

# Verkkosivuston toteuttaminen Webflow-työkalulla



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus  
syksy, 2022

Juuso Rutanen

Tietojenkäsittelyn koulutus

Tiivistelmä

Tekijä Juuso Rutanen

Vuosi 2022

Työn nimi Verkkosivuston toteuttaminen Webflow-työkalulla

Ohjaaja Pentti Ojaniemi

---

## TIIVISTELMÄ

Opinnäytetyön tarkoitus oli tutkia verkkosivujen rakennustyökalu Webflow'n hyötyjä verkkosivuston toteuttamiseen sekä minkälaisiin projekteihin Webflow'ta kannattaisi käyttää.

Opinnäytetyön teoreettisen osuuden alkupuoli koostuu verkkosivuston suunnittelun käytänteistä sekä verkkosivun toteuttamisen eri vaihtoehdoista. Lisäksi esitellään tarkemmin Webflow-työkalun ominaisuuksia ja sitä, miten se eroaa sen kilpailijoista.

Opinnäytetyön käytännön osuudessa toteutettiin verkkosivusto Webflow'n avulla ja esiteltiin sivustolla hyödynnettyjä Webflow'n ominaisuuksia. Verkkosivuston toteuttamisen perusteella voitiin tehdä johtopäätöksiä Webflow'n hyödyistä ja minkälaisiin projekteihin se sopii parhaiten.

Tutkimuksen käytännön osuuden perusteella voidaan esittää, että Webflow soveltuu etenkin sivustoille, joihin halutaan lisätä erilaisia toiminnallisia ominaisuuksia ja näyttävää grafiikkaa ilman ohjelmoinnin osaamista. Ohjelmoinnin osaaminen mahdollistaa kuitenkin vielä laajemman sivuston kustomoinnin ja varsinkin selainpuolen web-tekniikoiden osaamisesta on Webflow'n käytössä suuri hyöty. Webflow'n sisällönhallintajärjestelmä koettiin myös toimivaksi ja sisältöä on helppo päivittää. Tutkimuksen perusteella Webflow'n käyttöä voidaan suositella etenkin yrityksille ja yhdistyksille sekä yksityishenkilöille, jotka haluavat käyttöönsä laajat sivuston kustomointimahdollisuudet ja helppokäyttöisen sisällönhallintajärjestelmän.

Avainsanat Webflow, sisällönhallintajärjestelmä, verkkosivut, CSS, web-tekniologiat

Sivut 49 sivua ja liitteitä 1 sivu

Degree Programme in Business Information Technology	Abstract
Author Juuso Rutanen	Year 2022
Subject Creating a Website with Webflow	
Supervisor Pentti Ojaniemi	

---

## ABSTRACT

The purpose of this thesis is to research Webflow, which is an online platform for building websites. The objective was to research what advantages Webflow offers in creating websites and for what type of projects it is best suited.

The main points of website planning are discussed first in the theoretical background of the thesis, as well as key points in user interface design. Following sections of the theoretical background introduce Webflow and study its differences among other similar tools focusing mainly on WordPress.

In the practical part of the thesis, a website was created with Webflow, and the features of Webflow used on the website were presented. By creating a website with Webflow, conclusions could be drawn about the benefits of Webflow and what kind of projects it is best suited for.

Based on the practical part of this research, it can be suggested that Webflow is particularly suitable for companies, associations and individuals who want to create websites with creative visuals and interactive design features without the need of knowing advanced web techniques. In addition, Webflow's CMS-tool offers a simple way to create a content-oriented website, with a lot of possibilities for customization. However, knowing programming, and especially web programming, is beneficial when using Webflow.

