasimet ja kirjasintyytit            | 9  |
| 4.3 | Cascading Style Sheets (CSS)            | 13 |
| 4.4 | Kuvien optimointi                       | 16 |
| 4.5 | Äänen optimointi                        | 18 |
| 4.6 | Animaatiot                              | 19 |
| 4.7 | Videot                                  | 20 |
| 4.8 | Ohjelmointikielen vaikutus              | 21 |
| 5   | Esimerkkitoteutukset                    | 21 |
| 5.1 | Testauksessa käytetty verkkosivupohja   | 21 |
| 5.2 | Hidas ja raskas toteutus                | 22 |
| 5.3 | Nopea ja kevyt toteutus                 | 22 |
| 6   | Testimittaukset                         | 23 |
| 6.1 | Mittausjärjestelyt                      | 23 |
| 6.2 | Virrankulutuksen lähtötason mittaus     | 24 |
| 6.3 | Näyttö                                  | 24 |
| 6.4 | Kirjasimet                              | 27 |
| 6.5 | Cascading Style Sheets (CSS)            | 28 |
| 6.6 | Kuvat                                   | 28 |
| 6.7 | Äänitiedostojen luonti                  | 31 |

|      |   |           |
|------|---|-----------|
| 6.8  | Animaation ja videon vertailu                         | 33        |
| 6.9  | Hitaan ja raskaan sivun mittaus                       | 35        |
| 6.10 | Nopean ja kevyen sivun mittaus                        | 41        |
| 6.11 | Erittäin nopean ja kevyen sivun mittaus               | 45        |
| 7    | Testitulosten analysointi                             | 48        |
| 8    | Johtopäätökset - mitä suunnittelija voi tehdä?        | 51        |
| 9    | Yhteenveto  | 53        |
|      | <b>Lähteet</b>  | <b>55</b> |
|      | <b>Liitteet</b>                                       | <b>59</b> |
|      | MediaInfo-työkalun antamat tiedot videotiedostosta    | 59        |
|      | Termit ja lyhenteet                                   | 60        |
|      | .htaccess tiedostoon lisätyt rivit:                   | 62        |
|      | Hitaan ja raskaan toteutuksen HTML-lähdekoodi         | 63        |
|      | Erittäin nopean ja kevyen toteutuksen HTML-lähdekoodi | 72        |

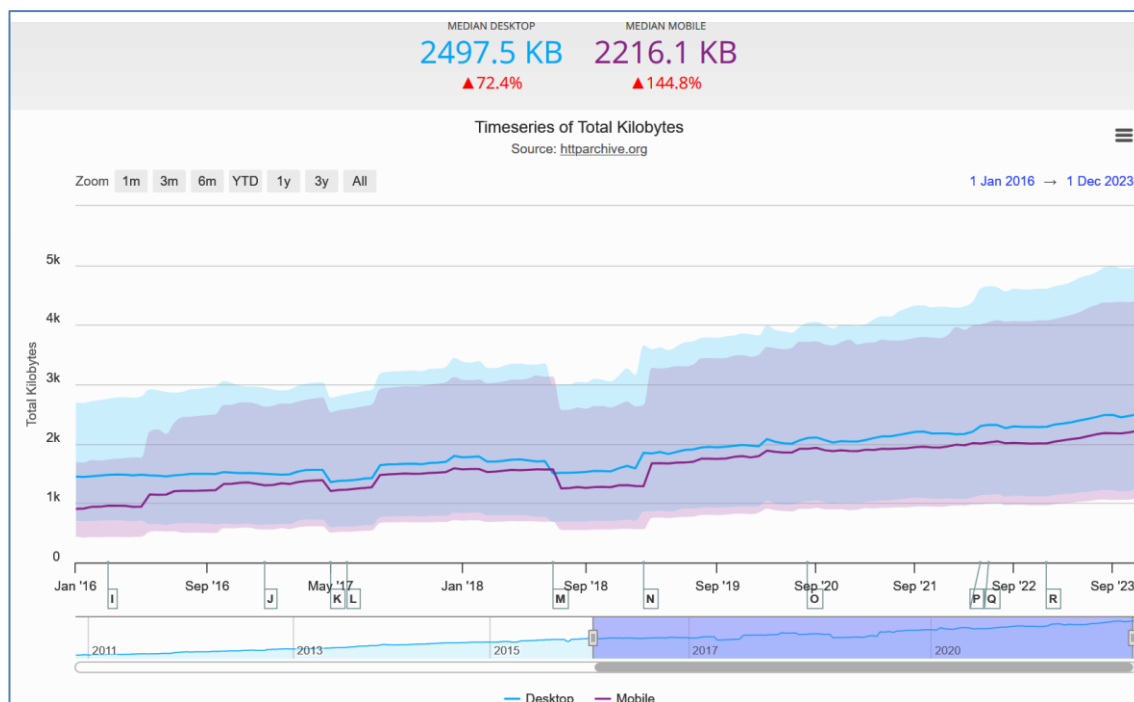
# 1 Johdanto

Opinnäytetyöni tekemisen motiivina toimii pyrkimys tasa-arvoon ja ekologisuuteen digitaalisten palveluiden toteuttamisessa. Opinnäytteeni käsittelee sitä, kuinka suunnittelija voi suunnitella verkkosivuja, jotka latautuvat mahdollisimman nopeasti ja toimivat samalla ekologisesti ja energiatehokkaasti. Energiatehokkuuden parantaminen on tärkeää, sillä se vähentää hiilidioksidipäästöjä, pienentää energiankulutusta ja tuottaa kustannussäästöjä (Energiavirasto, 2024).

Vaikka alun perin toivottiin, että toimintojen digitalisointi säästäisi energiaa, digitalisaatio on kuitenkin luonut uusia energiantarpeita, kuten palvelinten tarvitsemat datakeskukset. Lähteiden mukaan digitalisaatio kokonaisuudessaan lisää energian ja muiden resurssien kulutusta. Korkeampi digitalisaation aste syö digitalisaation alkuperäisen positiivisen vaikutuksen korkeamman energian- ja resurssienkulutuksen kautta. (Lange, Pohl & Santarius 2020; Ahmadova, Delgado-Márquez, Pedauga & Leyva-de la Hiz 2020.)

Verkkosivujen toteutus on sekä ympäristö- että tasa-arvokysymys, sillä halvemmat päätelaitteet ja myös vähävaraisten saatavilla olevat edullisemmat yhteydet vaativat siedettävän käyttökokemuksen aikaansaamiseksi kevyitä ja nopeita verkkosivuja. Jos tavoitteena on saavuttaa globaali yleisö, eikä vain rikkaiden maiden hyväosaiset, verkkosivujen suorituskykyyn on pakko kiinnittää huomiota. Verkkosivujen on toimittava muissakin kuin uusimmissa ja kalliimmissa laitteissa.

Hyvä käyttökokemus on ymmärrettävä, helppokäyttöinen, nopea, toimiva vikatilanteissa ja helposti opittava. Nopeat ja kevyet verkkosivut ovat myös käytettävyyden kannalta hyvä asia. Liian hitaat verkkosivut tuntuvat palvelun käyttäjistä huonosti toimivilta, jopa viallisilta. Toisaalta mitä vähemmän dataa siirretään ja sähköä kulutetaan, sitä vähemmän ympäristöä rasitetaan. Trendi on tällä hetkellä näitä pyrkimyksiä vastaan, esimerkiksi mobiilisivujen latauskoko on yli kaksinkertaistunut vuosien 2016–2023 välillä, ks. kuva 1 (HTTP Archive i.a.).



Kuva 1. Vuodesta 2016 mediaanityöpöytäisivuston tiedostokoko on kasvanut 72,4 % ja mobiilisivuston koko 144,8 %

Verkkosivut ja -palvelut vaativat toimiakseen palvelimen, ja nämä palvelimet sijaitsevat useimmiten suurissa datakeskuksissa, jotka voivat olla joko kaupallisia tai julkisten toimijoiden omistamia. Yhdessä datakeskuksessa voi olla tuhansia palvelimia, joissa toimii suuri määrä verkkosivuja ja sovelluksia. Datakeskuksissa toimivat myös sisällönjakeluverkot, jotka ovat maantieteellisesti hajautettu joukko palvelimia, jotka välimuistittavat sisältöjä lähelle käyttäjiä. Sisällönjakeluverkko mahdollistaa internet-aineistojen kuten HTML-sivujen, JavaScript-tiedostojen, CSS-tyylitiedostojen, kuvien ja videoiden nopean lähetyksen käyttäjälle. (Cloudflare, Inc., 2024)

Yksi datakeskus voi kuluttaa yli 60 000 asukkaan kaupungin verran sähköä. Maailmassa on tällä hetkellä (v. 2023) arviolta n. 8000 datakeskusta (Vardhman 2023), jotka kuluttavat 240–340 terawattitunnin (TWh) verran sähköä eli noin 1–1,3 % maailman vuotuisesta sähkötarpeesta (vuonna 2022), huolimatta tehokkuuden suhteen otetuista edistysaskelista (International Energy Agency i.a.). Tähän päälle tulee vielä kryptovaluuttojen kuluttama sähkö, joka oli noin 110 TWh vuonna 2022, eli noin 0,4 % maailman sähkötarpeesta. Suuria datakeskuksia

rakennetaan yhä lisää ja palveluiden siirtyminen pilveen jatkuu. Aivan viime aikoina myös omiin konesaleihin on osittain palattu, esimerkiksi projektinhallinta-ohjelmistoa tuottava Basecamp lopetti pilvipalveluiden käytön ja siirtyi käyttämään omia konesalejaan (Basecamp i.a.).

Varsinainen tutkimuskysymykseni on: Kuinka verkkosivujen osatekijöitä optimoidaan tiedostokoon ja energiankulutuksen pienentämiseksi? Lisäksi apukysymyksenä toimii: Mitkä muut asiat vaikuttavat verkkosivujen kokoon ja sähkönkulutukseen? Tarkastelen asiaa käytännön kautta, optimoimalla mm. kuvia ja ääniä ja mittaamalla kuinka paljon energiaa tai dataa ne kuluttavat ennen optimointia ja optimoinnin jälkeen. Mittauksiin ja optimointiin käytetään olemassa olevia, yleisesti käytettyjä ohjelmistoja ja työkaluja.

Tässä opinnäytetyössä suunnittelulla tarkoitetaan niitä asioita, joiden kanssa UI/UX-suunnittelija tai käyttöliittymäsuunnittelija on perinteisesti ollut suoraan tekemisissä, kuten värit, kuvat, kirjasimet, ulkoasu, videot, animaatiot ja ääni, eikä niinkään ohjelmointikielten tai -kehikon vaikutusta. Raja suunnitellun toiminnallisuuden ja teknisen toteutuksen välillä on häilyvä ja osittain keinotekoinen. Samaan toiminnalliseen lopputulokseen pääsee teknisesti useilla eri tavoilla, esimerkiksi eri ohjelmointikielen valitsemalla. Ohjelmointikielen ja ohjelmointikehikon (framework) valinnalla on merkittävä vaikutus palvelinten sähkönkulutukseen (Pereira ym. 2017), mutta myös päätelaitteen kuten puhelimen kuormitukseen. Ohjelmointikielten ja -kehikoiden vaikutus energiankulutukseen ja tiedoston siirtokokoon on kuitenkin rajattu pois tästä opinnäytetyöstä.

Datan siirtotien tyyppi vaikuttaa erittäin merkittävästi siihen kuinka paljon sähköä verkkosivujen lataaminen kuluttaa. Valokuitu on tehokkain tapa siirtää dataa. Mobiilitiedonsiirto on jopa satoja kertoja tehottomampaa kuin siirto valokuidulla. (Nuutinen 2021) Siirtotien vaikutus on rajattu pois tästä opinnäytetyöstä.



## 2 Keskeisten termien määritelmiä

### 2.1 Kestävä kehitys ja ekologinen kestävyys

Maailmanpankin pääjohtaja Ismail Serageldin on sanonut: "Kestävä kehitys tarkoittaa sitä, että jätämme tuleville sukupolville yhtä paljon mahdollisuuksia kuin meillä on ollut, elleimme jopa enemmän." Tulevaisuuden mahdollisuudet voidaan ajatella pääomina, joita on erityyppisiä. Suomessa Valtion taloudellinen tutkimuslaitos on jakanut pääomat neljään eri lajiin. Inhimilliseen pääomaan kuuluvat esim. osaaminen, tiede, tutkimus ja kehitys, patentit. Fyysinen pääoma koostuu esim. tuotantokoneistoista, infrastruktuurista ja rakennetusta ympäristöstä. Sosiaalinen pääoma sisältää esim. lainsäädännön, hallinnon, sosiaaliset verkostot, luottamuksen ja legitimitietin. Luontopääomaan kuuluvat uusiutuvat ja uusiutumattomat luonnonvarat.

Ekologisen kestävyuden periaatteina toimivat muun muassa biologisen monimuotoisuuden ja ekosysteemien toimivuuden säilyttäminen, ihmisen taloudellisen toiminnan sopeuttaminen luonnon kestävyyskykyyn, haittojen synnyn ennalta estäminen ja haittojen torjuminen niiden syntyilähteillä (Suomen ympäristöministeriö 2023).

Opinnäytetyössä keskitytään erityisesti ekologiseen kestävyyskykyyn sekä luontopääomaan ja sen säilyttämiseen ja kartoitetaan sitä, kuinka tekniset ratkaisut vaikuttavat tietoteknisten järjestelmien aiheuttamaan sähkönkulutukseen. Pääpaino opinnäytetyössä on verkkosivujen suunnitteluperiaatteiden vaikutuksella toteutuksen energiatehokkuuteen. Pyrkimyksenä on löytää tapoja optimoida sähkönkulutusta ja siirrettävän datan määrää. Tavoitteena on haittojen synnyn minimointi ja ennalta estäminen.

## 2.2 Hidas, raskas, kevyt ja nopea

Hidas tässä opinnäytetyössä merkitsee sellaista verkkosivujen toteutusta, joka sisältää suuria tiedostoja ja vaatii paljon tiedonsiirtokapasiteettia. Raskaat verkkosivut vaativat paljon prosessoritehoa ja ne kuluttavat siten paljon sähköä.

Nopea verkkosivu on tiedostokooltaan pieni ja kuluttaa siten vähän tiedonsiirtokapasiteettia. Kevyt on termi, joka merkitsee toteutusta, jonka prosessointi kuluttaa vähän sähköä.

## 2.3 Häviötön ja häviöllinen pakkaus

Häviötön pakkaus on pakkaamista, jossa kaikki tieto säilyy.

Häviöttömän pakkaamisen jälkeen data on mahdollista palauttaa täysin alkuperäiseen muotoonsa. Häviötöntä pakkaamista käytetään silloin, kun on tärkeää pystyä tuottamaan pakatusta tiedosta täysin alkuperäisen kaltainen tieto. (TEPA-termipankki, 2007)

Häviöllisessä pakkauksessa käytön kannalta epäolennaista tietoa poistuu.

Häviöllisen pakkaamisen jälkeen dataa ei ole mahdollista palauttaa täysin alkuperäiseen muotoonsa. Pakatessa esimerkiksi kuvan tai äänen laatu hieman heikkenee, mutta tietoa pyritään poistamaan siten, ettei silmä tai korva sitä huomaa. Häviöllisellä pakkaamisella ääni- ja kuvatiedostot saadaan pakattua murto-osaan alkuperäisestä koostaan. (TEPA-termipankki, 2007)

# 3 Suunnittelu ja optimointi

## 3.1 Suunnittelu

Informaatioarkkitehdit ja käyttäjäkokemussuunnittelijat voivat vaikuttaa huomattavasti verkkosivun aiheuttamiin ympäristöhaittoihin. Vaikka kestävyys jätettäisiin täysin huomiotta projektin myöhemmissä vaiheissa ja se johtaisi raskaaseen muotoiluun, tehottomaan koodiin ja pahiten saastuttavaan sisällön ylläpitoon (hosting), hyvä informaatioarkkitehtuuri ja käyttäjäkokemus voivat vähentää näiden myöhempien päätösten vaikutusta. Mitä vähemmän huomiota kiinnitetään

kestävyyteen web-projektin myöhemmissä vaiheissa, sitä tärkeämpää on suunnitella tehokkaita ratkaisuja projektin alkuvaiheessa myöhempien vaiheiden huonojen ratkaisujen vaikutuksen vähentämiseksi (Greenwood 2021).

Suunniteltaessa verkkosivuja on hyvä kiinnittää huomiota siihen, mitä mediaa käytetään tiedon välittämiseen. Onko video tarpeellinen vai riittäisikö yksinkertainen animaatio? Tarvitaanko animaatiota vai riittäisikö muutama kuva tai selkeä käyttöohje?

### 3.2 Datan siirto ja virrankulutus

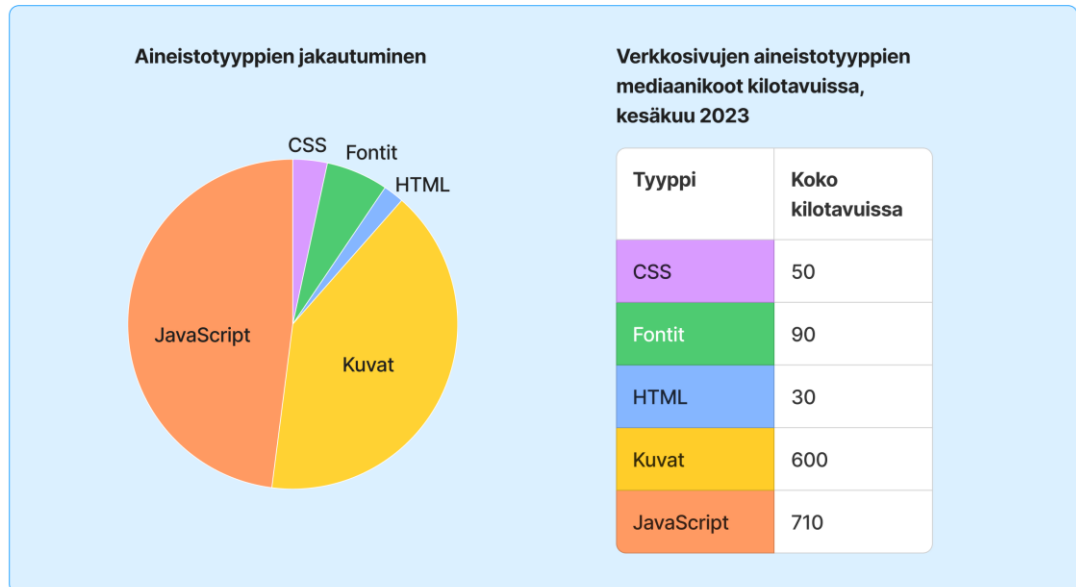
Uudet datan pakkausmuodot vievät huomattavasti vähemmän tilaa kuin edellisen sukupolven formaatit. Valitsemalla oikea pakkausalgoritmi kutakin datatyyppiä varten ennen sen lähetystä, voidaan säästää huomattava määrä energiaa (Piątkowski, Puślecki & Walkowiak 2024).

Tiedonsiirron kannalta tämä tarkoittaa vähemmän siirrettävää dataa ja nopeampia latausaikoja. Kuitenkin esimerkiksi kuvien tapauksessa vanhempien JPEG- ja WEBP-muotoisten kuvien näyttäminen vie vähemmän energiaa vastaavalla kuvanlaadulla kuin AVIF-muoto, mutta ne vievät myös enemmän tilaa (Irani, 2023). Videoiden tapauksessa uudemmat ja tehokkaammat koodekit kuten AV1 kuluttavat enemmän sähköä (Chachou, Hamidouche, Fezza & Belalem 2023).

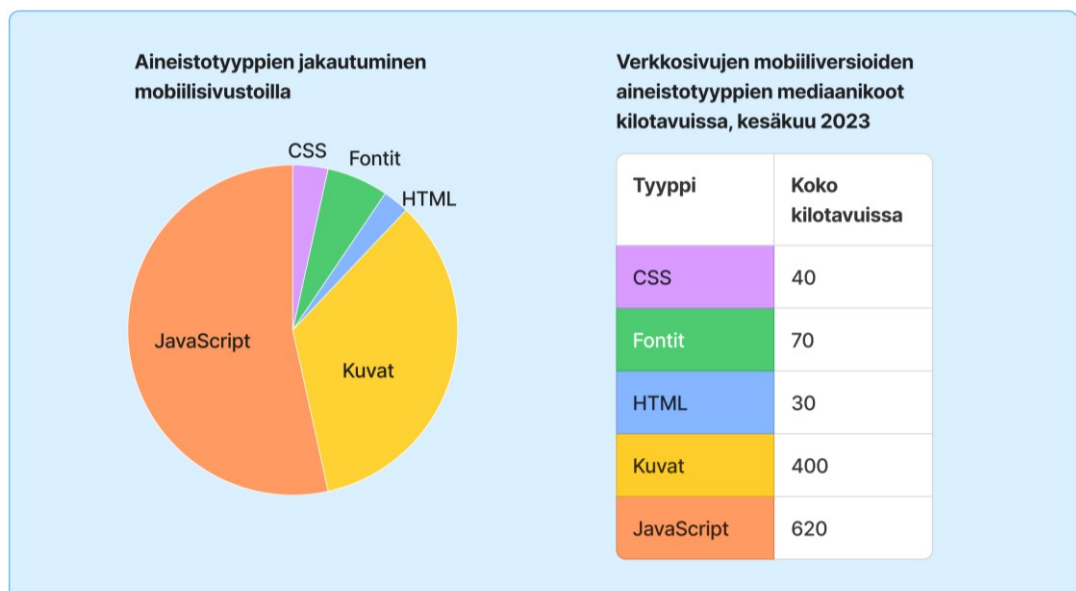
### 3.3 Verkkosivujen koko

Kesäkuussa 2023 100 000 suosituimman verkkosivun työpöytäversion mediaanikoko oli 2100 kilotavua ja mobiiliversioiden koko 1800 kilotavua. Kirjasimia käytettiin mediaanisivuston työpöytäversiossa kolmea erilaista perustuen palvelinpyyntöihin, joiden tiedostotyyppi on eot, ttf, woff, woff2 tai otf tai joiden MIME-tyyppi sisältää sanan "font". Mediaanisivuston mobiiliversiossa kirjasimia käytettiin kahta erilaista. Kuvia mediaanityöpöytäversiossivustolla oli 32 ja mobiiliversiossassa 28 pe-

rustuen palvelinpyyntöjen tiedostotyyppeihin ja MIME-tyyppeihin. JavaScriptia laddattiin mediaanityöpöytä sivustoilla 710 kilotavua peräti 27 eri tiedostosta ja mobiiliversiossa 620 kilotavua 24 eri tiedostosta. (HTTP Archive i.a.)



Kuva 2. Verkkosivujen aineistotyyppien jakautuminen työpöytäversiossa



Kuva 3. Verkkosivujen aineistotyyppien jakautuminen mobiiliversiossa

## 4 Eri aineistotyyppien optimointi

### 4.1 Näyttö ja värit

Päätelaitteen kuormitus on merkittävä tekijä kokonaispäästöissä, ja varsinkin sen näyttö vastaa suuresta osasta päätelaitteen energiankulutusta; jopa 70 % kokonaiskulutuksesta johtuu näytöstä (Moreau 2023). Nykyaikaisilla OLED-näyttöillä musta väri kuluttaa vähiten sähköä, vanhemmilla LCD-näyttöillä väreillä taas ei ole väliä, sillä näytön taustavalo on aina päällä taikka musta väri kuluttaa vähemmän (IO-tech 2021).

OLED-näyttöillä väreistä valkoinen syö eniten sähköä, sininen toiseksi eniten. Sininen pikseli ruudulla käyttää sähköä 25 % enemmän kuin punaiset tai vihreät pikselit (Sustainable Web Design 2024). Ns. dark mode eli tummia värejä käytävä tumma tila on edullinen energiankulutuksen kannalta. Sen käyttö voi vähentää sähköenergiankulutusta jopa 63 %. (Tung 2018.) Vuoden 2023 ensimmäisellä neljänneksellä noin puolessa myydyistä puhelimista on OLED-näyttö ja 94 % puhelimista, joiden kuluttajahinta on yli 250 dollaria, sisältää OLED-näytön (Rastogi 2023).

CSS:n prefers-color-scheme mediaominaisuudella voi määrittää erilaiset tyylit verkkosivulle sen mukaan kumpi teema-asetus on päällä käyttäjän käyttöjärjestelmässä, vaalea vai tumma. Kuvassa 4 esimerkki tumman teeman käytöstä CSS:n avulla:

```
.theme {  
  background: ■ rgb(243, 243, 243);  
  color: □ rgb(0, 0, 0);  
}  
@media (prefers-color-scheme: dark) {  
  .theme.dark {  
    background: □ rgb(19, 19, 19);  
    color: ■ rgb(239, 239, 239);  
  }  
}
```

Kuva 4. Esimerkki tumman teeman määrittämisestä CSS:n avulla

## 4.2 Kirjasimet ja kirjasintyypit

Kaikkein tehokkain tapa optimoida kirjasimet verkkosivulla on käyttää nk. web-turvallisia kirjasimia (web-safe font). Nämä kirjasimet löytyvät käyttöjärjestelmän tyypistä riippumatta (Windows, macOS, yleisimmät Linux-jakelut, Android ja iOS) valmiiksi asennettuna päätelaitteelta, jolloin niitä ei tarvitse erikseen ladata verkon kautta. Valitettavasti tämä kirjasinvalikoima on hyvin rajoittunut, se sisältää kirjasimet Arial, Courier New, Georgia, Times New Roman ja Verdana. (Mozilla, 2024a)

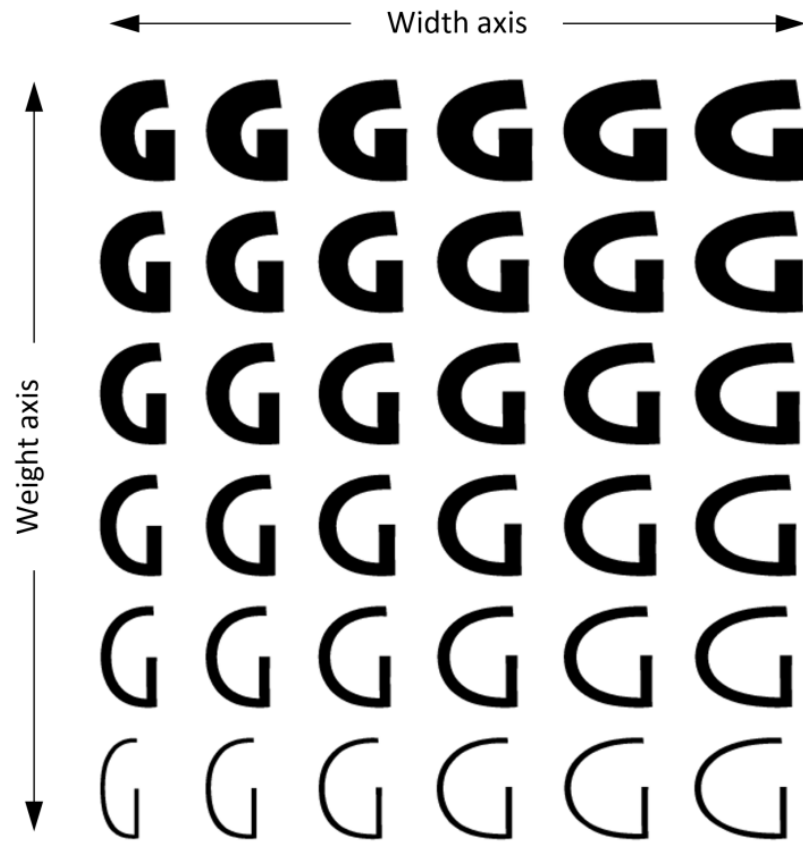
Jos kuitenkin halutaan käyttää omia kirjasimia, on otettava huomioon, että kirjasimet voivat lisätä merkittävästi siirretyn datan määrää. Datamäärä riippuu siitä, onko kirjasintiedostoista poistettu tarpeettomat merkistöt ja merkit ja kuinka monta kirjasintyyppiä on käytetty. Toisinaan kirjasinten lisenssi kieltää merkkien poistamisen kirjasintiedostosta.

Poistamalla kirjasintiedostosta tarpeettomat merkit ja käyttämällä samaan aikaan CSS-spesifikaation unicode-range-ominaisuutta voidaan saavuttaa säästöjä siirretyn datan määrässä. Unicode-rangen avulla voi määrittellä tietyistä kirjasintiedostosta ladattavat merkit, ja jos merkkejä ei käytetä sivun HTML-koodissa, kirjasintiedostoa ei ladata ollenkaan. Tämä on hyödyllistä, jos halutaan näyttää esi-

merkiksi vain artikkelin otsikko tietyllä kirjasimella, näin voidaan välttää kokonaisen kirjasintiedoston lataus vain tätä otsikkoa varten. Prosessi toimii siis niin, että kirjasintiedostosta poistetaan muut kuin otsikossa käytettävät merkit ja unicode-range CSS-määrittelyyn lisätään vain otsikossa käytettävät merkit. Testimittauksissa on käytetty tätä optimointitekniikkaa. (Mills & Weyl 2024; Bhattacharya, Chen & Willee 2024)

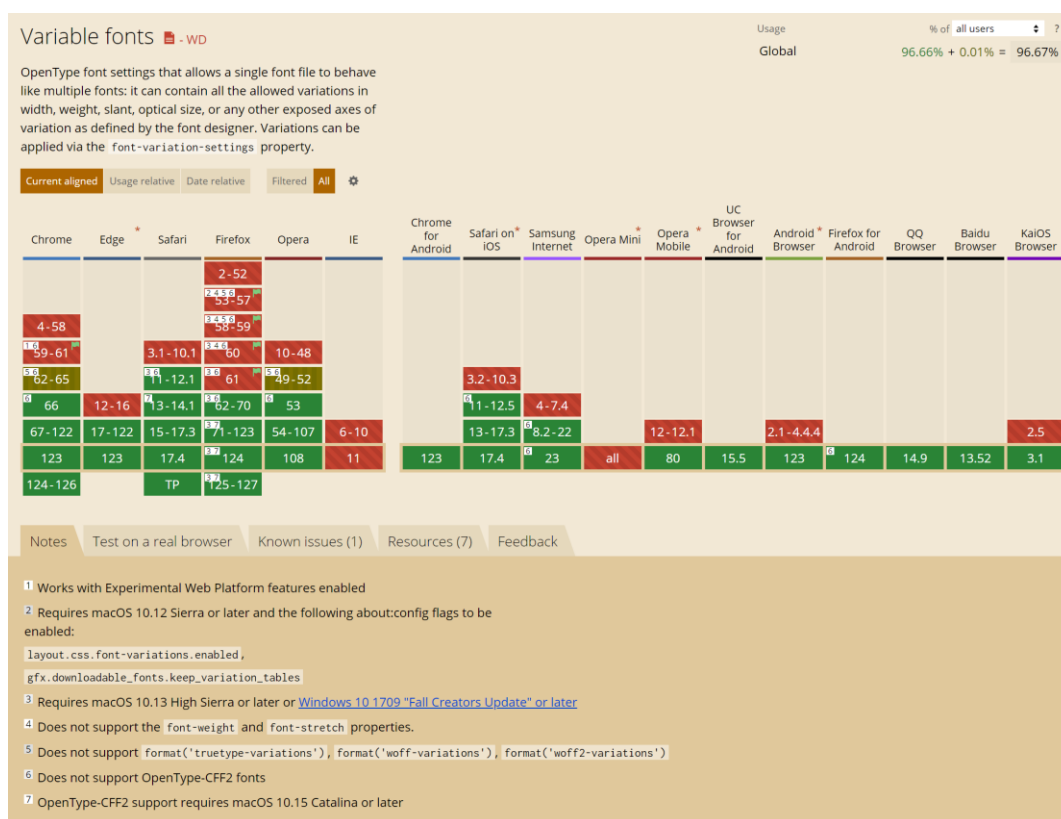
WOFF- ja WOFF2-formaatit ovat internetkäyttöä varten optimoituja kirjasinformatteja, joita kannattaa käyttää mieluummin kuin vanhempia TTF- ja OTF-formaatteja, sillä WOFF-formaatissa olevat tiedostot pakkautuvat paremmin tiedon siirtoa ajatellen ja latautuvat nopeammin.

Lisäksi voidaan käyttää nk. variable-kirjasinta, jossa lihavointi ja leveys on sisällytetty samaan tiedostoon, ks. kuva 5 (Constable, Turetzky, Nihar & Jacobs, 2022). Variable-kirjasinten tuki vaihtelee käyttöjärjestelmän version ja selaimen mukaan, ks. kuva 6. Variable-kirjasimista voidaan poistaa tietyn sivuston kannalta tarpeettomia paksuuksia, leveyksiä tai kallistuksia esimerkiksi Slice-työkalun avulla, joka on vapaan lähdekoodin fonttieditori. Näin voidaan kutistaa kirjasintiedosto helposti esimerkiksi puoleen alkuperäisestä koostaan.



Kuva 5. Variable-fontin kuvaus leveys- ja lihavointiakselilla





## Kuva 6. Variable-kirjasinten tuki eri selaimissa

Suosittua Google Fonts -palvelua käytettäessä on myös mahdollista ladata vain tarpeelliset osajoukot kustakin kirjasintiedostosta (Google 2022). Tämä tapahtuu määrittelemällä subset-parametrille arvo, kuvassa 7 esimerkki kyrillisten kirjainten lataamisesta Roboto Mono -kirjasimen tapauksessa.

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto+Mono&subset=cyrillic">
```

## Kuva 7. Kyrillisten kirjainten lataaminen Google Fonts -palvelusta

Myös yksittäisten tekstinkappaleiden lataaminen on mahdollista text-parametrin avulla, kuvassa 8 esimerkki kuinka ladataan teksti "Hello World" Inconsolata-kirjasimelle:

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Inconsolata&text=Hello%20World">
```

## Kuva 8. Pelkästään Hello World-tekstin lataaminen Google Fonts -palvelusta

## 4.3 Cascading Style Sheets (CSS)

CSS-tiedostot määrittävät verkkosivun ulkoasun, siinä missä HTML-koodi määrittää sen rakenteen. Verkkosivun CSS voi olla nk. render-blocking, eli se estää valmiin sivun piirtämisen ruudulle ja käyttäjän vuorovaikutuksen sivun kanssa. Selaimet tarvitsevat sivun piirtämiseen renderointipuun, joka taas koostuu sekä DOM-puusta (Document object model) että CSSOM-puusta (CSS object model).

CSSOM-puun rakentamisen optimoimiseksi ja sivun latausajan pienentämiseksi voi esimerkiksi poistaa ne CSS-tyylimääritykset, jotka eivät ole sivulla käytössä. CSS-tiedostot voi myös modularisoida jakamalla ne useampaan eri tiedostoon ja viittaamalla niihin HTML-koodissa vain tarvittaessa. Käyttämällä yksinkertaisia selektoreita selain lataa CSS:n nopeammin.

Sen sijaan että kirjoittaisit:

```
body div#main-content article.post h2.headline {  
  font-size: 24px;  
}
```

kirjoita:

```
.headline {  
  font-size: 24px;  
}
```

Vältä selektoreita, jotka osuvat liian useisiin elementteihin ja hyödynnä sen sijaan kaskadia. Eli sen sijaan että kirjoittaisit:

```
body * {  
  font-size: 24px;  
}
```

määrittele sen sijaan vaikkapa:

```
main {  
  font-size: 24px;  
}
```

Ensimmäinen esimerkki määrittää jokaikisen body-elementin lapsielementin kirjaskooksi 24 pikseliä. Jälkimmäinen esimerkki määrittää main-pääsisältöalueen ja sen lapsielementtien kirjaskooksi 24 pikseliä. Molemmat ovat laajoja määri-tyksiä, mutta body-elementtiä käyttävä on tarpeettoman laaja.

Esilataa tärkeät aineistot. Voi käyttää `rel="preload"` määritystä muuttaaksesi linkielementit esilataajiksi tärkeille aineistoille. Tämä sisältää CSS-tiedostot, kirjasimet ja kuvat, alla esimerkki:

```
<link rel="preload" href="style.css" as="style" />
```

```
<link rel="preload" href="ComicSans.woff2" as="font" type="font/woff2" crossorigin />
```

```
<link rel="preload" href="bg-image-wide.png" as="image" media="(min-width: 601px)" />
```

Esilatauksen avulla selain hakee aineistot mahdollisimman pian, jolloin ne ovat saatavilla ennen kuin niihin viitataan myöhemmin ladattavassa koodissa. On hyvä esiladata sellaiset resurssit, jotka esiintyvät sivun alussa, jotta käyttäjäkokemus olisi mahdollisimman sujuva. Huomaa, että voit myös käyttää media-attribuuttia luodaksesi responsiivisia esilatauslinkkejä, kuten yllä olevassa esimerkikoodissa on tehty `bg-image-wide.png` kuvatiedoston kohdalla. (Mills & Weyl 2024)

CSS:n voi myös "minifioida" eli pienentää. Minifioinnilla tarkoitetaan prosessia, jossa CSS-tiedoston koneluettavuuden kannalta turhat välimerkit kuten välilyönit ja kappalejaot poistetaan. (Mills & Weyl 2024) Toisinaan myös CSS-muuttujat nimetään uudelleen lyhyemmiksi ja useampi CSS-tiedosto yhdistetään yhteen tiedostoon.

```
1  ∨ body {
2    background-color: #000000;
3  }
4
5  ∨ h1 {
6    color: #9c9c9c;
7  }
8
9  /* Add a black background color to the top navigation */
10 ∨ .topnav {
11   background-color: #1c1c1c;
12   overflow: hidden;
13 }
14
15 /* Style the links inside the navigation bar */
16 ∨ .topnav a {
17   float: left;
18   color: #9c9c9c;
19   text-align: center;
20   padding: 14px 16px;
21   text-decoration: none;
22   font-size: 17px;
23 }
24
25 /* Change the color of links on hover */
26 ∨ .topnav a:hover {
27   background-color: #6a6a6a;
28   color: black;
29 }
30
31 /* Add a color to the active/current link */
32 ∨ .topnav a.active {
33   background-color: #1d1d1d;
34   color: rgb(120, 120, 120);
35 }
36
37 ∨ h1 {
38   font-family: 'OpenSans', sans-serif;
39   font-weight: 1000;
40 }
41
```

Kuva 9. Esimerkki pienentämättömästä CSS-tiedostosta

```
body{background-color:#000}h1{color:#9c9c9c}.topnav{background-color:#1c1c1c;overflow:hidden}.topnav a{float:left;color:#9c9c9c;text-align:center;padding:14px 16px;text-decoration:none;font-size:17px}.topnav a:hover{background-color:#6a6a6a;color:#000}.topnav a.active{background-color:#1d1d1d;color:#787878}h1{font-family:OpenSans,sans-serif;font-weight:1000}
```

Kuva 10. Sama CSS-tiedosto kuin kuvassa 9, nyt pienennettynä (huomaa poistettut kommentit)

Lisäksi on olemassa työkaluja, kuten PurgeCSS, joka pyrkii etsimään käyttämättömiä CSS-määrittäjiä vertaamalla sivuston lopullisia HTML- ja CSS-tiedostoja. PurgeCSS:n kaltaiset työkalut eivät kuitenkaan osaa ottaa huomioon JavaScriptin avulla dynaamisesti luotuja CSS-määrittäjiä, jolloin ne poistavat CSS-tiedostosta määrittäjiä, jotka itseasiassa ovat käytössä.

Kannattaa varmistaa, että palvelimella on käytössä CSS-tiedostojen Gzip- tai Brotli-pakkaus, sillä ne pienentävät tiedostojen kokoa jopa 70–90 %. (Grigorik & Wagner 2023)

#### 4.4 Kuvien optimointi

Kuvien optimointi vaikuttaa pääasiallisesti siirretyn datan määrään pienentämällä siirrettävien kuvien tiedostokokoa. Pienempään tilaan kuvan pakkaava formaatti kuten AVIF vaatii myös hieman enemmän prosessoritehoa ja sitä kautta sähköä kuvan dekodauksen yhteydessä. Kuvan pikselikoko vaikuttaa sen tiedostokokoon huomattavasti. 1920 kertaa 1080 pikseliä kokoinen kuva vie huomattavasti enemmän tilaa kuin 800 kertaa 600 pikseliä kokoinen kuva. Vektorikuvien kuten SVG-formaatissa olevien kuvien tapauksessa päätelaitteelta vaaditaan myös enemmän prosessoritehoa, riippuen kuvan monimutkaisuudesta.

Optimointi voidaan suorittaa ns. ”käsini” säätämällä kunkin kuvan laatuasetuksia tallennusvaiheessa tai esimerkiksi WordPress-sisällönhallintajärjestelmässä erilaisia automaattisesti kuvien laatua säätäviä lisäosia (plugin) hyödyntämällä.

Web-ympäristössä usein käytetty vektorikuvaformaatti Scalable Vector Graphics (SVG), pakkautuu erinomaisesti yksinkertaisten muotojen kuten logojen ja erilaisien geometrinen muotojen tapauksessa. SVG-tiedostot koostuvat matemaattisista yhtälöistä, jotka määrittelevät erilaisia muotoja, viivoja sekä käyriä. SVG-tiedostot voivat myös olla interaktiivisia ja niitä voi animoida CSS:n ja/tai JavaScriptin avulla. SVG-muoto ei sovellu valokuvien esittämiseen, sillä tiedostokoko ja kuvan esittämiseen tarvittava prosessoriteho kasvavat kohtuuttoman suuriksi (Cloudinary 2024).

Responsiivisen kuvasyntaksin avulla voidaan lähettää selaimelle yksi kuva useista eri vaihtoehdoista perustuen eri sääntöihin ja seikkoihin. Tähän käytetään HTML:n srcset-kuvamäärittelyä. Srcset-määrittelyä voidaan käyttää joko <img> tai <picture> -elementin kanssa. Picture sallii varavarmistuksien eli vanhempien kuvaformaattien lähetyksen vanhemmille selaimille.

AVIF-formaatti on uusi, tehokas kuvaformaatti. AVIF suoriutuu johdonmukaisesti selvästi JPEG-formaattia paremmin samalla tiedostokoolla (Ubl 2020). AVIF-tiedostot myös pakkautuvat pienempään tilaan, säilyttävät yksityiskohdat paremmin, niissä on vähemmän laatikkoartefakteja ja värit vuotavat vähemmän tarkkojen reunojen kohdalta (Mavlankar, De Cock, Concolato, Swanson, Moorthy & Aaron 2020). Kaikki vanhemmat selaimet eivät tue tätä formaattia. Tällöin kuvien toimivuus voidaan varmistaa varavarmistuksen (fallback) avulla eli lähettämälle vanhemmalle selaimelle kuvat esim. JPEG-formaatissa. Microsoftin Chromiumiin pohjautuva Edge-selain ei tukenut AVIF-formaattia ollenkaan, kun tämän opinäytetyön kirjoitus aloitettiin, vaikka Chromiumissa on valmiiksi tuki AVIF-kuville, Microsoft siis erikseen poisti tuen tälle kuvaformaatile omasta Edge-selaimestaan. Syy tähän ei ole tarkkaan selvillä. Vihdoin tammikuun 25. päivänä 2024 Microsoft julkaisi Edgestä version 121.0.2277.83, jossa on myös AVIF-tuki (Microsoft 2024).

WEBP, joka on Googlen kehittämä kuvaformaatti, piti korvata JPEG- ja PNG-muotoiset kuvat, mutta sen pakkausalgoritmi ei lopulta ollut niin tehokas kuin

Google antoi ymmärtää (Ginesu, Pintus & Giusto 2012). WEBP-formaatin häviöllisesti pakattujen kuvien piti olla 25–34 % pienempiä verrattuna JPEG-kuviin, joilla oli sama SSIM-laatuindeksi-arvo (Google 2023). Structural similarity index measure (SSIM) on metodi, jolla ennustetaan digitaalisten kuvien ja videoiden ihmisen hahmottamaa laatua.

MozJPG, joka on Mozillan kehittämä JPEG-yhteensopiva JPEG-pakkausalgoritmi, on verrattavissa WEBP:n antamiin tuloksiin (Siipola 2020). WEBP kuitenkin tukee läpinäkyvyyttä alpha-kanavien muodossa (Google 2023).

JPEG XL oli lupaava kuvaformaatti, mutta selaimet eivät koskaan alkaneet tukea sitä ennen kuin uudemmat formaatit yleistyivät.

#### 4.5 Äänen optimointi

MP3- (MPEG-1 Audio Layer III tai MPEG-2 Audio Layer III), Vorbis ja AAC-formaatit (Advanced Audio Coding format) ovat selainten parhaiten tukemat ääniformaatit. Uudempi OPUS-formaatti on myös hyvin tuettu ja tehokas äänimuoto.

Vorbis-formaatin tiedostot ovat yleensä OGG-säilössä. Applen iOS- ja macOS -käyttöjärjestelmien Safari-selain ei tue OGG-säilöä, joka rajaa ulkopuolelle kaikki Apple-käyttäjät.