Keywords Webflow, content management system, web site, CSS, web technologies

Pages 49 pages and appendices 1 page

## Sanasto

HTML	HyperText Markup Language, verkkosivujen määrittelykieli
CSS	Cascading Style Sheets, verkkosivujen tyylikieli
RGB	Red Green Blue, RGB-värimalli
PNG	Portable Network Graphics, bittikarttagrafiikan tallennusformaatti
JPEG	Joint Photographic Expert Group, bittikarttagrafiikan tallennusformaatti, JPG on myös sama tiedostomuoto
GIF	Graphics Interchange Format, bittikarttagrafiikan tallennusformaatti
WebP	Web Picture format, bittikarttagrafiikan tallennusformaatti
CSV	Comma Separated Values, tiedostomuoto taulukkomuotoiseen tietoon
CMS	Content Management System, sisällönhallintajärjestelmä
WCAG	Web Content Accessibility Guidelines, verkkosisällön saavutettavuusohjeet
W3C	The World Wide Web Consortium, kansainvälinen yritysten ja yhteisöjen yhteenliittymä, joka ylläpitää ja kehittää World Wide Webin standardeja
SSL	Secure Sockets Layer, tietoverkkosalausprotokolla
Backend	Palvelimen puolella toimiva ohjelmisto
Frontend	Ohjelmiston käyttäjälle näkyvä osa

## Sisälllys

1	Johdanto .....	1
2	Verkkosivuston suunnittelu ja rakenne.....	2
2.1	Verkkosivuston suunnitteluprosessi .....	2
2.1.1	Tavoitteen määrittäminen .....	2
2.1.2	Sivustokartan luominen .....	3
2.1.3	Rतालankamallin luominen .....	4
2.2	Käyttöliittymä.....	6
2.2.1	Verkkosivujen typografia.....	6
2.2.2	Värit verkkosivuilla .....	8
2.2.3	Kuvat.....	11
2.2.4	Responsiivisuus .....	12
2.3	Hakukoneoptimointi .....	13
3	Verkkosivuston toteutustavat .....	14
3.1	Verkkosivujen rakennustyökalut.....	14
3.2	Sisällönhallintajärjestelmät.....	15
3.3	Verkkosivuston ohjelmointi .....	16
4	Webflow .....	19
4.1	Webflow'n esittely .....	19
4.2	Webflow'n ja WordPressin eroavaisuuksia .....	21
5	Verkkosivuston luominen Webflow'n avulla.....	23
5.1	Sivuston suunnittelu .....	23
5.1.1	Tavoitteen määrittäminen .....	23
5.1.2	Sivuston rakenne ja rतालankamalli .....	23
5.1.3	Väripaletti ja typografia.....	24
5.2	Verkkosivuston luominen .....	26
5.3	Sivustolla käytetyt ominaisuudet.....	28
5.3.1	Webflow'n NavBar-elementti .....	28
5.3.2	Webflow'n Div Block-elementti .....	29
5.3.3	Webflow'n Grid-elementti .....	30
5.3.4	Tiedostojen lisääminen .....	31
5.3.5	CMS-työkalun käyttäminen.....	31
5.3.6	Sisällön päivittäminen .....	34
5.3.7	Webflow'n kehitysympäristö .....	34

5.3.8	Tyylipaneelin käyttäminen .....	38
5.3.9	Interaktiot-paneelin käyttäminen .....	41
5.3.10	Pysäytyspisteet.....	45
6	Johtopäätökset ja pohdinta .....	47
7	Yhteenveto .....	49
	Lähteet.....	50