MP3-formaatilla on paras selaintuki, jopa Internet Explorer versiosta 9 eteenpäin tukee sitä. AAC-formaatin etu on parempi äänenlaatu samalla bittinopeudella MP3-tiedostoihin verrattuna, tämä tarkoittaa siis tiedostokooltaan pienempää tiedostoa samalla äänenlaadulla. AAC-muotoon pakattu ääni kuulostaa paremmalta erityisesti, jos bittinopeus on 128 kilobittiä sekunnissa tai alle. (Brandenburg 1999)

OPUS on häviöllinen pakkausmuoto, joka on suunniteltu koodaamaan tehokkaasti puhetta ja musiikkia samalla formaatilla. Lisäksi se on tarpeeksi vähälatenssinen reaaliaikaiseen interaktiiviseen kommunikaatioon ja tarpeeksi yksinkertainen, jotta edulliset sulautetut prosessorit jaksavat purkaa sillä enkoodatut tiedostot. (Opus 2024) Sökkokuuntelutesteissä OPUS on luokiteltu korkealaatuisemmaksi kuin MP3 ja AAC matalimmista bittinopeuksista aina niin korkeisiin bittinopeuksiin asti, että häviöllisesti pakattuja tiedostoja ei voi enää erottaa alkupe-  
räisistä tiedostoista (Hoene, Valin, Vos, Skoglund 2013).

## 4.6 Animaatiot

Animaatioiden toteuttamiseen on useita tapoja: JavaScriptin avulla animointi, CSS-transitiot ja näiden yhdistelmä. Lisäksi tulevat SVG-animaatiot, GIF-animaatiot, Lottie-animaatiot ja Rive-animaatiot.

CSS-transitiot ovat nopea ja kevyt tapa animoida elementtejä web-sivulla ilman JavaScriptin apua. On myös mahdollista yhdistää CSS-transitiot ja JavaScript, jolloin esimerkiksi animaation sulavat liukumukset ovat mahdollisia. Myös SVG-tiedostot voivat sisältää JavaScriptiä, joka sisältää tiedoston animaatiot. SVG-tiedostoa voi myös animoida SVG-tiedoston ulkopuolella olevalla JavaScriptillä. WordPress-sisällönhallintajärjestelmässä on oletuksena, että SVG-tiedostojen lataamista järjestelmään ei sallita, sillä ne voivat sisältää vahingollista JavaScriptiä.

Lottie-animaatio koostuu koneluettavasta JSON-formaatin tiedostosta ja sitä animoivasta JavaScript-kirjastosta. Rive-animaatio taas on tavukoodiksi pakattu tiedosto, jota animoidaan JavaScriptin avulla. Näistä Riven lisenssi on suljetumpi. GIF-formaatti tukee maksimissaan 256 väriä per tiedosto ja sen animaatiokyky on alkeellisesti toteutettu, mikä johtaa laatuun verrattuna suureen tiedostokoon. GIF-animaatiot ovat tehottomin kaikista näistä toteutuksista.



## 4.7 Videot

Minuutti pakkaamatonta Full HD-laadun videota vaatii 14,93 gigatavua tilaa, mikä vie aivan liikaa palvelintilaa sekä kaistanleveyttä videota ladattaessa. Siksi on välttämätöntä pakata video jollakin videokoodekilla. (Heppell & Bhattacharya 2024)

Videokoodekki H.264 (MP4) tai MPEG-4 Part 10, tunnetaan myös nimellä Advanced Video Coding (AVC), on laajimmin tuettu videoformaatti ja toimii myös vanhoilla Internet Explorer -selaimen versioilla. Suuri osa nykyisin myytävästä kulutajaelektronikasta kykenee myös dekoodaamaan AVC-videotiedostot laitteiston avulla eikä prosessorin avulla, mikä on hyvin energiatehokasta.

Uudemmallalla AOMedia Video 1 (AV1) -formaattilla voidaan saavuttaa jopa 50 % parempi tiedonpakkaus kuin AVC-koodekilla. AV1-formaatin käyttö on lisäksi täysin ilmaista. Sen huonot puolet ovat puutteellinen selaintuki varsinkin Apple-laitteilla ja laitteistoenkoodauksen ja -dekoodauksen puute. Applen laitteilla AV1-muotoa tuetaan vain kalliimmilla laitteilla, joissa on laitteistodekooderi eli esim. puhelimista iPhone 15 Pro -malli ja kannettavista tietokoneista MacBook Pro M3-suorittimella. Koska laitteistotukea ei vielä ole laajalti olemassa, videon enkoodaus AV1-muotoon vie hyvin kauan.

Taulukko 4. Web-selaimissa toimivat videokoodekit

| Koodekin lyhenne    | Koodekin nimi                 | Tuetut säilöt   |
|---------------------|-------------------------------|-----------------|
| <b>AV1</b>          | AOMedia Video 1               | MP4, WebM       |
| <b>AVC (H.264)</b>  | Advanced Video Coding         | 3GP, MP4        |
| <b>H.263</b>        | H.263 Video                   | 3GP             |
| <b>HEVC (H.265)</b> | High Efficiency Video Coding  | MP4             |
| <b>MP4V-ES</b>      | MPEG-4 Video Elemental Stream | 3GP, MP4        |
| <b>MPEG-1</b>       | MPEG-1 Part 2 Visual          | MPEG, QuickTime |

| Koodekin lyhenne | Koodekin nimi        | Tuetut säilöt        |
|------------------|----------------------|----------------------|
| <b>MPEG-2</b>    | MPEG-2 Part 2 Visual | MP4, MPEG, QuickTime |
| <b>Theora</b>    | Theora               | Ogg                  |
| <b>VP8</b>       | Video Processor 8    | 3GP, Ogg, WebM       |
| <b>VP9</b>       | Video Processor 9    | MP4, Ogg, WebM       |

## 4.8 Ohjelmointikielen vaikutus

Ohjelmointikielten energiatehokkuus vaihtelee huomattavasti. Yleisellä tasolla käännettävät ohjelmointikielet ovat energiatehokkaampia kuin virtuaalikonetta hyödyntävät tai tulkittavat kielet. Poikkeus tähän sääntöön on virtuaalikonetta hyödyntävä Java, joka on yhtä energiatehokas kuin käännettävät ohjelmointikielet. Ikävä kyllä suosittu web-ohjelmointikieli JavaScript ja PHP sijoittuivat 27:n ohjelmointikielen vertailuissa sijoille 17 ja 21. (Pereira, Couto, Ribeiro, Rua, Cunha, Fernandes & Saraiva 2017) PHP on suosituin palvelinohjelmointikieli (W3Techs 2024a). JavaScript on suosituin selainohjelmointikieli (W3Techs, 2024b). Tässä opinnäytetyössä ei kartoiteta eri ohjelmointikielten vaikutusta energiatehokkuuteen tai tiedonsiirtoon.

## 5 Esimerkkitoteutukset

### 5.1 Testauksessa käytetty verkkosivupohja

Testeissä verkkosivupohjana toimii HTML5-Boilerplate, joka on nykyaikainen HTML-malline (Larsen & Oliff 2023). Malline sisältää Modernizr-JavaScript-kirjaston, jota käytetään selainten ominaisuuksien selvittämiseen ja progressiiviseen parannukseen (progressive enhancement). Lisäksi malline käyttää normalize.css CSS-nollausta (reset), jonka tarkoitus on tasoittaa selainten välisiä eroja oletustyyliissä. Tämän mallineen päälle rakennetaan kaksi erilaista verkkosivutoteu-

tusta: hidas ja raskas sekä nopea ja kevyt. Molemmat toteutukset näyttävät ulkoisesti samalta: ne sisältävät samat tekstit, kuvat, animaation ja videon. Pohja on koodattu pelkällä HTML:llä, CSS:llä ja animaatioihin tarvittavilla JavaScriptillä. Ohjelmakirjastoja kuten React tai muita ohjelmointikehikoita ei ole käytetty. Liitteenä 4 hitaan ja raskaan sivun HTML-lähdekoodi. Kevyen ja nopean sivun lähdekoodi on lähes samanlainen, vain kuvien, animaation, äänitiedoston ja videon tiedostonimet ovat erilaisia. Liitteenä 5 erittäin nopean ja kevyen toteutuksen HTML.

## 5.2 Hidas ja raskas toteutus

Raskaassa toteutuksessa fonttiedostoja ei ole optimoitu, vaan jokaista kirjaintyyliä (lihavoitu, kursiivi ym.) kohden ladataan yksi fonttiedosto ttf-muodossa. Kuvat on pakattu JPEG-formaatissa parhaalla mahdollisella laadulla (laatuasetus 100). JPEG-kuvat tallennetaan ilman EXIF-, IPTC-, XMP-metadataa ja JPG-kommentteja. Kuvia on yhteensä 9, joista 5 on JPEG-muodossa ja neljä häviöttömässä PNG-formaatissa. PNG-kuvat muunnettiin alkuperäisistä SVG-kuvista. Animaatio on GIF-animaatio. Video on pakattu H.264-formaattiin MP4-säilöön. Ääni on MP3-muodossa.

## 5.3 Nopea ja kevyt toteutus

Kevyessä toteutuksessa kirjasintiedostoista on poistettu tarpeettomat merkistöt ja merkit ja käytössä on ns. variable-kirjasin. Värit on vaihdettu tummiin väreihin, ja sivun tausta on musta. Siinä missä hitaassa ja raskaassa toteutuksessa käytettiin PNG-kuvia, käytetään nyt alkuperäisiä SVG-vektoritiedostoja ja muuten nopeassa ja kevyessä toteutuksessa käytetään AVIF-formaattia. Yhden vektorikuvan värit vaihdettiin mustasta valkoiseksi, jotta se erottuisi taustasta.

Animaatio on toteutettu SVG-animaationa. Video on AV1-formaatissa MP4-säilössä.

## 6 Testimittaukset

### 6.1 Mittausjärjestelyt

Mittauksessa käytettiin päätelaitteena OnePlus Nord CE 2 puhelinta (malli IV2201) jossa on 4500 mAh akku ja 6,43 tuuman AMOLED näyttö 2400x1080 pikselin resoluutiolla. Käyttöjärjestelmänä puhelimessa toimii OxygenOS 13.0, joka perustuu Android-käyttöjärjestelmään (Davies 2023).

Mitattavat sivut sijaitsevat fyysisesti Zonerin webhotellissa Telia Helsinki Data Center -konesalissa Helsingin Pitäjänmäellä. Mittauksen suorituspaikasta Kirkkonummelta kannettavalla tietokoneella ”pingatessa” viiveeksi saatiin keskimäärin 19 millisekuntia.

Mittausten luotettavuuden parantamiseksi kaikki mahdolliset palvelut kytketään pois päältä. Näihin kuuluvat WiFi, NFC, Bluetooth, sijainti ja lähijakaminen. Päälle jää maksupalvelu Wallet (koska sitä ei saa laitettua pois päältä) ja mobiilidata. Puhelimen käyttöjärjestelmä asetetaan tummaan tilaan. Kaikki muut sovellukset paitsi selain suljetaan. Tiedonsiirto puhelimen ja palvelimen välillä tapahtuu matkapuhelinverkon kautta. Kaikki testimittaukset kestävät 30 min. AccuBattery-sovelluksen avulla saadaan lukemat sähkönkulutuksesta milliampeeritunteina (mAh).

Palvelimen asetusten avulla poistetaan selaimen välimuisti käytöstä. Tämä tapahtuu .htaccess tiedoston avulla, sillä Zonerin webhotelli käyttää Apache-palvelinohjelmistoa, jossa Nginx toimii kuormantasaajana ja välimuistina. Liitteessä 3 ovat rivit jotka .htaccess tiedostoon lisättiin.

Mittauksissa hyödynnettiin AccuBattery-sovellusta, joka nolaa sovelluskohtaisen virrankulutustilaston jokaisen akun latauksen jälkeen. AccuBattery luokittelee kaiken virrankulutuksen etualalla olevan sovelluksen aiheuttamaksi. (AccuBattery 2020) AccuBattery:n lähestymistapa soveltuu näihin mittauksiin hyvin,

sillä ne ovat verkkoselainpohjaisia ja selain saattaa hyödyntää erilaisia käyttöjärjestelmän palveluita esimerkiksi dekodatessaan videota internet-sivulla, tällöin virrankulutus tilastoituu oikein.

## 6.2 Virrankulutuksen lähtötason mittaus

Puhelin valmisteltiin samoilla mittausjärjestelyillä kuin luvussa 6.1. Sitten näyttö sammutettiin painamalla lukitusnäppäintä, jonka jälkeen puhelimen annettiin olla 30 minuuttia. Lopuksi luettiin AccuBattery:n virrankulutuskokema. Tämä toistettiin kolme kertaa, lopputulokseksi saatiin, että virtaa kului 30 minuutissa joka kerta tasan 20 milliampeerituntia.

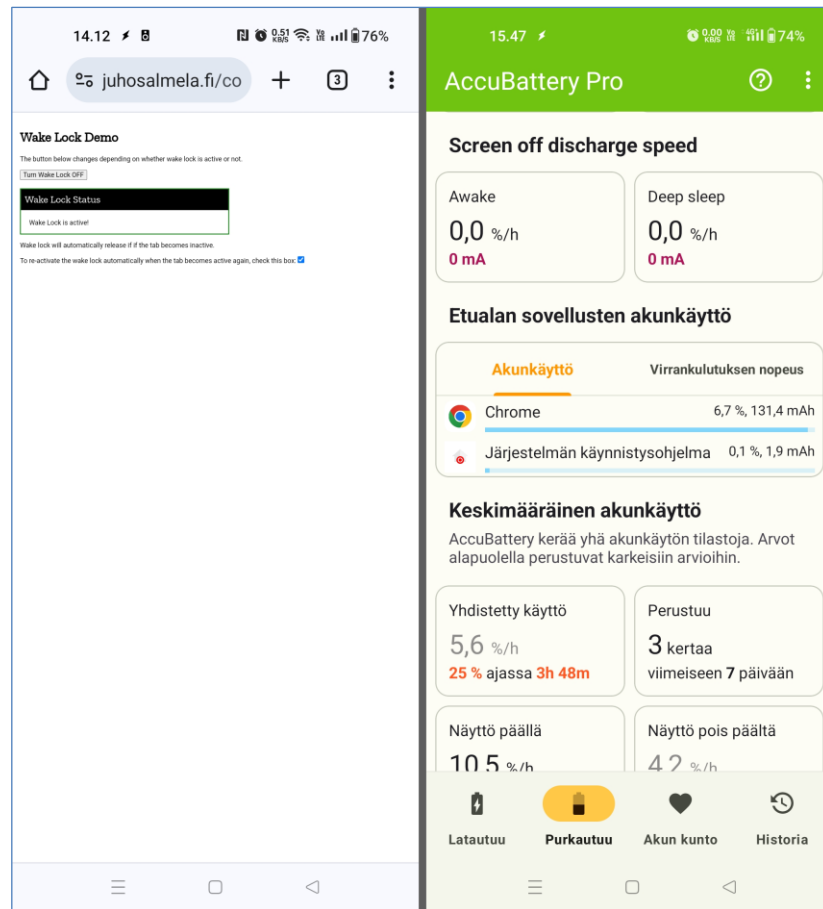
## 6.3 Näyttö

Näytön virrankulutuksen selvittämiseksi, näyttöä pidettiin päällä puoli tuntia täydellä kirkkaudella, aluksi näyttö valkoisena ja sitten mustana. Ero virrankulutuksessa oli huomattava, aivan kuten lähdekirjallisuudesta voi päätellä.

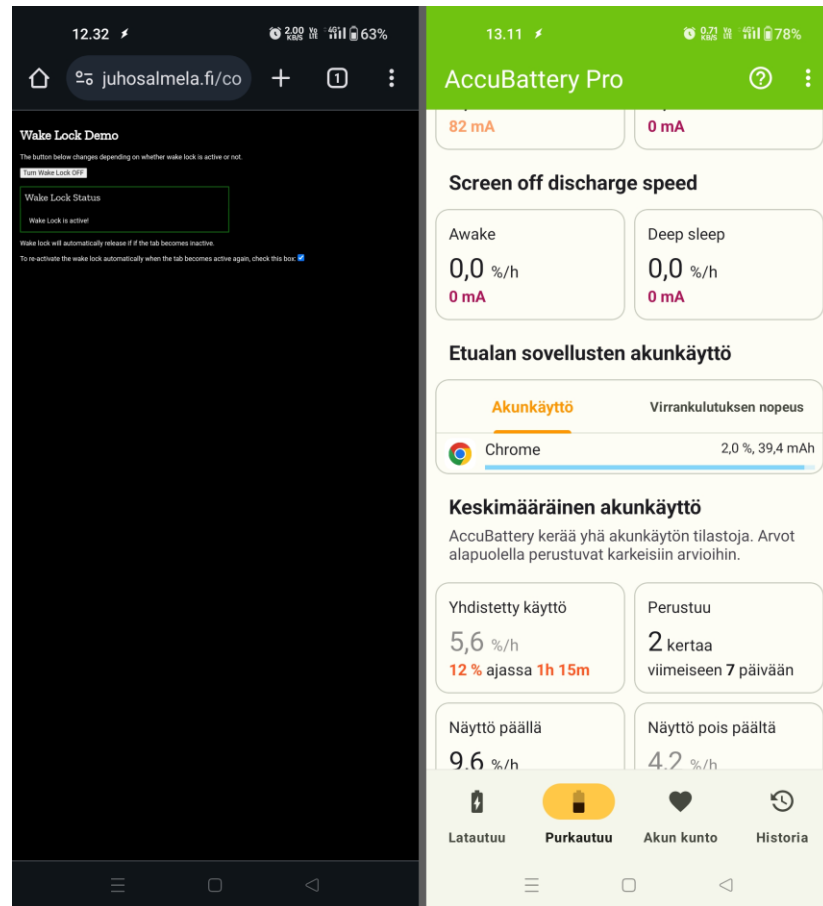
Aluksi tehtiin mittaus valkoisella näytöllä. Puhelin valmisteltiin samalla tavalla kuin virrankulutuksen lähtötason mittauksessa luvussa 6.1.

Selaimena tässä mittauksessa toimi Chrome. Puhelimen käyttöjärjestelmän teemaksi asetettiin vaalea teema ja näytön kirkkaus asetettiin 100 %:n. Tämän jälkeen ladattiin verkkosivu, joka oli miltei kokonaan valkoinen ja pidettiin näyttö päällä 30 minuutin ajan (kuva 8).

Näyttö pidettiin päällä Chrome-selaimen Wake lock -ominaisuuden avulla, hyödyntäen valmista demonstraatiota, jonka tarjosi Mozilla Foundation (Mozilla, 2024b). Demon koodia muokattiin siten, että värit vaihdettiin tätä testiä varten sopiviksi. AccuBattery:n mittauksen mukaan Chrome kulutti sähköä 30 min aikana 131,4 milliampeerituntia (kuva 11).



Kuva 11. Kuvankaappaus sivusta, jota käytettiin valkoisessa testissä ja Accu-Battteryyn virrankulutustilastosta



Kuva 12. Kuvankaappaus mustasta sivusta ja AccuBattery:n virrankulutustilastosta

Seuraavaksi mittaus tehtiin samanlaisella järjestelyllä, mutta käyttäen järjestelmän tummaa teemaa ja mustaa verkkosivua (kuva 12). AccuBattery:n mukaan virtaa kului 39,4 mAh (kuva 12) eli noin yksi kolmasosa siitä mitä valkoisella ruudulla tehdyssä mittauksessa ( $39,4/131,4 = 0,3$ ).

Taulukko 1. Akun virrankulutus tummalla ja valkoisella näytöllä

|                               | Tumma näyttö | Valkoinen näyttö |
|-------------------------------|--------------|------------------|
| <b>Virrankulutus (30 min)</b> | 39,4 mAh     | 131,4 mAh        |

## 6.4 Kirjasimet

Kirjasintiedostojen optimoinnin testausta varten tehtiin sivu, jossa on otsikko Roboto Bold -kirjasintyypillä ja viisi pitkää kappaletta ipsum lorem paikanpitäjättekstiä Roboto Regular -kirjasintyypillä. Molemmat kirjasintiedostot olivat TTF-formaatissa. Sitten poistettiin Roboto Bold- kirjasintiedostosta kaikki muut kirjaimet paitsi o, O, t, s, i ja k. Otsikon sisältö oli siis "Otsikko". Roboto Regular -fontista poistin muut paitsi Basic Latin -osajoukkoon kuuluvat merkit, eli jäljelle jäi yhteensä 95 merkkiä. Roboto Bold -fontitiedosto pieneni 164 kilotavusta 6,96 kilotavuun ja Roboto Regular -tiedosto pieneni 164 kilotavusta 15,2 kilotavuun.

officia. officia.

Kuva 13. Ligatuuri ei näy oikein optimoidussa kirjasintiedostossa (vasemmalla)

Lopuksi muunsin tiedostot vielä WOFF2-formaattiin, jolloin Roboto Regular -tiedosto (leipäteksti) pieneni 7,63 kilotavuun ja Roboto Bold -tiedosto (otsikko) 2,66 kilotavuun. Visuaalisesti lopputulos oli miltei sama, tosin kaksi peräkkäistä f-kirjainta, eli ligatuuri ei ollut ligatuurimuodossa optimoidulla sivulla, katso kuva 10. Lisäksi otsikon iso O-kirjain oli aavistuksen korkeampi optimoidulla sivulla.

Taulukko 2. Kirjasinten optimoinnin tulokset

| Kirjasimet                   | Ennen optimointia | Optimoitu |
|------------------------------|-------------------|-----------|
| Roboto Bold (otsikko)        | 164 kt            | 2,66 kt   |
| Roboto Regular (leipäteksti) | 164 kt            | 7,63 kt   |



## 6.5 Cascading Style Sheets (CSS)

Kun HTML5-Boilerplate mallineen main.css tiedosto minifioitiin Visual Studio Code-editorin avulla, saatiin jo valmiiksi vaatimaton 8,38 kilotavun tiedosto pienennettyä 3,54 kilotavuun. Lisäksi mallineeseen kuuluva normalize.css tiedosto saatiin pienenemään 5,99 kilotavusta 1,77 kilotavuun.

## 6.6 Kuvat

Testikuvat saatiin Rawzor-yrityksen testikuvasivulta (Rawzor 2015). Ne olivat häviöttömästi pakatussa PPM-formaatissa. Tästä 14:n kuvan paketista valittiin kymmenen kuvaa, joista tehtiin JPEG- ja AVIF-versiot. JPEG-kuvia käytettiin hitaassa ja raskaassa verkkosivutoteutuksessa ja AVIF-kuvia nopeassa ja kevyessä verkkosivussa.

Valmiitakin kuvakokoelmia, jotka on tarkoitettu kuvanpakkausalgoritmien testaamiseen tai tekoälyavusteiseen objektintunnistukseen, on olemassa, mutta ne eivät sisällä sekä vektori- että bittikarttakuvia. Tässä opinnäytetyössä on haluttu kokeilla kuinka vektorikuvat ja bittikarttakuvat vertautuvat tiedonsiirron määrän osalta, sekä myös kuinka vektorikuvasta muunnettu bittikarttakuva PNG-muodossa pärjää vertailussa.

JPEG-kuvat saatiin aikaiseksi avaamalla PPM-kuvat Irfanview-ohjelmistolla, asettamalla niiden leveydeksi 1000 pikseliä ja tallentamalla kuvat JPEG-formaattiin laatuasetuksella 100. JPEG-kuvien korkeus pikseleinä määräytyi automaattisesti kuvasuhteen mukaan. AVIF-kuvat tehtiin tallentamalla Irfanview-ohjelmistolla PPM-kuvat 1000 pikseliä leveiksi ja muuntamalla sen jälkeen kuvat convertio.co -verkkotyökalulla AVIF-muotoon. Kuvassa 14 on esimerkit eri muotoisista kuvista. Kuvien koot eri tiedostomuodoissa esitetään taulukossa 3.

Taulukko 3. Kuvien koot eri tiedostomuodoissa.

| <b>Kuvan nimi</b>  | <b>Alkuperäinen PPM-tiedosto</b> | <b>JPEG-tiedosto</b>  | <b>AVIF-tiedosto</b> |
|--------------------|----------------------------------|-----------------------|----------------------|
| big_tree           | 6088x4550px<br>158Mt             | 1000x747px<br>679kt   | 1000x747px<br>263kt  |
| bridge             | 2749x4049px<br>63,6Mt            | 1000x1473<br>1,23Mt   | 1000x1473<br>389kt   |
| cathedral          | 2000x3003px<br>34,4Mt            | 1000x1504px<br>1,13Mt | 1000x1504px<br>257kt |
| deer               | 4043x2641px<br>61Mt              | 1000x653px<br>429kt   | 1000x653px<br>48,9kt |
| fireworks          | 3136x2352px<br>42,2Mt            | 1000x750px<br>342kt   | 1000x750px<br>64kt   |
| flower_foveon      | 2268x1512px<br>19,6Mt            | 1000x667px<br>312kt   | 1000x667px<br>41,3kt |
| hdr                | 3072x2048px<br>36,0Mt            | 1000x667px<br>367kt   | 1000x667px<br>53,3kt |
| leaves_iso_1600    | 3008x2000px<br>34,4Mt            | 1000x665px<br>699kt   | 1000x665px<br>347kt  |
| nightshot_iso_1600 | 3136x2352px<br>42,2Mt            | 1000x750px<br>574kt   | 1000x750px<br>157kt  |
| spider_web         | 4256x2848px<br>69,3Mt            | 1000x669px<br>342kt   | 1000x669px<br>35,8kt |



Kuva 14. Vasemmallä ylhäällä alkuperäinen häviöttömästi pakattu PPM-kuva, oikealla ylhäällä ImageMagickin tuottama heikkolaatuinen AVIF-kuva, vasemmallä alhaalla convertio.co-työkalun tekemä AVIF-kuva, oikealla alhaalla JPEG-kuva laatuasetuksella 100

Myös ImageMagick-ohjelmaa kokeiltiin, mutta jostain syystä se heikensi kuvien laatua huomattavan paljon. ImageMagickilla muuntamisen jälkeen kuvissa oli huomattavia pakkausartefakteja, värit vaalenivat ja kontrasti heikentyi. ImageMagickin muuntamia kuvia ei siis käytetty, kuvassa 14 oikealla ylhäällä on kuitenkin näytetty esimerkki sen tuottamasta lopputuloksesta. Lisäksi kokeiltiin tallentaa PPM-kuvat PNG-muotoon Irfanview-ohjelmalla ja muuntaa ne AVIF-formaattiin avif.io-verkkopalvelussa, mutta avif.io:n tuottamat AVIF-kuvat eivät toimi ollenkaan selaimissa.

## 6.7 Äänitiedostojen luonti

Lähdetiedostona, josta luotiin MP3- ja Opus-muotoiset tiedostot, toimi Wikimedia Commonsista peräisin oleva FLAC-tiedosto, jonka koko nimi on ”Antonín Dvořák - Concerto for Violoncello and Orchestra in B minor, op. 104 - 3. Finale (Pierre Fournier, George Szell, Berliner Philharmoniker, 1962)”. Audacity-ohjelman avulla tehtiin tästä FLAC-tiedostosta MP3- ja OPUS-tiedostot käytettäväksi myöhemmissä mittauksissa.

FLAC (Free Lossless Audio Codec) on ilmainen vapaan lähdekoodin häviötön äänikoodekki. MP3-tiedostoa käytettiin raskaassa ja hitaassa verkkosivutoteutuksessa, Opus-tiedostoa taas nopeassa ja kevyessä verkkosivutoteutuksessa. On olennaista käyttää lähdetiedostona mahdollisimman hyvälaatuisia tiedostoa. Kun vaikkapa Opus-tiedosto muunnetaan MP3-tiedostoksi, pakkausartefaktit kasaantuvat ja lopputulos ei anna hyvää kuvaa kodekin suorituskyvystä, sillä molemmat näistä formaateista ovat häviöllisiä. Ei siis kannata muuntaa häviöllisestä formaatista toiseen häviölliseen formaattiin. Kuvassa 15 on esimerkki siitä, kuinka JPEG-tiedoston laatu huononee pakkausartefaktien kasaantumisen vuoksi. Kuvaa on ensin käännetty 90 astetta ja sitten tallennettu itsensä päälle 2000 kertaa.



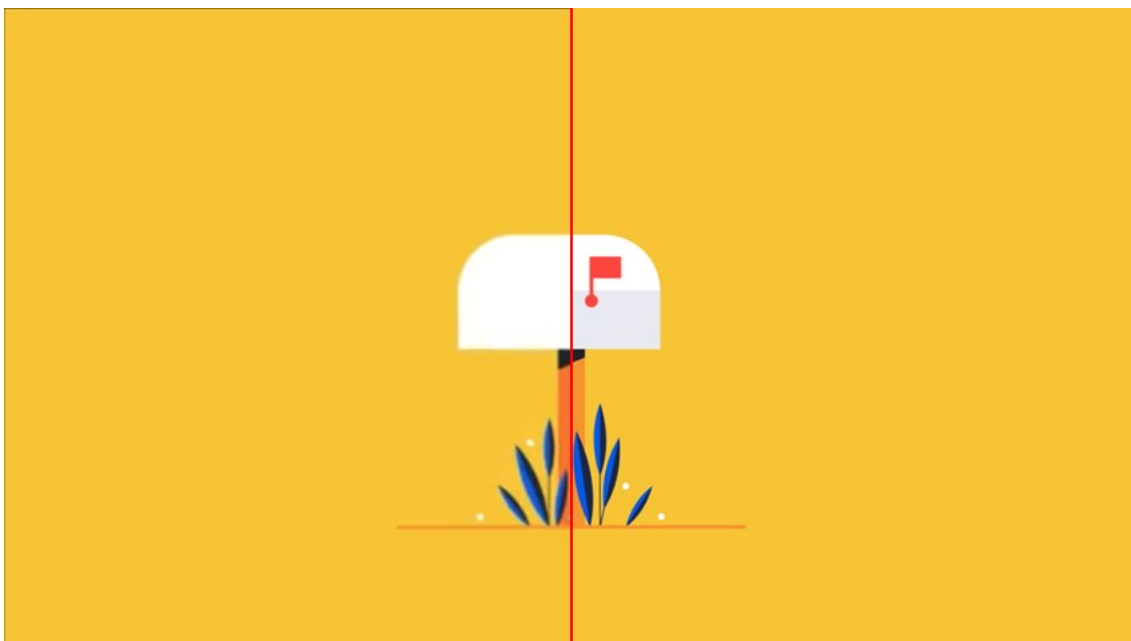
Kuva 15. Esimerkki pakkausartefaktien kasaantumisesta (Burny 2017)

Vertailu esimerkiksi Opus-muotoisen ja MP3-muotoisen äänitiedoston välillä on vaikeaa. Ei ole yksiselitteistä, millä bittinopeudella saavutetaan toista formaattia vastaava äänenlaatu kussakin muodossa. Tämän selvittäminen vaatisi koejärjestelyn, jossa erilaatuisia tiedostoja soitettaisiin suurelle joukolle ihmisiä.

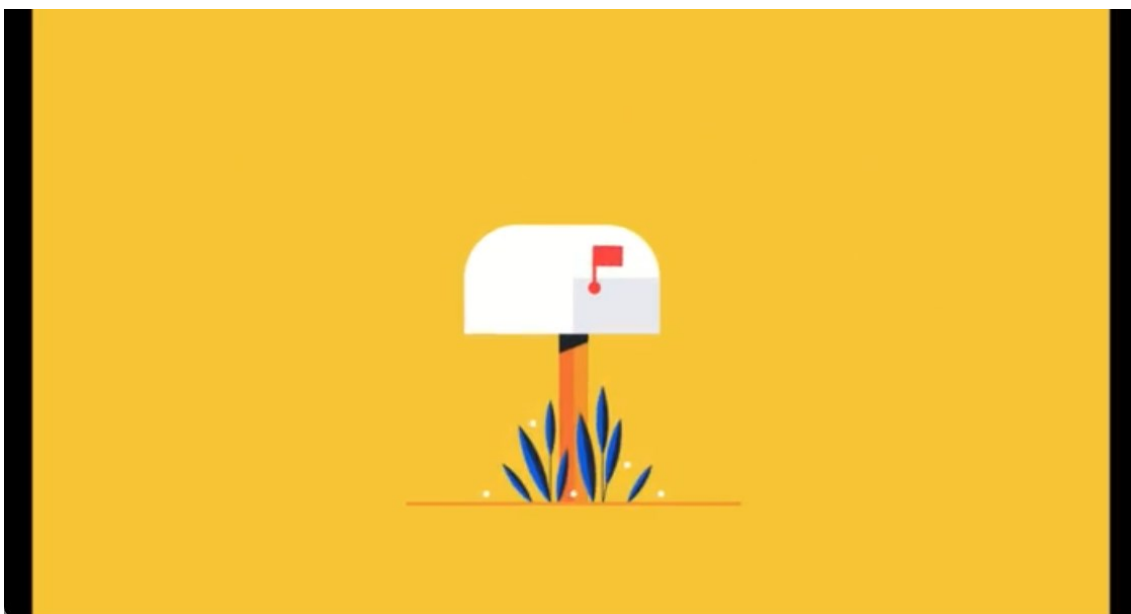
## 6.8 Animaation ja videon vertailu

Kuvassa 16 on vertailtu vierekkäin SVG-animaation ja AVC-videon laatua, vasemmalla puolella video, oikealla SVG-animaatio. Molempien natiiviresoluutio on 700 kertaa 400 pikseliä. MediaInfo-työkalun mukaan videon kuvataajuus on 53,12 kuvaa sekunnissa ja bittinopeus 115 kilobittiä sekunnissa (ks. liite 1). SVG-animaatio ei pikselöidy missään koossa, koska se koostuu vektoreista. Siksi SVG-animaatio näyttää huomattavasti terävämmältä tässä koko ruudun kuvankaappauksessa. Tiedostokoossa ei ole merkittävää eroa tässä 3,5 sekunnin videossa, SVG-animaatio on 45 kilotavua ja AVC-video 50 kilotavua.

Kuva 17 on kuvakaappaus videosta natiiviresoluutiossa 700 kertaa 400 pikseliä, kuva 18 on kuvakaappaus SVG-animaatiosta samalla resoluutiolla. Riippuen animaation sisällöstä, se kumpi tiedostomuoto vie vähemmän tilaa voi vaihdella. Jos animoidaan useita monimutkaisia objekteja, joilla on monimutkaisia liikeratoja, sitä suuremmaksi SVG-animaation tiedostokoko muodostuu.



Kuva 16. Vasemmalla video, oikealla SVG-animaatio



Kuva 17. Kuvakaappaus AVC-videosta natiiviresoluutiolla



Kuva 18. Kuvakaappaus SVG-animaatiosta samalla resoluutiolla kuin video (kuva 17)

## 6.9 Hitaan ja raskaan sivun mittaus

Raskas sivu sisältää 30 sekunnin mittaisen videon. Video on AVC-muodossa MP4-säilössä, resoluutiolla 1280x720 pikseliä, bittinopeudella 849kt/s. Videossa ei ole ääniraitaa.

Sivulla on myös minuutin mittainen musiikkinäyte. Musiikkinäyte on MPEG-1 Audio Layer III -muodossa MP3-säilössä näytteenottotaajuudella 44.1 kHz, stereoäänenä, keskimäärin 160 kt/s bittinopeudella.

Lisäksi sivulla on viisi maksimilaatuista JPEG-kuvaa, joiden leveys on 1000 pikseliä, neljä PNG-kuvaa, joiden leveys on 400 pikseliä ja yksi kolmen sekunnin GIF-animaatio. GIF-animaatio on 700x400 pikselin kokoinen. Sivulla on myös otsikko Roboto Bold -kirjasimella ja viisi kappaletta ipsum lorem -paikanpitäjättekstiä Roboto Regular -kirjasimella. Kirjasin on ttf-formaatissa. Sivun latautuu uudelleen JavaScriptin avulla, kun minuutin mittainen musiikkinäyte on soinnut loppuun. Kuvassa 18 on esitetty eri aineistotyyppien koot.



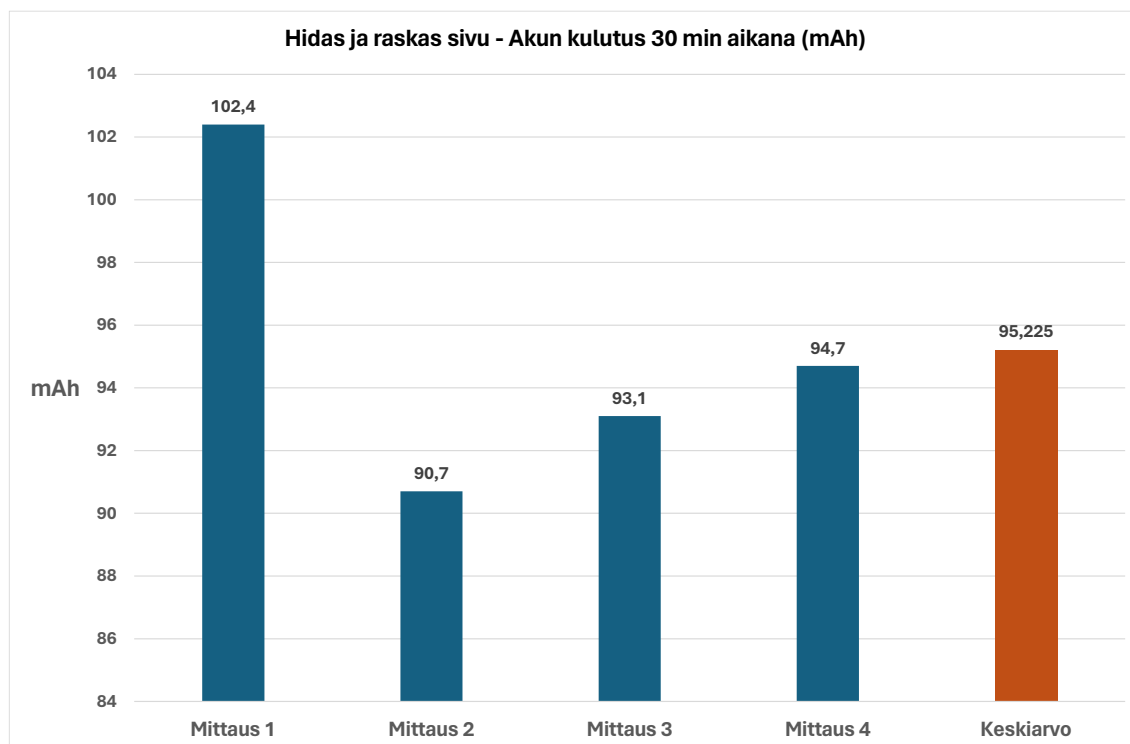
| Aineistotyyppien koot raskaassa ja hitaassa toteutuksessa |                   | Aineistotyyppien koot kevyessä ja nopeassa toteutuksessa |                   |
|---|-------------------|--|-------------------|
| Tyyppi  | Koko kilotavuissa | Tyyppi   | Koko kilotavuissa |
| CSS   | 14,2              | CSS  | 5,31              |
| Fontit  | 327               | Fontit   | 39,4              |
| HTML  | 14                | HTML   | 14                |
| Kuvat   | 2920              | Kuvat  | 656               |
| JavaScript  | 11,6              | JavaScript   | 11,6              |
| Ääni  | 1119              | Ääni   | 957               |
| Animaatio   | 423               | Animaatio  | 42,4              |
| Video   | 3182              | Video  | 2824              |
| Yhteensä  | 7,89 Mt           | Yhteensä   | 4,45 Mt           |

Kuva 19. Aineistotyyppien jakautuminen eri toteutuksissa. Hitaassa ja raskaassa toteutuksessa kuvat ovat yhtä suuria sekä työpöytä- että mobiiliversiossa. Kevyen ja nopean sivun työpöytäversiossa kuvien tiedostokoko on taulukon mukainen, mobiiliversiossa kuvien tiedostokoko on yhteensä vain 65 kilotavua

Puhelin valmisteltiin samoilla mittausjärjestelyillä kuin luvussa 6.1. Näytön kirkkaus ja median äänenvoimakkuus asetettiin puoleenväliin liittämällä puhelin USB-johdolla kannettavaan ja sitten Android debug bridgen avulla annettiin komennot: "adb shell cmd settings put system screen\_brightness 1023" ja "adb shell cmd media\_session volume --show --stream 3 --set 8". Kirkkauden maksimiarvo oli 2047 ja äänenvoimakkuuden 15.

Sivun annettiin latautua uudestaan 30 minuutin ajan, eli n. 30 kertaa. Selaimena toimi Firefox Android-käyttöjärjestelmälle. Selaimeksi valikoitui Firefox, koska sillä on mahdollista asettaa verkkosivun äänitiedosto toistumaan automaattisesti

ja koska uudelleenlataus JavaScriptin avulla oli riippuvainen äänitiedoston loppuun asti soimisesta. Androidin Chromessa ei ole mahdollista saada äänitiedostoa toistumaan automaattisesti vain asetuksia muuttamalla. AccuBattery-sovelluksen avulla mitattiin selaimen virrankulutus. Mittaus toistettiin neljä kertaa identtisin mittausjärjestelyin. Samanlaisista järjestelyistä huolimatta mittauksissa havaittiin yllättävän suurta vaihtelua, kts. kuva 20.

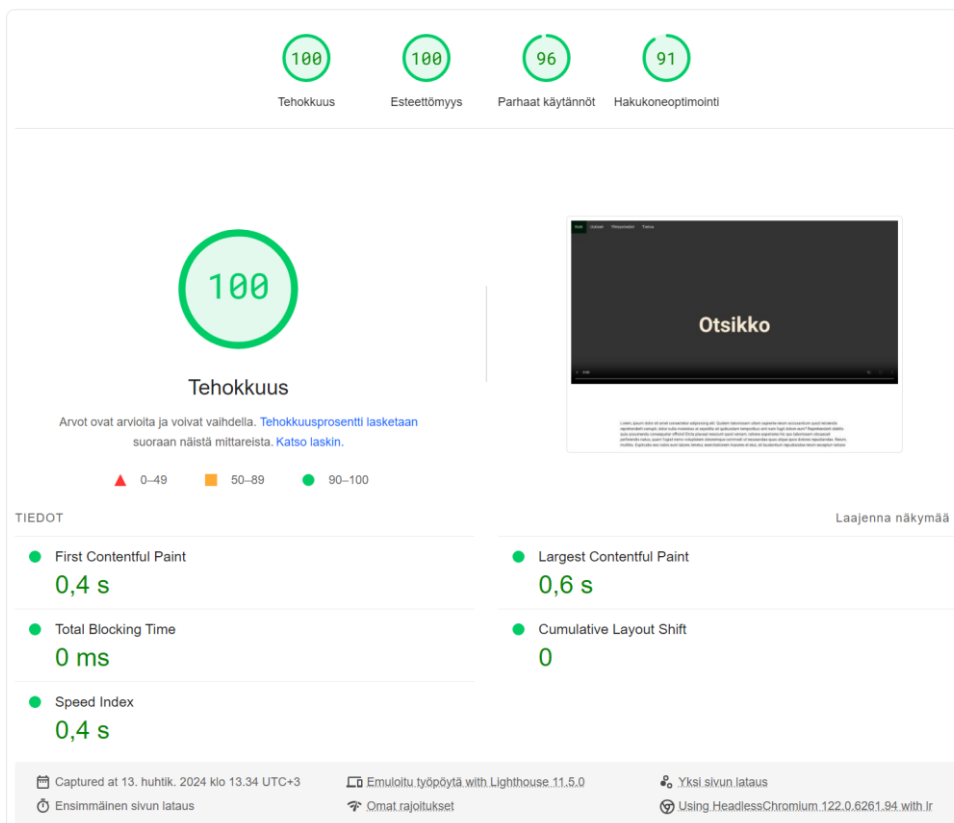


Kuva 20. Akun kulutus raskaan sivun mittauksen aikana AccuBattery-sovelluksen mukaan

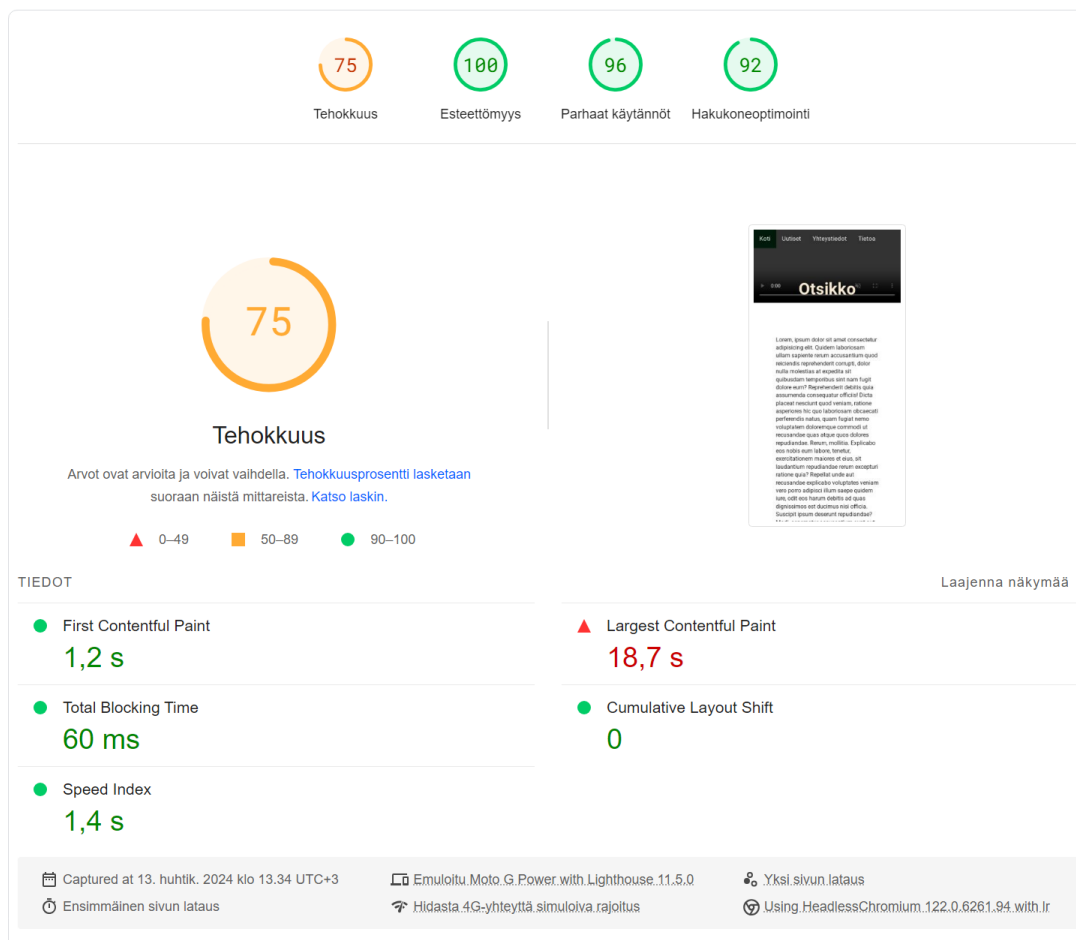
| Name  | Type                 | Size    |
|---|----------------------|---------|
| raskas/   | document             | 5.0 kB  |
| Roboto-Regular.ttf                                    | font                 | 169 kB  |
| Roboto-Bold.ttf                                       | font                 | 168 kB  |
| postilaatikko-shotcut.gif                             | gif                  | 435 kB  |
| big_tree.jpg  | jpeg                 | 697 kB  |
| cathedral.jpg   | jpeg                 | 1.2 MB  |
| deer.jpg  | jpeg                 | 440 kB  |
| flower_foveon.jpg                                     | jpeg                 | 320 kB  |
| spider_web.jpg  | jpeg                 | 352 kB  |
| site.webmanifest                                      | manifest             | 470 B   |
| H264-720-AVGBR849-23976fpspeak.mp4                    | media                | 3.3 MB  |
| Concerto_for_Violoncello_and_Orchestra_in_B_minor.mp3 | media                | 1.1 MB  |
| Paris_m_3m_jms.png                                    | png                  | 21.0 kB |
| Paris_m_corr_jms.png                                  | png                  | 17.7 kB |
| Convex_tangram_shapes_black_01.png                    | png                  | 12.0 kB |
| Flag_of_Europe_blue.png                               | png                  | 17.1 kB |
| icon.png  | png                  | 4.2 kB  |
| cropped-favico-32x32.png                              | png                  | 663 B   |
| modernizr-3.11.2.min.js                               | script               | 4.0 kB  |
| plugins.js  | script               | 676 B   |
| normalize.css   | stylesheet           | 2.2 kB  |
| main.css  | stylesheet           | 3.4 kB  |
| favicon.ico   | text/html / Redirect | 250 B   |

Kuva 21. Raskaan sivun osatekijät

Kuvassa 21 listattuna kaikki raskaan sivun osatekijät. Googlen PageSpeed Insights -työkalun mukaan raskaan sivun työpöytäversio suoriutui hyvin suorituskyvyn kannalta (kuva 22), mutta mobiiliversio toimi hitaasti (kuva 23). Kuvassa 24 on parannusehdotuksia mobiiliversioon.



Kuva 22. Raskaan sivun työpöytäversion tehokkuus PageSpeed Insights -työkalun mukaan



Kuva 23. Raskaan sivun mobiiliversion tehokkuus PageSpeed Insights -työkalun mukaan

| DIAGNOSTIIKKA                       |  |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | Kuvaelementeiltä puuttuu kiinteä <code>width</code> ja <code>height</code>     |
| <input checked="" type="checkbox"/> | Jakele kuvat seuraavan sukupolven muodoissa — Potentiaalinen säästö: 2 513 KiB |
| <input checked="" type="checkbox"/> | Jakele animaatiosisältöä videomuodossa — Potentiaalinen säästö: 267 KiB        |
| <input checked="" type="checkbox"/> | Varmista, että teksti pysyy näkyvässä verkkofontin lataamisen aikana           |
| <input checked="" type="checkbox"/> | Määritä kuvien koko oikein — Potentiaalinen säästö: 371 KiB                    |
| <input checked="" type="checkbox"/> | Koodaa kuvat tehokkaasti — Potentiaalinen säästö: 1 959 KiB                    |
| <input checked="" type="checkbox"/> | Vältä valtavia verkkoresursseja — Yhteenlaskettu koko oli 6 944 KiB            |
| <input type="checkbox"/>            | Avoid large layout shifts — 2 layout shifts found                              |
| <input type="checkbox"/>            | Palvelimen vasteaika alussa oli lyhyt — Päädokumentti käytti 50 ms             |
| <input type="checkbox"/>            | Vältä liian suurta DOM:ää — 46 elementtiä                                      |
| <input type="checkbox"/>            | Vältä kriittisten pyyntöjen ketjuttamista — 5 ketjua löydetty                  |
| <input type="checkbox"/>            | JavaScriptin suorittamiseen kuluva aika — 0,0 s                                |
| <input type="checkbox"/>            | Minimoi pääsäikeen työkuorman — 0,2 s  |
| <input type="checkbox"/>            | Suurin renderöity sisältöosa — 610 ms  |

Lisätietoja sovelluksen toiminnasta. Luvut eivät suoraan vaikuta tehokkuusprosenttiin.

Kuva 24. PageSpeed Insights -työkalun parannusehdotukset mobiiliversiolle

Raskas sivu latautuu hitaasti mediatiedostojen suuren tiedostokoon vuoksi, eikä siksi, että se vaatisi monimutkaista prosessointia tai sen rakenne olisi liian monimutkainen.

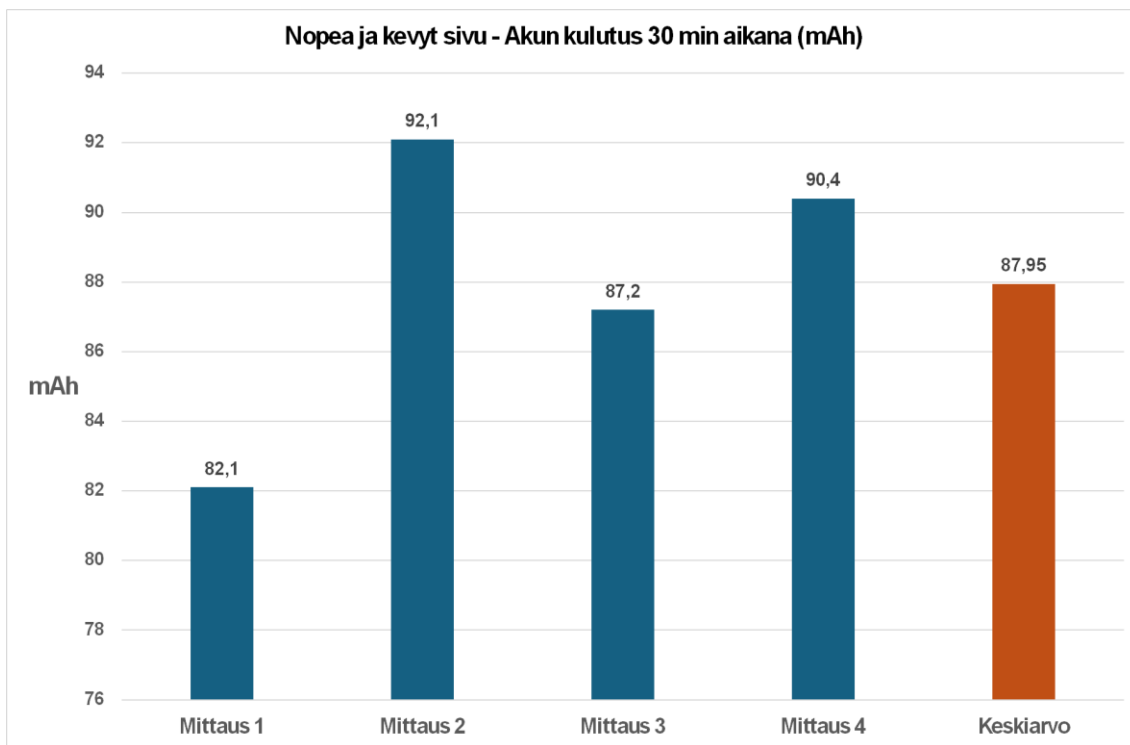
## 6.10 Nopean ja kevyen sivun mittaus

Kevyt sivu sisältää saman 30 sekunnin videon kuin raskas sivu, mutta video on pakattu AV1-koodekilla. Video on MP4-säilössä, resoluutiolla 1280x720 pikseliä, bittinopeudella 768kt/s. Huolimatta pienemmästä bittinopeudesta, video on silmämääräisesti katsoen jopa laadukkaampi kuin raskaan sivun AVC-muotoinen video.

Musiikkinäyte on OPUS-muodossa, näytteenottotaajuudella 48.0 kHz, stereoäänellä, 130 kt/s bittinopeudella.

Srcset-määrittäystä hyödyntäen selaimelle tarjotaan mobiilissa 400 pikseliä leveitä AVIF-kuvia, pienin kuva on 4,19 kilotavua ja suurin 16,1 kilotavua.

Kirjasimena toimii Open Sans variable-kirjasin. Slice-työkalulla rajataan kirjaimen leveysakseli välille 300–700. Lisäksi kirjasintiedostosta poistetaan muut kirjaimet paitsi Latin-osajoukko.



Kuva 25. Akun kulutus nopean ja kevyen sivun mittauksen aikana AccuBattery-sovelluksen mukaan

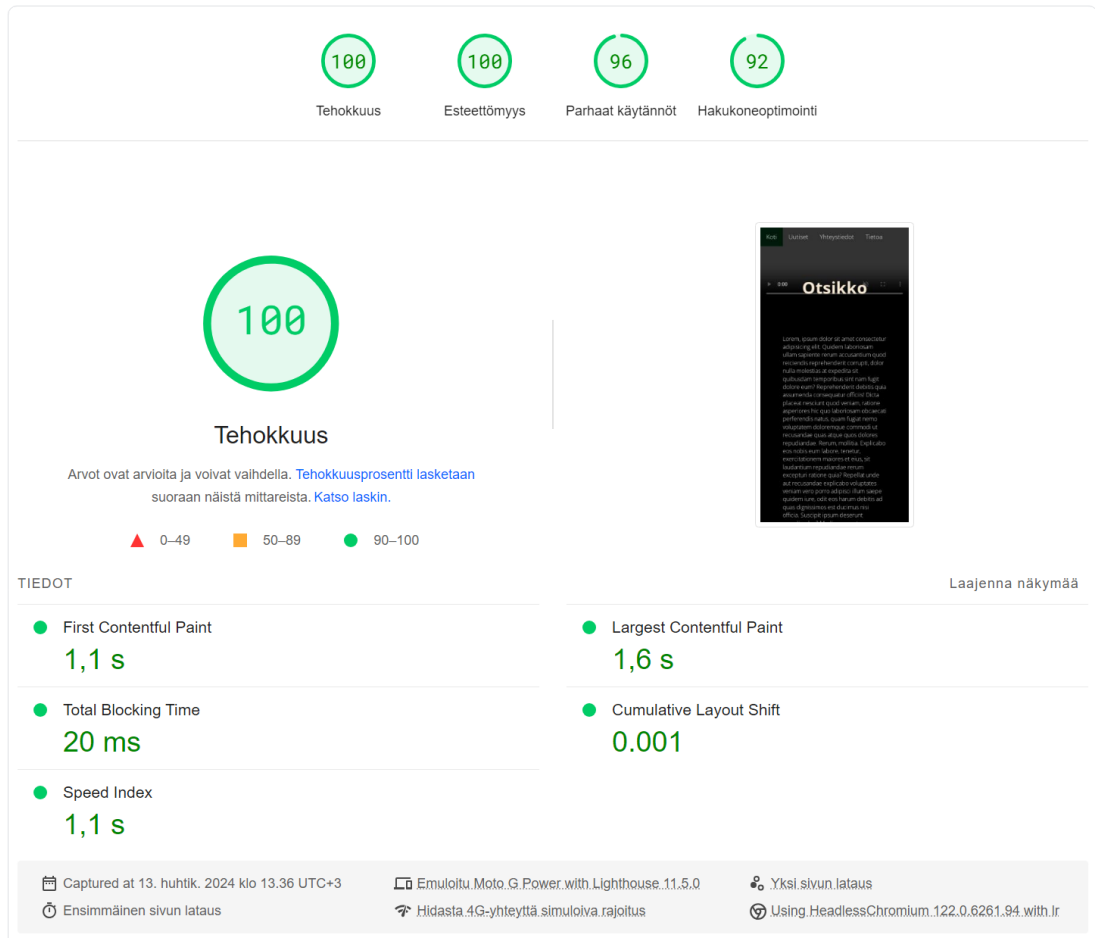
| Name  | Type                 | Size    |
|---|----------------------|---------|
| kevyt3/   | document             | 5.2 kB  |
| normalize.css   | stylesheet           | 2.2 kB  |
| main.css  | stylesheet           | 3.5 kB  |
| modernizr-3.11.2.min.js                                   | script               | 4.0 kB  |
| AV1-720-AVGBR849-23976fps.mp4                             | media                | 209 B   |
| plugins.js  | script               | 682 B   |
| big_tree400.avif  | octet-stream         | 18.6 kB |
| cathedral400.avif   | octet-stream         | 16.8 kB |
| deer400.avif  | octet-stream         | 5.0 kB  |
| Concerto_for_Violoncello_and_Orchestra_in_B_minor.opus    | media                | 16.4 kB |
| OpenSans-VariableFont_wdth,wght_EDIT_sliced2-subset.woff2 | font                 | 40.7 kB |
| flower_foveon400.avif                                     | octet-stream         | 5.3 kB  |
| spider_web400.avif  | octet-stream         | 4.5 kB  |
| Concerto_for_Violoncello_and_Orchestra_in_B_minor.opus    | media                | 95.6 kB |
| Paris_m_3m_jms.svg  | svg+xml              | 780 B   |
| Paris_m_corr_jms.svg                                      | svg+xml              | 815 B   |
| Convex_tangram_shapes_black_01.svg                        | svg+xml              | 3.0 kB  |
| Concerto_for_Violoncello_and_Orchestra_in_B_minor.opus    | media                | 771 kB  |
| Flag_of_Europe_blue.svg                                   | svg+xml              | 1.7 kB  |
| message-delivered-to-mailbox-animation.svg                | svg+xml              | 7.5 kB  |
| favicon.ico   | text/html / Redirect | 249 B   |
| site.webmanifest  | manifest             | 477 B   |
| favicon.ico   | text/html / Redirect | 249 B   |
| cropped-favico-32x32.png                                  | png                  | 662 B   |
| cropped-favico-32x32.png                                  | png                  | 662 B   |

Kuva 26. Kevyen sivun osatekijät mobiiliversiossa

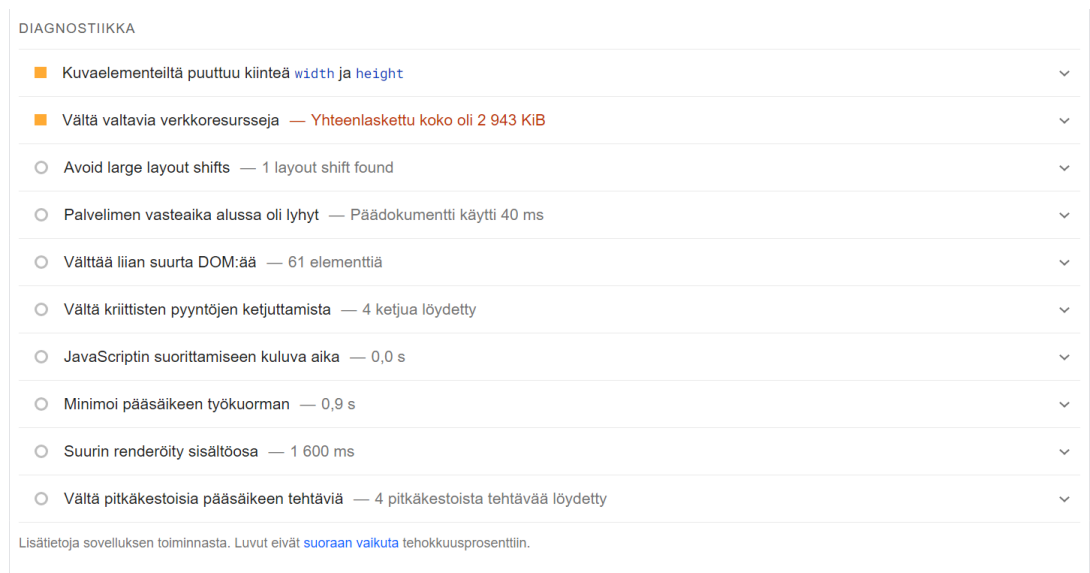
Neljän mittauskerran virrankulutuksen keskiarvoksi saatiin 87,95 milliampeerituntia, kts. kuva 25. Tulos oli hieman yllättävä, sillä osatekijöiden tiedostokoot olivat niin paljon pienempiä (kts. kuva 26), että tältä kevyeltä sivulta odotettiin parempaa energiatehokkuutta. Tiedonpakkauksen tapauksessa on kuitenkin niin, että pakkauksen purkaminen vaatii aina prosessointitehoa. Ilmeisesti AVIF-kuvat, AV1-video ja OPUS-äänitiedostot vaativat suhteellisen paljon prosessointia.

PageSpeed Insights -työkalun tehokkuusarvio kevyelle sivulle kuvassa 27 ja parannusehdotukset kuvassa 28.





Kuva 27. Kevyen sivun mobiiliversion tehokkuus PageSpeed Insights -työkalun mukaan



Kuva 28. PageSpeed Insights -työkalun parannusehdotukset kevyen sivun mobiiliversiolle

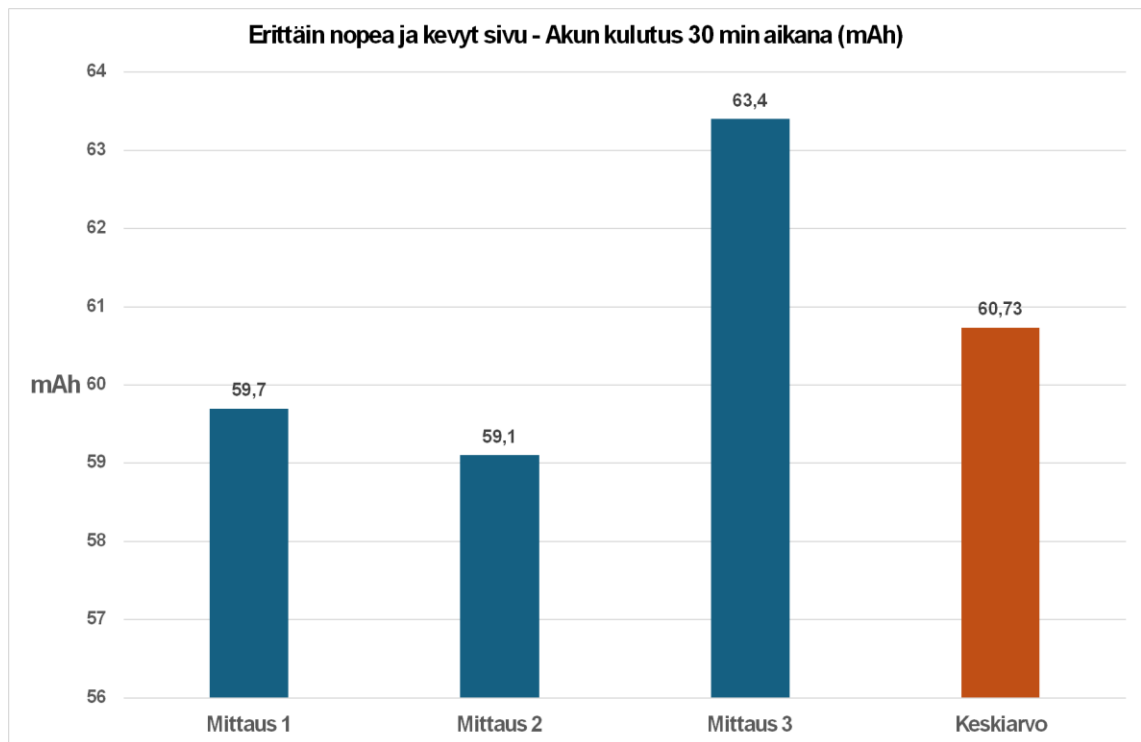
## 6.11 Erittäin nopean ja kevyen sivun mittaus

Lopuksi mitattiin vielä erittäin nopean ja kevyen sivun toimintaa. Sivu on muuten sama kuin nopea ja kevyt sivu, mutta siitä poistettiin video. Kuvat, äänitiedosto ja animaatio korvattiin linkeillä kuhunkin tiedostoon. Jäljelle jäivät otsikko, tekstikappaleet ja SVG-muotoiset kuvat. HTML-koodia yksinkertaistettiin poistamalla aivan kaikki, mikä ei ollut välttämätöntä sivun toiminnan kannalta.

Sivun uudelleenlataus toteutettiin eri tavalla kuin hitaassa ja nopeassa toteutuksessa. HTML:n head-elementtiin lisättiin selaimelle pyyntö uudelleen ladata sivua 60 sekunnin välein. Tämä siksi että äänitiedostoa ei enää ollut, jolloin JavaScriptiin ja äänitiedoston loppumiseen perustuva lähestymistapa ei enää olisi toiminut.

HTML:n oikeellisuus tarkastettiin W3C Markup Validation -palvelulla, se läpäisi tarkistuksen täysin. Lisäksi HTML5-Boilerplaten mukana tulleet normalize.css ja Modernizr-JavaScript-kirjasto otettiin pois. Erillisiä kirjasintiedostoja ei käytetty ollenkaan vaan fontiksi asetettiin Arial, joka löytyy selaimesta suoraan.

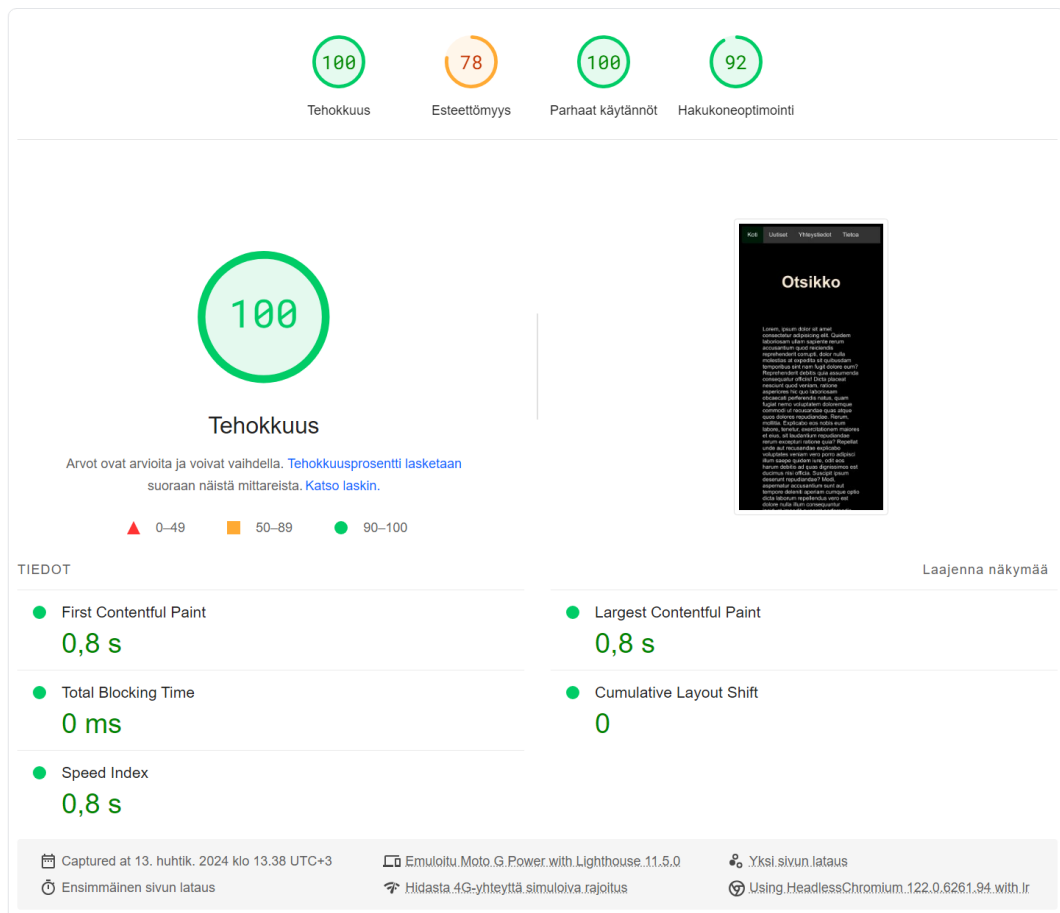
AccuBatteryn mukaan virtaa kului keskimäärin n. 61 mAh/30 min, kuvassa 29 tarkemmat mittaustulokset.



Kuva 29. Erittäin nopean ja kevyen sivun virrankulutus mittausten mukaan

| Name                               | Type                 | Size   |
|------------------------------------|----------------------|--------|
| erinopee/                          | document             | 4.4 kB |
| main.css                           | stylesheet           | 823 B  |
| Paris_m_3m_jms.svg                 | svg+xml              | 780 B  |
| Paris_m_corr_jms.svg               | svg+xml              | 815 B  |
| Convex_tangram_shapes_black_01.svg | svg+xml              | 3.0 kB |
| Flag_of_Europe_blue.svg            | svg+xml              | 1.7 kB |
| favicon.ico                        | text/html / Redirect | 249 B  |
| cropped-favico-32x32.png           | png                  | 662 B  |

Kuva 30. Erittäin nopean ja kevyen sivun osatekijät



Kuva 31. Erittäin nopean ja kevyen mobiilisivun tehokkuus PageSpeed Insights -työkalun mukaan

| DIAGNOSTIIKKA                         |  |
|---------------------------------------|--|
| <span style="color: orange;">■</span> | Kuvaelementeiltä puuttuu kiinteä <code>width</code> ja <code>height</code> |
| <input type="radio"/>                 | Palvelimen vasteaika alussa oli lyhyt — Päädokumentti käytti 40 ms         |
| <input type="radio"/>                 | Välttää valtavia verkkoresursseja — Yhteenlaskettu koko oli 12 KiB         |
| <input type="radio"/>                 | Välttää liian suurta DOM:ää — 38 elementtiä                                |
| <input type="radio"/>                 | Vältä kriittisten pyyntöjen ketjuttamista — 1 ketju löydetty               |
| <input type="radio"/>                 | JavaScriptin suorittamiseen kuluva aika — 0,0 s                            |
| <input type="radio"/>                 | Minimoi pääsäikeen työkuorman — 0,2 s                                      |
| <input type="radio"/>                 | Suurin renderöity sisältöosa — 830 ms                                      |

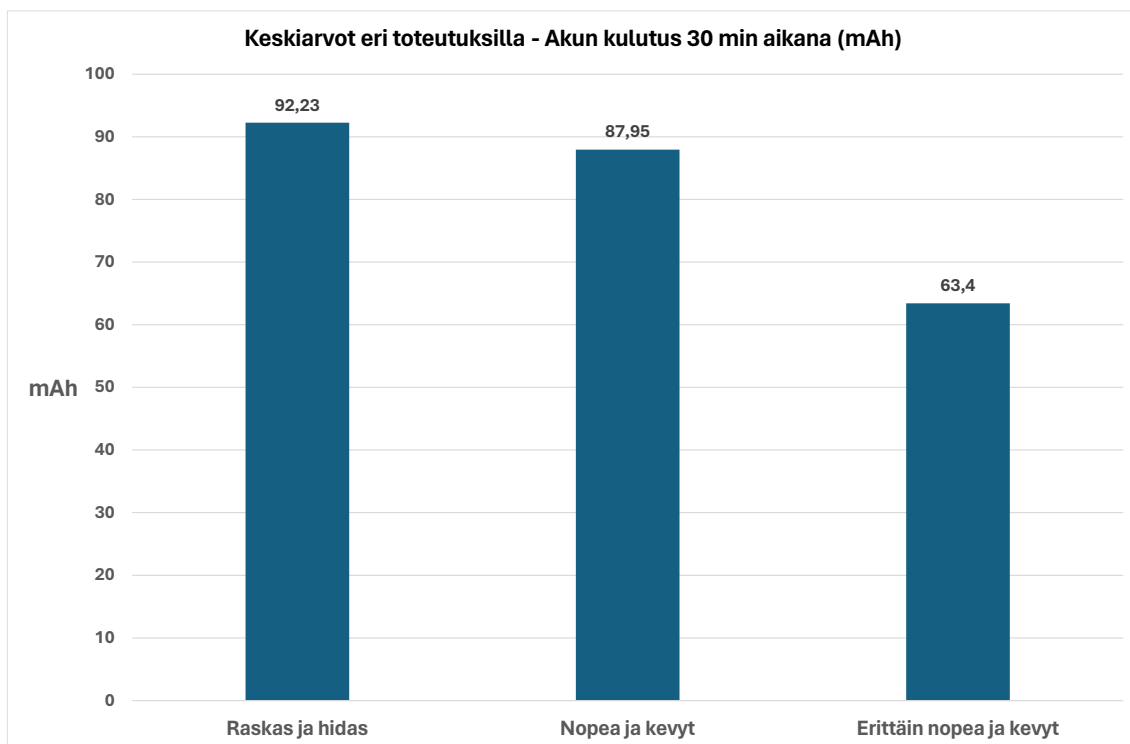
Lisätietoja sovelluksen toiminnasta. Luvut eivät [suoraan vaikuta](#) tehokkuusprosenttiin.

Kuva 32. Erittäin nopean ja kevyen sivun PageSpeed Insights -työkalun antamat parannusehdotukset

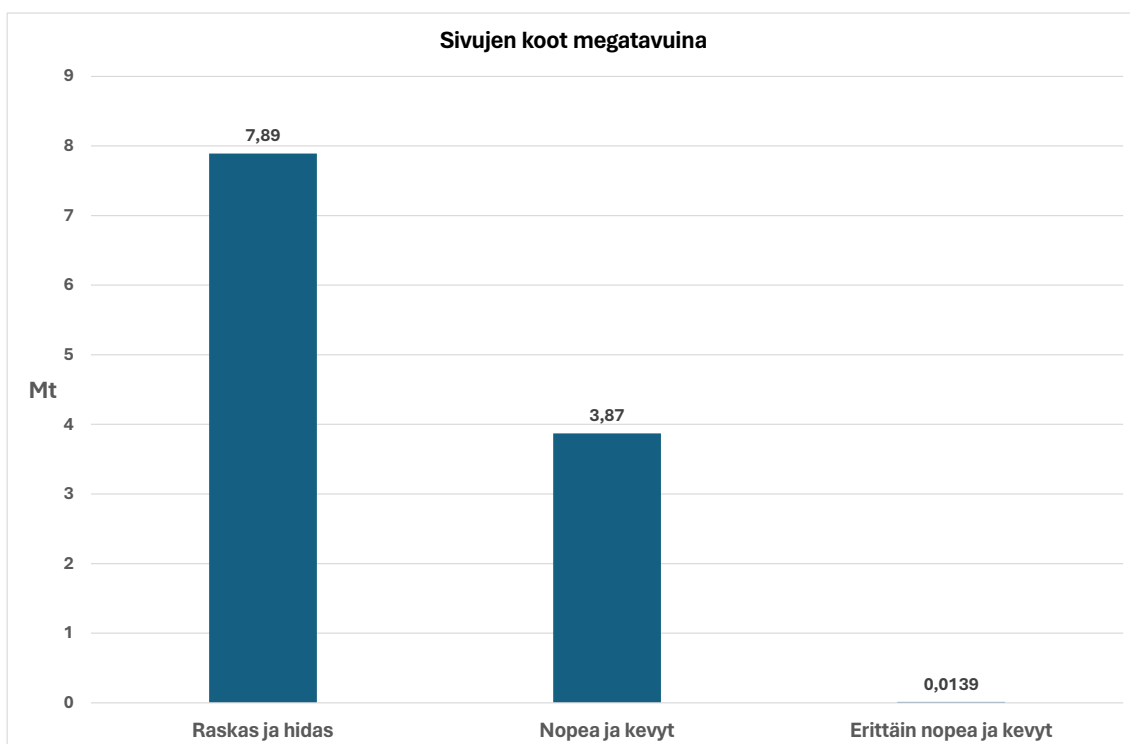
## 7 Testitulosten analysointi

Videon ja äänitiedoston toistaminen sujuu mittauksissa käytetyltä puhelimelta melko pienellä virrankulutuksella, vaikka niitä ei ladattu välimuistista vaan aina itse palvelimelta. Vaikka AVIF-kuvat, OPUS-ääni ja AV1-video eivät tuokaan suurta säästöä puhelimen virrankulutukseen, niiden tuomat datansiirron säästöt ovat niin suuria, että niitä kannattaa silti mahdollisuuksien mukaan käyttää. Suuri osa käyttäjistä käyttää mobiilidataa, joka kuluttaa paljon energiaa.

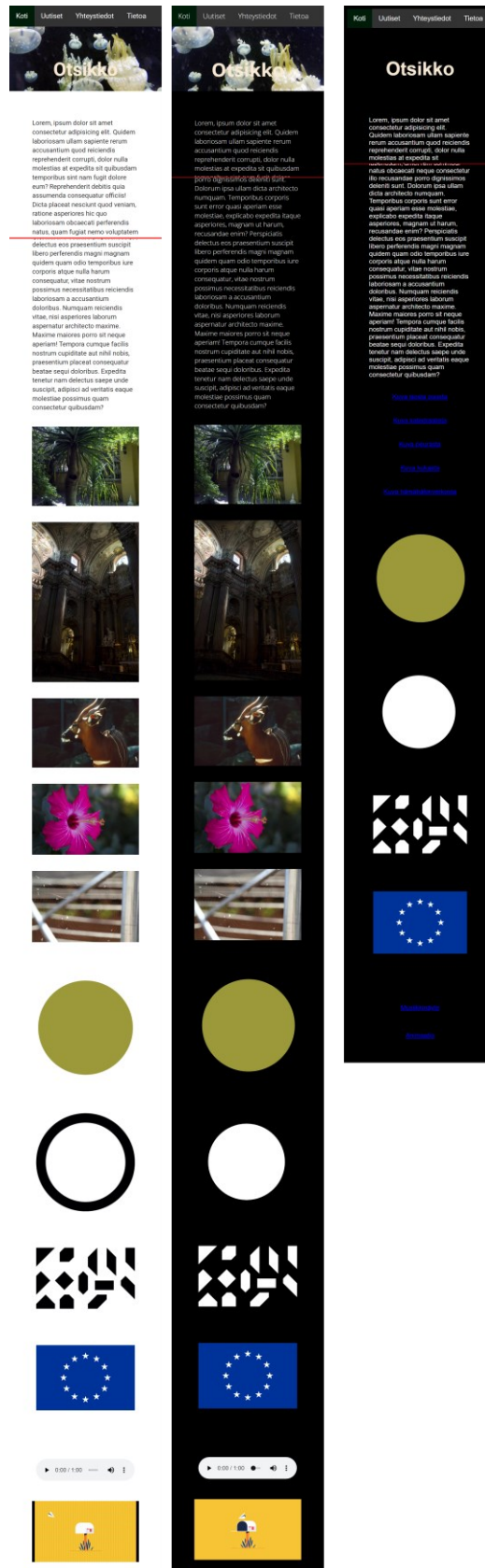
Poistamalla kuvat, animaatiot ja videot, kuten erittäin nopeassa ja kevyessä toteutuksessa on tehty, voidaan kuitenkin saavuttaa miltei yhden kolmasosan pudotus sähkönkulutuksessa verrattuna hitaaseen ja raskaaseen sivuun. Toisaalta tämä säästö korreloi vain vähäisessä määrin sivun koon kanssa, jonka voi todeta vertailemalla kuvia 33 ja 34. Erittäin nopea ja kevyt sivu on äärimmäisen nopea ja sen käyttäjäkokemus ei ole merkittävästi huonompi. Se ei kuitenkaan ole ulkoasultaan samanlainen kuin muut testatut sivut. Mediatiedostot täytyy avata erikseen klikkaamalla, mutta tämä saattaa olla saavutettavuuden kannalta jopa hyvä asia.



Kuva 33. Keskiarvot eri toteutusten virrankulutuksesta milliampeeritunteina



Kuva 34. Sivujen koot megatavuina



Kuva 35. Vertailu mobiiliversioiden ulkoasujen välillä, vasemmalla hidas ja raskas, keskellä nopea ja kevyt, oikealla erittäin nopea ja kevyt. Kuvat katkaistu tekstikappaleista punaisen viivan kohdalta.

## 8 Johtopäätökset - mitä suunnittelija voi tehdä?

Verkkosivujen suunnittelussa on osoittautunut, että hyvä suorituskyky ja alhainen tehonkulutus ovat yhteydessä toisiinsa. On suositeltavaa, että tämä otetaan huomioon verkkosivujen kehityksessä. Hyvät suorituskykypisteet selainten suorituskykytyökalujen (esim. Chromen DevTools) pisteyttäminä korreloivat negatiivisesti energiankulutuksen kanssa, ts. mitä suorituskykyisempi sivusto, sitä vähemmän se kuluttaa energiaa (Chan-Jong-Chu, Islam, Exposito, Sheombar, Valladares, Philippot, Grua & Malavolta 2020). Suunnittelijat ja kehittäjät voivat käyttää esimerkiksi Chrome-selaimen Lighthouse-työkalua korkeaa suorituskykyä ja alhaista tehonkulutusta tavoitellessaan. Jos verkkosivusto saa hyvät suorituskykypisteet Lighthouse-mittaustyökalun arvioimana, on se todennäköisesti myös energiatehokas.

Suunnittelijan tulisi ottaa huomioon eri päätelaitteiden ominaisuudet siten, että sivustot toimivat myös vanhemmilla laitteilla. Hyvän informaatioarkkitehtuurin ja mahdollisimman lyhyiden käyttäjäpolkujen avulla siirrettävät datamäärät vähenevät. Datansiirron optimoinnin avulla myös käyttäjäkokemus paranee, sillä mitä nopeammin sivut latautuvat, sitä tyytyväisempi käyttäjä on.

Tässä opinnäytetyössä tehdyt mittaukset osoittavat, että verkkosivun tiedonsiirron optimointi ja energiatehokkuus riippuu monista asioista: kuvakoot ja -formaatit, sivulla käytetyt värit, kirjasinten optimointi ja CSS-tiedostojen koko vaikuttavat kaikki siirrettävän datan määrään ja energiatehokkuuteen. Lisäksi energiatehokkuuteen vaikuttavat monet tekijät, jotka eivät liity verkkosivun sisällön ja ulkoasun toteutukseen. Vaikuttavia tekijöitä ovat mm. palvelimen sijainti, käytetty tiedonsiirtotekniikka (valokuitu, radioverkko), käytetyt ohjelmointikielet, ohjelmakirjastot ja -kehukset sekä päätelaitteen tekniset ominaisuudet.

Koska kuvat ovat JavaScriptin ohella merkittävin verkkosivun kokoon vaikuttava tekijä, suunnittelijan kannattaa kiinnittää niiden käyttöön ja optimointiin erityistä huomiota. Ensinnäkin tulee miettiä, tarvitaanko kuvaa ollenkaan, ja jos tarvitaan, missä formaatissa ja koossa sen pitäisi olla. Yli 2560 pikseliä leveää kuvaa ei



tarvita käytännössä juuri koskaan. Periaatteena tulisi olla, että jokaisen kuvan koko asetetaan sen mukaan minkä kokoinen näyttö käyttäjän päätelaitteessa on. Yksinkertaiset logot ym. geometriset muodot kannattaa esittää SVG-formaatissa, valokuvat ja muu monimutkaisempi grafiikka taas AVIF-formaatissa. Jos on tarpeen tukea vanhoja selaimia, AVIF-muotoisten kuvien kanssa tulee käyttää varavarmistusta, jossa vanhoille selaimille lähetetään JPEG-muotoinen kuva.

Myös värit vaikuttavat paljon päätelaitteen energiankulutukseen, jos se on varustettu uudenaikaisella OLED- tai AMOLED-näytöllä. Tällöin on suositeltavaa käyttää suunnitellessa tummia värejä ja välttää sinistä väriä. Tummista väreistä ei ole silloinkaan haittaa, kun päätelaitteessa on vanhempi LCD-tekniikkaa hyödyntävä näyttö.

Kirjasimet ovat tarvittavan datansiirtokapasiteetin kannalta kolmanneksi suurin aineistotyyppi verkkosivuilla. Kirjasinten suhteen tärkeintä on, että ei käytetä liikaa eri kirjasintyyppisiä ja suositaan mahdollisuuksien mukaan yhtä variable-kirjasinta, jota käytetään eri lihavoinneilla ja leveyksillä koko sivustolla.

Kirjasinten optimoinnin voi viedä hyvinkin pitkälle eri työkalujen ja tekniikoiden avulla, mutta kannattaa miettiä onko se vaivan arvoista, mikäli on muitakin optimointikohteita, jotka kaipaavat huomiota. Google Fonts -palvelun avulla voidaan optimoida sekä kirjasinten osajoukkoja (mm. latinalaiset tai kreikkalaiset aakkoset) että yksittäisten kirjainten lataamista esim. otsikoita varten. Aina Google Fonts ei kuitenkaan ole vaihtoehto. Esimerkiksi saksalainen tuomioistuim on tullut kinnut, että Google Fonts vuotaa käyttäjän IP-osoitteen Googlelle, joka on GDPR-asetuksen vastaista, joten saksalaisilla verkkosivuilla ei voi käyttää Google Fonts -palvelua (Claburn 2022).

Aineistotyypeistä neljänneksi eniten levytilaa vaatii CSS, jonka tiedostokoko kannattaa minifioida koontityökalun (build tool) avulla. Palvelimella tulisi olla lisäksi käytössä Gzip- tai Brotli-pakkaus, sillä ne pakkaavat CSS-tiedostot hyvin pieniksi, jolloin siirrettävän datan määrä vähenee.

Kaiken kaikkiaan energiatehokkuuteen vaikuttaa moni tekijä, joista monet eivät yleensä ole suunnittelijan päätettävissä. Suunnittelija voi kuitenkin tehdä paljon hyvän käyttökokemuksen, tehokkaan informaatioarkkitehtuurin ja selkeän käyttäjapolun avulla. Optimoimalla aineistojen tiedostokoot voidaan vähentää tarvittavaa tiedonsiirtokapasiteettia, jolloin säästyy myös energiaa.

## 9 Yhteenveto

Tässä opinnäytetyössä pyritään etsimään ratkaisuja verkkosivujen käytön aiheuttaman energiankulutuksen vähentämiseen ja turvaamaan vanhojenkin laitteiden nopea ja sujuva verkkopalveluiden käyttö.

Opinnäytetyössä pyrittiin löytämään eroja hitaiden ja nopeiden sivujen energiankulutuksen välillä. Päätelaitteiden osalta erot eri toteutusten välillä olivat erittäin pieniä, keskimäärin muutamia milliampeeritunteja. Tämä saattaa johtua osittain siitä, että mitattavat toteutukset olivat käsin koodattuja staattisia HTML-tiedostoja eikä niitä luotu dynaamisesti tietokannasta tietoa hakien. Myös JavaScript-tiedostoja ladattiin merkittävästi vähemmän kuin kaupallisilla verkkosivuilla. Esimerkiksi analytiikkaa ja mainosverkostoja, jotka vaativat JavaScriptiä toimiakseen, ei käytetty ollenkaan. Tiedonsiirron optimoinnissa onnistuttiin saavuttamaan hyviä tuloksia ja merkittäviä eroja tiedostokoossa. Nopea ja kevyt sivu oli noin puolet pienempi kuin hidas ja raskas sivu visuaalisen laadun pysyessä lähes samana.

Työskentelyprosessin aikana havaittiin, että hyvien työkalujen löytäminen oli yllättävän hankalaa. Sähkönkulutuksen mittaaminen milliampeereinatunteina vaati useamman työkalun kokeilua. Opinnäytetyön piti alun perin perustua Android Battery Historian -työkaluun, mutta sen antama data oli niin vaikeaa tulkita, että ajatuksesta luovuttiin. Tämän jälkeen kokeiltiin useampaa Android-sovellusta, kunnes löytyi AccuBattery Pro, joka näytti suoraan sähkönkulutuksen milliampeeritunteina.

Yleisesti voidaan sanoa, että useimmat työkalut ovat joko liian yksinkertaisia tai ne näyttävät liikaa dataa liian vaikeasti tulkittavassa muodossa. Lisäksi yllätys oli,

että AVIF-kuvien käsittelyyn tarkoitetut työkalut toimivat huonosti ilmeisesti ohjelmointivirheiden takia, ImageMagick väärästi värejä ja avif.io tuotti viallisia, toimimattomia tiedostoja. Opinnäytetyön tekijä on myös hyvin kyllästynyt Antonín Dvořákin opinnäytteessä käytetyn sävellyksen alkutahteihin (kappaleen ensimmäinen minuutti soi mittauksen aikana 30 kertaa per mittausta).

Opinnäytetyön antamat tulokset ovat sähkönkulutuksen osalta vain suuntaa antavia, mutta tiedonsiirron osalta saatiin selvä tulos: uudet kuva- ja videoformaatit mahdollistavat merkittävästi pienemmän tiedostokoon ja vähentävät siten tiedonsiirron tarvetta. Sähkönkulutuksen tarkempia mittauksia varten tarvittaisiin useampi erityyppinen laite ja enemmän toistoja eri tavoilla toteutettuja sisältöjä ladatain.

Jatkotutkimusmahdollisuuksia on useita. Samoja mittauksia voisi tehdä useilla eri laitetyyppillä. JavaScript-kehikkojen ja kirjastojen (esim. React) vaikutusta tiedonsiirron määrään ja energiankulutukseen voisi kartoittaa. Lisäksi voisi selvittää eri tiedostomuotojen vaikutusta palvelimen sähkönkulutukseen ja prosessorikuormaan sekä eri ohjelmointikielten vaikutus palvelimen kuormitukseen. Myös data-siirtotien vaikutusta sähkönkulutukseen on mahdollista tutkia.

Olen toiveikas sen suhteen, että pyrkimys ekologisesti kestäväan kehitykseen tulee vaikuttamaan myös verkkosivujen kokoon ja palvelutoteutusten energiatehokkuuteen siten, että tiedostokoot pienenevät ja energiatehokkuus paranee. Tällä hetkellä trendi on päinvastainen, mutta kun tietoisuus digitalisaation aiheuttamasta energiankulutuksesta lisääntyy suunnittelijoiden ja kehittäjien keskuudessa, tilanne toivottavasti muuttuu. Energiatehokkuuden varmistamisen ja tiedostokokojen minimoimisen pitäisi tulla olennaiseksi osaksi palvelujen kehitysprosessia. Työkalut ovat jo nyt olemassa, niitä täytyy vain käyttää.

## Lähteet

AccuBattery, 2020. Tab 2: discharging screen. Verkkosivu. <https://accubattery.zendesk.com/hc/en-us/articles/210226125-Tab-2-discharging-screen> (viitattu 26.3.2024).

Ahmadova, Gozal & Delgado-Márquez, Blanca L. & Pedauga, Luis E. & Leyva-de la Hiz, Dante I. 2020. Too good to be true: The inverted U-shaped relationship between home-country digitalization and environmental performance. *Ecological Economics*, Volume 196. <https://doi.org/10.1016/j.ecolecon.2022.107393> (viitattu 17.7.2023).

Basecamp i.a. Leaving the Cloud: Cloud Computing Isn't For Everyone. Verkkosivu. <https://basecamp.com/cloud-exit> (viitattu 3.11.2023).

Bhattacharya, D., Chen, J. & Willee, H. 2024. Unicode-range. Verkkosivu. <https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face/unicode-range> (viitattu 25.3.2024).

Brandenburg, K., 1999. MP3 and AAC Explained. Firenze, Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding.

Burny. 2017. File:JPEG Generation Loss rotating 90 (stitch of 0,100,200,500,900,2000 times).png. Verkkosivu. [https://commons.wikimedia.org/wiki/File:JPEG\\_Generation\\_Loss\\_rotating\\_90\\_\(stitch\\_of\\_0,100,200,500,900,2000\\_times\).png](https://commons.wikimedia.org/wiki/File:JPEG_Generation_Loss_rotating_90_(stitch_of_0,100,200,500,900,2000_times).png) (viitattu 9.4.2024).

Chachou, T., Hamidouche, W., Fezza, S. A. & Belalem, G., 2023. Energy Consumption and Carbon Footprint of Modern Video Decoding Software. Poitiers, IEEE. <https://doi.org/10.1109/MMSP59012.2023.10337654> (viitattu 8.4.2024).

Chan-Jong-Chu Kwame, Islam Tanjina, Morales Exposito Miguel, Sheombar Sanjay, Valladares Christian, Philippot Olivier, Martino Grua Eoin & Malavolta Ivano 2020. Investigating the Correlation between Performance Scores and Energy Consumption of Mobile Web Apps. Trondheim, Association for Computing Machinery. <https://doi.org/10.1145/3383219.3383239> (viitattu 9.4.2024).

Cloudinary 2024. SVG Format: Features, Common Uses, and Pros/Cons You Should Know. Verkkosivu. <https://cloudinary.com/guides/image-formats/svg-format-features-common-uses-and-pros-cons-you-should-know-2#complex-images> (viitattu 9.4.2024).

Constable, P., Turetzky, K., Nihar, V. & Jacobs, M., 2022. OpenType Font Variations Overview. Verkkosivu. <https://learn.microsoft.com/enus/typography/opentype/spec/otvaroverview> (viitattu 25.3.2024).

Davies, H., 2023. What is OxygenOS? All about the OnePlus operating system. Verkkosivu. <https://www.trustedreviews.com/explainer/what-is-oxygenos-4300088> (viitattu 26.3.2024).

Energiavirasto, 2024. Energiatehokkuus. Verkkosivu. <https://energiavirasto.fi/energiatehokkuus> (viitattu 10.4.2024).

Ginesu, G., Pintus, M. & Giusto, D. D., 2012. Objective assessment of the WebP image coding algorithm. Signal Processing: Image Communication, 27(8), s. 867-874.

Google, 2023. An image format for the Web. Verkkosivu. <https://developers.google.com/speed/webp> (viitattu 25.3.2024).

Greenwood, Tom 2021. Sustainable Web Design. E-kirja. A Book Apart. <https://metropolia.finna.fi/Record/nelli15.4100000011763455> (viitattu 3.11.2023). Rajoitettu käyttöoikeus.

Grigorik, I. & Wagner, J., 2023. Optimize the encoding and transfer size of text-based assets. Verkkosivu. [https://web.dev/articles/optimizing-content-efficiency-optimize-encoding-and-transfer#text\\_compression\\_with\\_compression\\_algorithms](https://web.dev/articles/optimizing-content-efficiency-optimize-encoding-and-transfer#text_compression_with_compression_algorithms) (viitattu 2.4. 2024).

Heppell, J. & Bhattacharya, D., 2024. Web video codec guide. Verkkosivu. [https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video\\_codecs](https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs) (viitattu 25.3.2024).

Hoene, C., Valin, J.-M., Vos, K. & Jan, S., 2013. Summary of Opus listening test results. Verkkosivu. <https://datatracker.ietf.org/doc/html/draft-ietf-codec-results-03> (viitattu 9.4.2024).

HTTP Archive a i.a. Page Weight. Verkkosivu. [https://httparchive.org/reports/page-weight?lens=top100k&start=2023\\_06\\_01&view=list](https://httparchive.org/reports/page-weight?lens=top100k&start=2023_06_01&view=list) (viitattu 14.12.2023).

HTTP Archive b i.a. State of the Web. Verkkosivu. <https://httparchive.org/reports/state-of-the-web#bytesTotal> (viitattu 07.5.2023).

International Energy Agency i.a. Data Centres and Data Transmission Networks. Verkkosivu. <https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks> (viitattu 3.11.2023).

IO-tech, 2021. OLED- vai LCD-näyttö kannettavassa tietokoneessa? Verkkosivu. <https://www.io-tech.fi/artikkelit/oled-vai-lcd-naytto-kannettavassa-tietokoneessa/> (viitattu 15.3. 2024).

Irani, F., 2023. Power consumption of JPEG, WebP, and AVIF. Verkkosivu. <https://fershad.com/writing/power-consumption-jpeg-webp-and-avif/> (viitattu 10.4. 2024).

- Lange, Steffen & Pohl Johanna & Santarius Tilman 2020. Digitalization and energy consumption. Does ICT reduce energy demand? Ecological Economics, Volume 176. <https://doi.org/10.1016/j.ecolecon.2020.106760> (viitattu 15.7.2023).
- Larsen, R. & Oliff, C., 2023. HTML5 Boilerplate. Verkkosivu. <https://github.com/h5bp/html5-boilerplate> (viitattu 5.11.2023).
- Mavlankar, A. ym., 2020. AVIF for Next-Generation Image Coding. Verkkosivu. <https://netflixtechblog.com/avif-for-next-generation-image-coding-b1d75675fe4> (viitattu 6.4.2024).
- Mills, C. & Weyl, E., 2024a. CSS performance optimization. Verkkosivu. <https://developer.mozilla.org/en-US/docs/Learn/Performance/CSS> (viitattu 25.3.2024).
- Mills, C. & Weyl, E., 2024b. CSS performance optimization. Verkkosivu. [https://developer.mozilla.org/enUS/docs/Learn/Performance/CSS#loading\\_only\\_the\\_glyphs\\_you\\_need](https://developer.mozilla.org/enUS/docs/Learn/Performance/CSS#loading_only_the_glyphs_you_need) (viitattu 25.3.2024).
- Moreau, Pierre-Etienne 2023. Argos: Comparing the Energy Consumption of Two Web Stacks. Blogi 4.3.2023. Marmelab. <https://marmelab.com/blog/2021/03/04/argos-comparing-the-energy-consumption-of-two-web-stacks.html> (viitattu 3.10.2023).
- Mozilla, 2024a. Fundamental text and font styling. Verkkosivu. [https://developer.mozilla.org/enUS/docs/Learn/CSS/Styling\\_text/Fundamentals#fonts](https://developer.mozilla.org/enUS/docs/Learn/CSS/Styling_text/Fundamentals#fonts) (viitattu 12.2.2024).
- Mozilla, 2024b. Wake Lock Demo. Verkkosivu. <https://mdn.github.io/dom-examples/screen-wake-lock-api/> (viitattu 8.11.2023).
- Opus, 2024. Opus Interactive Audio Codec. Verkkosivu. <https://opus-codec.org/> (viitattu 9.4.2024).
- Pereira, Rui & Couto, Marco & Ribeiro, Francisco & Rua, Rui & Cunha, Jácome & Fernandes, João Paulo & Saraiva, João. 2017. Energy efficiency across programming languages: how do energy, time, and memory relate? Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, 256–267. <https://doi.org/10.1145/3136014.3136031> (viitattu 3.11.2023).
- Piątkowski, Dominik & Puślecki, Tobiasz & Walkowiak, Krzysztof. 2024. "Study of the Impact of Data Compression on the Energy Consumption Required for Data Transmission in a Microcontroller-Based System" Sensors 24, no. 1: 224. <https://doi.org/10.3390/s24010224> (viitattu 13.4.2024).
- Rastogi, Harshit 2023. Share of OLED Smartphones at Record High. Blogi 9.6.2023. <https://www.counterpointresearch.com/insights/smartphone-oled-penetration-q1-2023/> (viitattu 14.12.2023).

Rawzor, 2015. The New Test Images. Verkkosivu. [http://imagecompression.info/test\\_images/](http://imagecompression.info/test_images/) (viitattu 9.4.2024).

Siipola, J., 2020. Is WebP really better than JPEG?. Verkkosivu. <https://siipo.la/blog/is-webp-really-better-than-jpeg> (viitattu 25.3.2024).

Suomen ympäristöministeriö 2023. Mitä on kestävä kehitys? Verkkosivu. <https://ym.fi/mita-on-kestava-kehitys> (viitattu 3.11.2023).

Sustainable Web Design, 2024. Are you considering accessibility and energy efficiency when making color choices? Verkkosivu. <https://sustainableweb-design.org/has-the-color-palette-prioritized-colors-that-used-less-energy-on-oled-screens/> (viitattu 18.10.2023).

Tung, Liam 2018. Google: Here's why dark mode massively extends your OLED phone's battery life. 9.11.2018. ZDNET Tech. <https://www.zdnet.com/article/google-heres-why-dark-mode-massively-extends-your-oled-phones-battery-life/> (viitattu 14.12.2023).

Ubl, M., 2020. Maximally optimizing image loading for the web. Verkkosivu. <https://www.industrialempathy.com/posts/image-optimizations/> (viitattu 25.3.2024).

Vardhman, Raj 2023. 15 Crucial Data Center Statistics to Know in 2023. Tech Trends. Blogi 27.7.2023. Techjury.net. <https://techjury.net/blog/data-center-statistics/> (viitattu 3.10.2023).

W3Techs, 2024a. Usage statistics of server-side programming languages for websites. Verkkosivu. [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language) (viitattu 25.3.2024).

W3Techs, 2024b. Usage statistics of client-side programming languages for websites. Verkkosivu. [https://w3techs.com/technologies/overview/client\\_side\\_language](https://w3techs.com/technologies/overview/client_side_language) (viitattu 25.3. 2024).

## Liitteet

### MediaInfo-työkalun antamat tiedot videotiedostosta

The screenshot displays the MediaInfo application window. The title bar reads "MediaArea.net/MediaInfo - C:\Users\juhop\Desktop\Oppari\animaatio\postilaatikko-shotcut.mp4". The menu bar includes "File", "View", "Options", "Debug", "Help", and "Language", with a notification "A new version is available". The main content area shows a tree view with "General" and "Video" sections expanded. The "General" section lists: Complete name: C:\Users\juhop\Desktop\Oppari\animaatio\postilaatikko-shotcut.mp4, Format: MPEG-4, Format profile: Base Media, Codec ID: isom (isom/iso2/avc1/mp41), File size: 49.0 KiB, Duration: 3 s 502 ms, Overall bit rate: 115 kb/s, Frame rate: 53.120 FPS, and Writing application: Lavf60.3.100. The "Video" section lists: ID: 1, Format: AVC, Format/Info: Advanced Video Codec, Format profile: Constrained Baseline@L3.1, Format settings: 3 Ref Frames, Format settings, CABAC: No, Format settings, Reference frames: 3 frames, Codec ID: avc1, Codec ID/Info: Advanced Video Coding, Duration: 3 s 502 ms, Bit rate: 111 kb/s, Width: 700 pixels, Height: 400 pixels, Display aspect ratio: 1.85:1, Frame rate mode: Constant, Frame rate: 53.120 FPS, Color space: YUV, Chroma subsampling: 4:2:0, Bit depth: 8 bits, Scan type: Progressive, Bits/(Pixel\*Frame): 0.007, Stream size: 47.4 KiB (97%), Writing library: x264 core 164 r3107M a8b68eb, Encoding settings: cabac=0 / ref=3 / deblock=1:0:0 / analyse=0x1:0x111 / me=hex / subme=7 / psy=1 / Color range: Limited, Color primaries: BT.709, Transfer characteristics: BT.709, Matrix coefficients: BT.709, and Codec configuration box: avcC.

MediaArea.net/MediaInfo - C:\Users\juhop\Desktop\Oppari\animaatio\postilaatikko-shotcut.mp4

File View Options Debug Help Language | A new version is available

▼ C:\Users\juhop\Desktop\Oppari\animaatio\postilaatikko-shotcut.mp4

▼ General

- ... Complete name: C:\Users\juhop\Desktop\Oppari\animaatio\postilaatikko-shotcut.mp4
- ... Format: MPEG-4
- ... Format profile: Base Media
- ... Codec ID: isom (isom/iso2/avc1/mp41)
- ... File size: 49.0 KiB
- ... Duration: 3 s 502 ms
- ... Overall bit rate: 115 kb/s
- ... Frame rate: 53.120 FPS
- ... Writing application: Lavf60.3.100

▼ Video

- ... ID: 1
- ... Format: AVC
- ... Format/Info: Advanced Video Codec
- ... Format profile: Constrained Baseline@L3.1
- ... Format settings: 3 Ref Frames
- ... Format settings, CABAC: No
- ... Format settings, Reference frames: 3 frames
- ... Codec ID: avc1
- ... Codec ID/Info: Advanced Video Coding
- ... Duration: 3 s 502 ms
- ... Bit rate: 111 kb/s
- ... Width: 700 pixels
- ... Height: 400 pixels
- ... Display aspect ratio: 1.85:1
- ... Frame rate mode: Constant
- ... Frame rate: 53.120 FPS
- ... Color space: YUV
- ... Chroma subsampling: 4:2:0
- ... Bit depth: 8 bits
- ... Scan type: Progressive
- ... Bits/(Pixel\*Frame): 0.007
- ... Stream size: 47.4 KiB (97%)
- ... Writing library: x264 core 164 r3107M a8b68eb
- ... Encoding settings: cabac=0 / ref=3 / deblock=1:0:0 / analyse=0x1:0x111 / me=hex / subme=7 / psy=1 /
- ... Color range: Limited
- ... Color primaries: BT.709
- ... Transfer characteristics: BT.709
- ... Matrix coefficients: BT.709
- ... Codec configuration box: avcC



## Termit ja lyhenteet

|         |   |
|---------|---|
| AAC     | Advanced Audio Coding on häviöllinen tiedonpakkaus, joka on kehitetty puhutun äänen ja musiikin pakkaamiseen  |
| AV1     | AOMedia Video 1 on Alliance for Open Median kehittämä avoin ja ilmainen videokoodausformaatti, joka tehtiin videon siirtämiseksi Internetissä   |
| AVC     | Advanced Video Coding on videon pakkausstandardi (sama kuin H.264 ja MPEG-4 Part 10)  |
| AVIF    | AV1 Image File Format on kuvatiedostoformaatti, jolla tallennetaan kuvia tai kuvasekvenssejä, jotka on pakattu AV1-säilöformaatilla   |
| CSS     | Cascading Style Sheets on tyylitiedostokieli, jota käytetään HTML- tai XML-tiedoston ulkoasun kuvaamiseen   |
| EXIF    | Exchangeable image file format -metatietoihin voidaan tallentaa mm. aikaleima ja paikkatieto  |
| Full HD | Teräväpiirtovideo resoluutiolla 1920x1080 pikseliä  |
| GIF     | Graphics Interchange Format on häviötön bittikarttagrafiikan tallennusformaatti   |
| HTML    | Hypertext Markup Language on avoimen standardin merkintäkieli jota käytetään internetsivujen tekemiseen   |
| IPTC    | International Press Telecommunications Councilin kehittämä tietojen vaihtomalli on tiedostorakenne ja joukko metatietoattributteja, jotka on kehitetty nopeuttamaan kansainvälistä uutisten vaihtoa sanomalehtien ja uutistoimistojen välillä |

|           |  |
|-----------|--|
| JPEG, JPG | Joint Photographic Experts Group on häviöllistä pakkausta käyttävä bittikarttagrafiikan tallennusformaatti   |
| JSON      | JavaScript Object Notation on avoimen standardin tiedostomuoto tiedonvälitykseen ja tallennukseen  |
| LCD       | Liquid Crystal Display nestekidenäyttö   |
| MIME      | Multipurpose Internet Mail Extension -sisältötyyppejä käytetään ilmaisemaan välitetyn datan muotoa   |
| MP3       | MPEG-1 Audio Layer III on MPEG-1-standardiin perustuva äänenpakkausmenetelmä   |
| OLED      | Organic Light Emitting Diode on litteä näyttö, jonka toiminta perustuu orgaaniseen aineeseen, joka käyttäytyy samalla tavalla kuin diodi               |
| OPUS      | Häviöllinen, lisenssimaksuton ja avoin äänenpakkausmenetelmä   |
| PNG       | Portable Network Graphics (lausutaan "ping") on häviötöntä pakkausta käyttävä bittikarttagrafiikan tallennusformaatti                                  |
| SVG       | Scalable Vector Graphics on vektorigrafiikkaformaatti  |
| WEBP      | Googlen kehittämä bittikarttagrafiikan tiedostomuoto, jossa voidaan käyttää häviöllistä ja häviötöntä pakkausta  |
| XML       | Extensible Markup Language -merkitäkielellä datan joukkoon lisätään sen merkitystä kuvaavaa tietoa   |
| XMP       | Extensive Metadata Platform on ISO-standardi, joka on kehitetty standardisoidun ja räätälöidyn metadatan luontia, prosessointia ja välittämistä varten |

**.htaccess tiedostoon lisätyt rivit:**

```
<IfModule mod_headers.c>  
  Header set Cache-Control "no-cache, no-store, must-revalidate"  
  Header set Pragma "no-cache"  
  Header set Expires 0  
</IfModule>
```

## Hitaan ja raskaan toteutuksen HTML-lähdekoodi

```
<!DOCTYPE html>
<html class="no-js" lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Raskas ja hidas</title>
    <meta name="description" content="" />
    <meta name="viewport" content="width=device-width, initial-scale=1"
  />

  <meta http-equiv="Cache-control" content="no-store" />

  <meta property="og:title" content="" />
  <meta property="og:type" content="" />
  <meta property="og:url" content="" />
  <meta property="og:image" content="" />

  <link rel="manifest" href="site.webmanifest" />
  <link rel="apple-touch-icon" href="icon.png" />
  <!-- Place favicon.ico in the root directory -->

  <link rel="stylesheet" href="css/normalize.css" />
  <link rel="stylesheet" href="css/main.css" />

  <meta name="theme-color" content="#fafafa" />
</head>

<body>
  <div class="topnav">
    <a class="active" href="#Koti">Koti</a>
    <a href="#Uutiset">Uutiset</a>
    <a href="#Yhteystiedot">Yhteystiedot</a>
    <a href="#Tietoa">Tietoa</a>
  </div>

  <div class="hero-container">
    <div class="content"><h1 class="title">Otsikko</h1></div>
  </div>

  <div class="video-container">
    <video class="testivideo" autoplay muted controls>
      <source src="img/H264-720-AVGBR849-23976fpspeak.mp4" type="vi-
deodeo/mp4" />

```

Download the

```
<a href="img/H264-720-AVGBR849-23976fpspeak.mp4">mp4</a>
  video.
</video>
</div>

<div class="main">
  <!-- Mitä näille tehdään? -->
  <script src="js/vendor/modernizr-3.11.2.min.js"></script>
  <script src="js/plugins.js"></script>
  <!-- <script src="js/main.js"></script> -->
  <p>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quidem
    laboriosam ullam sapiente rerum accusantium quod reiciendis
    reprehenderit corrupti, dolor nulla molestias at expedita sit
    quibusdam
    temporibus sint nam fugit dolore eum? Reprehenderit debitis quia
    assumenda consequatur officiis! Dicta placeat nesciunt quod ve-
    niam,
    ratione asperiores hic quo laboriosam obcaecati perferendis na-
    tus, quam
    fugiat nemo voluptatem doloremque commodi ut recusandae quas
    atque quos
    dolores repudiandae. Rerum, mollitia. Explicabo eos nobis eum la-
    bore,
    tenetur, exercitationem maiores et eius, sit laudantium repudian-
    dae
    rerum excepturi ratione quia? Repellat unde aut recusandae expli-
    cabo
    voluptates veniam vero porro adipisci illum saepe quidem iure,
    odit eos
    harum debitis ad quas dignissimos est ducimus nisi officia.
    Suscipit
    ipsum deserunt repudiandae? Modi, aspernatur accusantium sunt aut
    tempore deleniti aperiam cumque optio dicta laborum repellendus
    vero est
    dolore nulla illum consequuntur incidunt impedit quaerat perfe-
    rendis
    excepturi culpa aliquid nostrum?
  </p>
  <p>
    Facilis accusamus veritatis repellat eos, in minima aliquid! Re-
    pellat
    blanditiis voluptatum dolorem saepe perferendis maiores consec-
    tetur
    eligendi nisi deleniti neque in, eaque, vero quod quam quae iste
    odit
    molestias quasi aliquam cumque porro harum dicta fugiat eius?
```

Necessitatibus quaerat perferendis, nobis nesciunt hic vitae nisi saepe.

Reprehenderit eaque minus alias praesentium, ad doloremque iusto commodi fugiat! Facere, minima non eos doloremque saepe tenetur maxime eum tempora! Nulla vitae ipsa reprehenderit nesciunt earum sed tempora dolore cum nihil corporis commodi, placeat neque tempore velit unde.

Dicta, eveniet fuga. Similique inventore reprehenderit vitae magni optio a quod id autem dicta eum quos voluptatibus ipsum, soluta saepe blanditiis totam ducimus ut quo, eius nesciunt doloribus vel repellat?

Ad, inventore magni ratione voluptatibus quidem, minima itaque quia quam optio rerum sed illo dolore velit. Possimus quasi doloribus commodi fugiat, nam dolor a dolore eveniet vitae iusto molestias repellendus fuga quia voluptatum veniam voluptates qui quibusdam minus odit doloremque? Odio laboriosam architecto fugiat amet quis, consequuntur animi perferendis ullam dolore asperiores, vel nostrum totam sapiente modi iure quas deleniti numquam ipsam? Doloremque labore minus optio, totam nam magni laudantium facere possimus placeat doloribus natus sequi quos earum ex praesentium deleniti veniam impedit, non sed quidem cupiditate. Tempore ipsum sit laborum excepturi! Similique, officiis, numquam veniam sed hic quis quae recusandae optio voluptate fugit, vero maiores? Soluta voluptatem et quae mollitia cum nemo ratione quod tenetur expedita itaque, velit odit. Asperiores nam odit aperiam impedit cumque sit?

</p>

<p>

Ut commodi, voluptatem. Facilis accusamus veritatis repellat eos, in minima aliquid! Repellat blanditiis voluptatum dolorem saepe perferendis maiores consecetur eligendi nisi deleniti neque in, eaque, vero quod

quam quae iste odit molestias quasi aliquam cumque porro harum  
 dicta  
 fugiat eius? Necessitatibus quaerat perferendis, nobis nesciunt  
 hic  
 vitae nisi saepe. Reprehenderit eaque minus alias praesentium, ad  
 doloremque iusto commodi fugiat! Facere, minima non eos dolorem-  
 que saepe  
 tenetur maxime eum tempora! Nulla vitae ipsa reprehenderit nes-  
 ciunt  
 earum sed tempora dolore cum nihil corporis commodi, placeat ne-  
 que  
 tempore velit unde. Dicta, eveniet fuga. Similique inventore  
 reprehenderit vitae magni optio a quod id autem dicta eum quos  
 voluptatibus ipsum, soluta saepe blanditiis totam ducimus ut quo,  
 eius  
 nesciunt doloribus vel repellat? Ad, inventore magni ratione  
 voluptatibus quidem, minima itaque quia quam optio rerum sed illo  
 dolore  
 velit. Possimus quasi doloribus commodi fugiat, nam dolor a do-  
 lore  
 eveniet vitae iusto molestias repellendus fuga quia voluptatum  
 veniam  
 voluptates qui quibusdam minus odit doloremque? Odio laboriosam  
 architecto fugiat amet quis, consequuntur animi perferendis ullam  
 dolore  
 asperiores, vel nostrum totam sapiente modi iure quas deleniti  
 numquam  
 ipsam? Doloremque labore minus optio, totam nam magni laudantium  
 facere  
 possimus placeat doloribus natus sequi quos earum ex praesentium  
 deleniti veniam impedit, non sed quidem cupiditate. Tempore ipsum  
 sit  
 laborum excepturi! Similique, officiis, numquam veniam sed hic  
 quis quae  
 recusandae optio voluptate fugit, vero maiores? Soluta voluptatem  
 et  
 quae mollitia cum nemo ratione quod tenetur expedita itaque, ve-  
 lit odit.  
 Asperiores nam odit aperiam impedit cumque sit?  
 </p>  
 <p>  
 dolore ipsa tempora consecetur porro quasi eaque! Voluptates  
 atque eius  
 et esse sapiente suscipit vel corrupti sit reiciendis voluptatum  
 delectus a cumque cum, unde dolorum aut. Illum dolorum in odio  
 perspiciatis deleniti minus deserunt laborum, pariatum nostrum  
 porro  
 dolorem maiores nemo ad sequi, labore saepe ex aut aperiam sint

reprehenderit neque corrupti. Fuga, beatae, explicabo temporibus repudiandae qui nemo eligendi eveniet eos quia unde dolor rem obcaecati quam ullam accusantium magni itaque possimus? Ipsa consecetur inventore reprehenderit ipsam! Nobis vero nesciunt officia sint eaque. Dicta tempora at inventore error debitis. Magnam omnis aut consequatur soluta? Quos corrupti commodi ad facilis vitae possimus? Voluptatum, temporibus quo numquam voluptate odit neque! Autem nam doloribus eligendi dolores corporis aspernatur veritatis pariatur recusandae preferendis? Laudantium quis itaque atque labore repudiandae voluptatem obcaecati quam molestias sapiente vero mollitia impedit esse fuga animi dolore, quod veritatis dignissimos? Error sapiente amet in distinctio ipsum ullam similique reiciendis iusto? Praesentium incidunt assumenda obcaecati aut est? Similique quae, facilis id voluptatem nulla minus! Architecto iure eaque voluptatibus ipsam vel quos quod ad, nulla unde impedit hic, optio autem officia esse iusto alias rem earum corrupti rerum facere dolor? Repellat, cumque exercitationem quas accusantium placeat quibusdam accusamus debitis labore dolorum ratione doloribus soluta reprehenderit vitae nisi unde in? Libero totam numquam tenetur repudiandae reprehenderit consequatur itaque beatae officiis possimus odit quos, velit necessitatibus molestiae eligendi, magnam modi ipsam maiores deleniti quo asperiores quaerat nesciunt voluptatem dolor temporibus! Cupiditate earum, accusamus ipsam, vel illo officia esse reiciendis nobis aspernatur doloribus tenetur? Quia ea, distinctio consecetur minus temporibus nemo minima repudiandae perspiciatis ullam, adipisci vel veritatis ducimus nisi? Quae, repellendus cupiditate. Atque mollitia ipsum, dolore cumque iste placeat illo ducimus necessitatibus



consequatur laborum, voluptas eos voluptates soluta vel aliquam  
delectus  
adipisci repellat. Recusandae tempora exercitationem fugiat accu-  
santium  
adipisci quod quaerat nobis perspiciatis expedita voluptas aspe-  
riores  
porro quas officia eos necessitatibus minus sint, fugit ex totam  
temporibus. Illum placeat fugit exercitationem repellendus debi-  
tis  
veniam? Voluptate eum vitae ex assumenda nam exercitationem quo a  
dolorum quasi nihil harum tempora praesentium voluptatibus de-  
leniti  
itaque quia fugiat, corrupti, deserunt explicabo. Eum impedit de-  
bitis  
molestias vitae laborum assumenda, unde, harum quae velit, porro  
quasi  
laboriosam maiores tenetur vero quo doloribus! Exercitationem il-  
lum,  
adipisci quia magnam veritatis iusto magni mollitia neque eum  
quos  
eveniet? Animi, libero provident iure cum illum at autem eum du-  
cimus  
consequuntur ipsa necessitatibus enim iusto porro eligendi culpa  
cupiditate odit quidem id natus aperiam accusamus?

</p>  
<p>  
Quae non perspiciatis dolor, sint facere quia placeat id? Rerum  
officia  
dolorum ipsam optio perspiciatis sit nostrum debitis, dolore ul-  
lam saepe  
facilis numquam dolor architecto eaque! Magnam, cupiditate. Dolo-  
remque  
eum voluptatum in veritatis repudiandae ducimus eveniet! In harum  
quod  
deserunt. Qui nemo cupiditate quam nostrum laudantium ad incidunt  
non  
quia culpa minus explicabo omnis eum, maiores placeat ab ipsum  
excepturi  
tempore, voluptatum quos illum ea doloribus consequuntur. Dolori-  
bus,  
veritatis explicabo ipsam corporis eos voluptatibus fuga optio  
labore ab  
eius blanditiis accusantium quam impedit quis nesciunt numquam  
aperiam  
hic minus est nisi? Consequuntur ex corrupti eveniet culpa, et ut  
esse  
laborum vero, minima dolorem harum accusantium eum! Quae labori-  
osam,

```
    amet rem commodi natus obcaecati neque consecetur illo recusandae porro dignissimos deleniti sunt. Dolorum ipsa ullam dicta architecto numquam. Temporibus corporis sunt error quasi aperiam esse molestiae, explicabo expedita itaque asperiores, magnam ut harum, recusandae enim? Perspiciatis delectus eos praesentium suscipit libero perferendis magni magnam quidem quam odio temporibus iure corporis atque nulla harum consequatur, vitae nostrum possimus necessitatibus reiciendis laboriosam a accusantium doloribus. Numquam reiciendis vitae, nisi asperiores laborum aspernatur architecto maxime. Maxime maiores porro sit neque aperiam! Tempora cumque facilis nostrum cupiditate aut nihil nobis, praesentium placeat consequatur beatae sequi doloribus. Expedita tenetur nam delectus saepe unde suscipit, adipisci ad veritatis eaque molestiae possimus quam consecetur quibusdam?
</p>

<!-- Kuvat -->
<div class="img-container">
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
</div>
```

```
</div>
<div class="img-item">
  
</div>
<div class="img-item">
  
</div>

<div class="small-img-cont">
  <div class="img-item">
    
  </div>
  <div class="img-item">
    >
  </div>
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
</div>
</div>

<div class="audio-container">
  <audio id="classic" autoplay controls>
    <source
      src="img/Concerto_for_Violoncello_and_Orchestra_in_B_mi-
nor.mp3"
      type="audio/mpeg"
    />
  </audio>
</div>
```

```
<div class="anim-container">
  
</div>
</div>

<script>
window.onload = () => {
  const audio = document.getElementById("classic");
  const play = document.getElementById("play");
  audio.play();

  audio.addEventListener("ended", (event) => {
    console.log("VALMIS");
    location.reload(true);
  });
};
</script>

</body>
</html>
```

**Erittäin nopean ja kevyen toteutuksen HTML-lähdekoodi**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Erinopee</title>
  <meta http-equiv="refresh" content="60">
  <meta http-equiv="Cache-control" content="no-store">
  <link rel="stylesheet" href="css/main.css">
</head>
<body>

  <div class="topnav">
    <a class="active" href="#Koti">Koti</a>
    <a href="#Uutiset">Uutiset</a>
    <a href="#Yhteystiedot">Yhteystiedot</a>
    <a href="#Tietoa">Tietoa</a>
  </div>

  <div class="hero-container">
    <div class="content"><h1 class="title">Otsikko</h1></div>
  </div>

  <div class="main">

    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quidem
      laboriosam ullam sapiente rerum accusantium quod reiciendis
      reprehenderit corrupti, dolor nulla molestias at expedita sit
      quibusdam
      temporibus sint nam fugit dolore eum? Reprehenderit debitis quia
      assumenda consequatur officiis! Dicta placeat nesciunt quod ve-
      niam,
      ratione asperiores hic quo laboriosam obcaecati perferendis na-
      tus, quam
      fugiat nemo voluptatem doloremque commodi ut recusandae quas
      atque quos
      dolores repudiandae. Rerum, mollitia. Explicabo eos nobis eum la-
      bore,
      tenetur, exercitationem maiores et eius, sit laudantium repudian-
      dae
      rerum excepturi ratione quia? Repellat unde aut recusandae expli-
      cabo
      voluptates veniam vero porro adipisci illum saepe quidem iure,
      odit eos
```

harum debitis ad quas dignissimos est ducimus nisi officia.

Suscipit  
 ipsum deserunt repudiandae? Modi, aspernatur accusantium sunt aut  
 tempore deleniti aperiam cumque optio dicta laborum repellendus  
 vero est  
 dolore nulla illum consequuntur incidunt impedit quaerat perfe-  
 rendis  
 excepturi culpa aliquid nostrum?

</p>  
 <p>  
 Facilis accusamus veritatis repellat eos, in minima aliquid! Re-  
 pellat  
 blanditiis voluptatum dolorem saepe perferendis maiores consec-  
 tetur  
 eligendi nisi deleniti neque in, eaque, vero quod quam quae iste  
 odit  
 molestias quasi aliquam cumque porro harum dicta fugiat eius?  
 Necessitatibus quaerat perferendis, nobis nesciunt hic vitae nisi  
 saepe.  
 Reprehenderit eaque minus alias praesentium, ad doloremque iusto  
 commodi  
 fugiat! Facere, minima non eos doloremque saepe tenetur maxime  
 eum  
 tempora! Nulla vitae ipsa reprehenderit nesciunt earum sed tem-  
 pora  
 dolore cum nihil corporis commodi, placeat neque tempore velit  
 unde.  
 Dicta, eveniet fuga. Similique inventore reprehenderit vitae  
 magni optio  
 a quod id autem dicta eum quos voluptatibus ipsum, soluta saepe  
 blanditiis totam ducimus ut quo, eius nesciunt doloribus vel re-  
 pellat?  
 Ad, inventore magni ratione voluptatibus quidem, minima itaque  
 quia quam  
 optio rerum sed illo dolore velit. Possimus quasi doloribus com-  
 modi  
 fugiat, nam dolor a dolore eveniet vitae iusto molestias repel-  
 lendus  
 fuga quia voluptatum veniam voluptates qui quibusdam minus odit  
 doloremque? Odio laboriosam architecto fugiat amet quis, conse-  
 quuntur  
 animi perferendis ullam dolore asperiores, vel nostrum totam sa-  
 piente  
 modi iure quas deleniti numquam ipsam? Doloremque labore minus  
 optio,  
 totam nam magni laudantium facere possimus placeat doloribus na-  
 tus sequi  
 quos earum ex praesentium deleniti veniam impedit, non sed quidem

cupiditate. Tempore ipsum sit laborum excepturi! Similique, offi-  
 ciis,  
 numquam veniam sed hic quis quae recusandae optio voluptate fu-  
 git, vero  
 maiores? Soluta voluptatem et quae mollitia cum nemo ratione quod  
 tenetur expedita itaque, velit odit. Asperiores nam odit aperiam  
 impedit  
 cumque sit?  
 </p>  
 <p>  
 Ut commodi, voluptatem. Facilis accusamus veritatis repellat eos,  
 in  
 minima aliquid! Repellat blanditiis voluptatum dolorem saepe per-  
 ferendis  
 maiores consecetur eligendi nisi deleniti neque in, eaque, vero  
 quod  
 quam quae iste odit molestias quasi aliquam cumque porro harum  
 dicta  
 fugiat eius? Necessitatibus quaerat perferendis, nobis nesciunt  
 hic  
 vitae nisi saepe. Reprehenderit eaque minus alias praesentium, ad  
 doloremque iusto commodi fugiat! Facere, minima non eos dolorem-  
 que saepe  
 tenetur maxime eum tempora! Nulla vitae ipsa reprehenderit nes-  
 ciunt  
 earum sed tempora dolore cum nihil corporis commodi, placeat ne-  
 que  
 tempore velit unde. Dicta, eveniet fuga. Similique inventore  
 reprehenderit vitae magni optio a quod id autem dicta eum quos  
 voluptatibus ipsum, soluta saepe blanditiis totam ducimus ut quo,  
 eius  
 nesciunt doloribus vel repellat? Ad, inventore magni ratione  
 voluptatibus quidem, minima itaque quia quam optio rerum sed illo  
 dolore  
 velit. Possimus quasi doloribus commodi fugiat, nam dolor a do-  
 lore  
 eveniet vitae iusto molestias repellendus fuga quia voluptatum  
 veniam  
 voluptates qui quibusdam minus odit doloremque? Odio laboriosam  
 architecto fugiat amet quis, consequuntur animi perferendis ullam  
 dolore  
 asperiores, vel nostrum totam sapiente modi iure quas deleniti  
 numquam  
 ipsam? Doloremque labore minus optio, totam nam magni laudantium  
 facere  
 possimus placeat doloribus natus sequi quos earum ex praesentium  
 deleniti veniam impedit, non sed quidem cupiditate. Tempore ipsum  
 sit

laborum excepturi! Similique, officiis, numquam veniam sed hic  
quis quae  
recusandae optio voluptate fugit, vero maiores? Soluta voluptatem  
et  
quae mollitia cum nemo ratione quod tenetur expedita itaque, ve-  
lit odit.  
Asperiores nam odit aperiam impedit cumque sit?

</p>  
<p>  
dolore ipsa tempora consecetur porro quasi eaque! Voluptates  
atque eius  
et esse sapiente suscipit vel corrupti sit reiciendis voluptatum  
delectus a cumque cum, unde dolorum aut. Illum dolorum in odio  
perspiciatis deleniti minus deserunt laborum, pariatum nostrum  
porro  
dolorem maiores nemo ad sequi, labore saepe ex aut aperiam sint  
reprehenderit neque corrupti. Fuga, beatae, explicabo temporibus  
repudiandae qui nemo eligendi eveniet eos quia unde dolor rem  
obcaecati  
quam ullam accusantium magni itaque possimus? Ipsa consecetur  
inventore  
reprehenderit ipsam! Nobis vero nesciunt officia sint eaque.  
Dicta  
tempora at inventore error debitis. Magnam omnis aut consequatur  
soluta?  
Quos corrupti commodi ad facilis vitae possimus? Voluptatum, tem-  
poribus  
quo numquam voluptate odit neque! Autem nam doloribus eligendi  
dolores  
corporis aspernatur veritatis pariatum recusandae perferendis?  
Laudantium quis itaque atque labore repudiandae voluptatem  
obcaecati  
quam molestias sapiente vero mollitia impedit esse fuga animi do-  
lore,  
quod veritatis dignissimos? Error sapiente amet in distinctio ip-  
sum  
ullam similique reiciendis iusto? Praesentium incidunt assumenda  
obcaecati aut est? Similique quae, facilis id voluptatem nulla  
minus!  
Architecto iure eaque voluptatibus ipsam vel quos quod ad, nulla  
unde  
impedit hic, optio autem officia esse iusto alias rem earum cor-  
rupti  
rerum facere dolor? Repellat, cumque exercitationem quas accusan-  
tium  
placeat quibusdam accusamus debitis labore dolorum ratione dolo-  
ribus



soluta reprehenderit vitae nisi unde in? Libero totam numquam te-  
 netur  
 repudiandae reprehenderit consequatur itaque beatae officiis pos-  
 simus  
 odit quos, velit necessitatibus molestiae eligendi, magnam modi  
 ipsam  
 maiores deleniti quo asperiores quaerat nesciunt voluptatem dolor  
 temporibus! Cupiditate earum, accusamus ipsam, vel illo officia  
 esse  
 reiciendis nobis aspernatur doloribus tenetur? Quia ea, distinc-  
 tio  
 consecetur minus temporibus nemo minima repudiandae perspiciatis  
 ullam,  
 adipisci vel veritatis ducimus nisi? Quae, repellendus cupidi-  
 tate. Atque  
 mollitia ipsum, dolore cumque iste placeat illo ducimus necessi-  
 tatibus  
 consequatur laborum, voluptas eos voluptates soluta vel aliquam  
 delectus  
 adipisci repellat. Recusandae tempora exercitationem fugiat accu-  
 santium  
 adipisci quod quaerat nobis perspiciatis expedita voluptas aspe-  
 riores  
 porro quas officia eos necessitatibus minus sint, fugit ex totam  
 temporibus. Illum placeat fugit exercitationem repellendus debi-  
 tis  
 veniam? Voluptate eum vitae ex assumenda nam exercitationem quo a  
 dolorum quasi nihil harum tempora praesentium voluptatibus de-  
 leniti  
 itaque quia fugiat, corrupti, deserunt explicabo. Eum impedit de-  
 bitis  
 molestias vitae laborum assumenda, unde, harum quae velit, porro  
 quasi  
 laboriosam maiores tenetur vero quo doloribus! Exercitationem il-  
 lum,  
 adipisci quia magnam veritatis iusto magni mollitia neque eum  
 quos  
 eveniet? Animi, libero provident iure cum illum at autem eum du-  
 cimus  
 consequuntur ipsa necessitatibus enim iusto porro eligendi culpa  
 cupiditate odit quidem id natus aperiam accusamus?  
 </p>  
 <p>  
 Quae non perspiciatis dolor, sint facere quia placeat id? Rerum  
 officia  
 dolorum ipsam optio perspiciatis sit nostrum debitis, dolore ul-  
 lam saepe

facilis numquam dolor architecto eaque! Magnam, cupiditate. Dolo-  
 remque  
 eum voluptatum in veritatis repudiandae ducimus eveniet! In harum  
 quod  
 deserunt. Qui nemo cupiditate quam nostrum laudantium ad incidunt  
 non  
 quia culpa minus explicabo omnis eum, maiores placeat ab ipsum  
 excepturi  
 tempore, voluptatum quos illum ea doloribus consequuntur. Dolori-  
 bus,  
 veritatis explicabo ipsam corporis eos voluptatibus fuga optio  
 labore ab  
 eius blanditiis accusantium quam impedit quis nesciunt numquam  
 aperiam  
 hic minus est nisi? Consequuntur ex corrupti eveniet culpa, et ut  
 esse  
 laborum vero, minima dolorem harum accusantium eum! Quae labori-  
 osam,  
 amet rem commodi natus obcaecati neque consecetur illo recusandae porro  
 dignissimos deleniti sunt. Dolorum ipsa ullam dicta architecto  
 numquam.  
 Temporibus corporis sunt error quasi aperiam esse molestiae, explicabo  
 expedita itaque asperiores, magnam ut harum, recusandae enim?  
 Perspiciatis delectus eos praesentium suscipit libero perferendis  
 magni  
 magnam quidem quam odio temporibus iure corporis atque nulla harum  
 rum  
 consequatur, vitae nostrum possimus necessitatibus reiciendis laboriosam  
 a accusantium doloribus. Numquam reiciendis vitae, nisi asperiores  
 riores  
 laborum aspernatur architecto maxime. Maxime maiores porro sit  
 neque  
 aperiam! Tempora cumque facilis nostrum cupiditate aut nihil nobis,  
 bis,  
 praesentium placeat consequatur beatae sequi doloribus. Expedita  
 tenetur  
 nam delectus saepe unde suscipit, adipisci ad veritatis eaque molestiae  
 lestiae  
 possimus quam consecetur quibusdam?  
 </p>
 <div class="img-container">
 <div class="img-item">
 <a href="img/big\_tree.avif">Kuva isosta puusta</a>
 </div>

```
<div class="img-item">
  <a href="img/cathedral.avif">Kuva katedraalista</a>
</div>
<div class="img-item">
  <a href="img/deer.avif">Kuva peurasta</a>
</div>
<div class="img-item">
  <a href="img/flower_foveon.avif">Kuva kukasta</a>
</div>
<div class="img-item">
  <a href="img/spider_web.avif">Kuva hämähäkinverkosta</a>
</div>

<div class="small-img-cont">
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
  <div class="img-item">
    
  </div>
</div>
</div>

<div class="audio-container">
  <a href="img/Concerto_for_Violoncello_and_Orchestra_in_B_mi-
nor.opus">Musiikkinäyte</a>
</div>

<div class="anim-container">
  <a href="img/message-delivered-to-mailbox-animation.svg">Animaa-
tio</a>
</div>
</div>

</body>
</html>
```