## Kuvat, ohjelmakoodit ja taulukot

Kuva 1.	Vaakasuntainen yksinkertainen sivustokartta (Williams, ei pvm.-a, s. 10).....	4
Kuva 2.	Rautalankamallien eri tarkkuudet matalasta korkeaan (mentormate, n.d.). .....	5
Kuva 3.	Värien herättämät tuntemukset (usertesting.com, 2019). .....	9
Kuva 4.	Väriympyrä (Tidwell ym., 2020, s. 259). .....	10
Kuva 5.	Analoginen väripaletti (Cartwright, 2021). .....	11
Kuva 6.	Esimerkki pääotsikon HTML-elementistä ja HTML-tunnisteista. ....	17
Kuva 7.	Esimerkki CSS-säännöstä ja luokkavalitsimesta (web.dev, n.d.). .....	17
Kuva 8.	CSS:n laatikkomalli (w3schools, n.d.).....	18
Kuva 9.	Figmalla luotu rautalankamalli sivuston etusivusta. ....	24
Kuva 10.	Sivustolle valittu väripaletti. ....	25
Kuva 11.	Webflow'n paneeli värien hallintaan.....	25
Kuva 12.	Kontrastisuhteen tarkistus Webflow'ssa. ....	26
Kuva 13.	Designerin suunnittelupohja. ....	27
Kuva 14.	Webflow Designerin vasemman puolen työkaluvalikko ja oikean puolen paneelit elementtien muokkaamiseen. ....	27
Kuva 15.	Webflow'n Dashboard-osio. ....	28
Kuva 16.	Sivustolta löytyvä navigaatiovalikko. ....	29
Kuva 17.	Webflow'n koodieditorissa näkyvä div-tunniste. ....	29
Kuva 18.	Sivuston rakennetta Webflow'n suunnittelutyökalussa.....	30
Kuva 19.	Grid-elementin hyödyntäminen alatunnisteessa. ....	30
Kuva 20.	Tietokantaan valitut kentät. ....	32
Kuva 21.	Esimerkki CMS-työkalun hyödyntämisestä.....	33
Kuva 22.	Valmis lopputulos HTML-embed-elementillä lisätystä kartasta.....	35
Kuva 23.	Elementti suunnittelupohjalla ja luokan nimi.....	37

Kuva 24. Valmis lopputulos JavaScript-elementin käytöstä. (Huom. ylimmän rivin teksti kirjoittuu itsestään ja sana vaihtuu) .....	38
Kuva 25. Luokkavalitsin Webflow'n suunnittelutyökalussa. ....	39
Kuva 26. CSS-valitsimen koodi Webflow'n tuottamassa koodissa .....	39
Kuva 27. Globaalin luokan käyttäminen painikkeen varjostuksessa.....	40
Kuva 28. Spacing-työkalu.....	40
Kuva 29. Layout-työkalu. ....	41
Kuva 30. Elementin laukaisijat.....	42
Kuva 31. Interaktiivisen elementin ensimmäinen klikkaus. ....	42
Kuva 32. Interaktiivisen elementin toinen klikkaus.....	42
Kuva 33. Valmis lopputulos Tietoa meistä-osiosta.....	43
Kuva 34. Effect-työkalun hyödyntäminen. ....	43
Kuva 35. Animaation asetukset liikkuviin ympyröihin.....	44
Kuva 36. Valmis lopputulos animoiduista sumennetuista ympyröistä. ....	44
Kuva 37. Pysäytyspisteiden valinta.....	45
Kuva 38. Valmis lopputulos sivuston etusivun mobiiliversiosta.....	46
Ohjelmakoodi 1. Webflow'n koodieditoriin lisätty Google Mapsin upotettu koodi.....	34
Ohjelmakoodi 2. Webflow'n sisäiseen koodieditoriin tehty syntaksi. ....	36
Ohjelmakoodi 3. JavaScript-koodin lisääminen Webflow'n editoriin. ....	37

## **Liitteet**

Liite 1	Aineistonhallintasuunnitelma
---------	------------------------------

## 1 Johdanto

Webflow on visuaalinen verkkosivujen rakennustyökalu, jonka avulla voidaan luoda verkkosivut ilman varsinaista web-ohjelmointia. Verkkosivujen rakennustyökalut ovat hyvin suosittuja tänä päivänä ja Webflow on kasvattanut tasaisesti suosiotaan viime vuosina. Se on kuitenkin vähemmän tunnettu vaihtoehto, kun vertaa sitä muihin verkkosivujen rakennustyökaluihin ja sisällönhallintajärjestelmiin. (Ramoser, 2022) Webflow on perustettu vuonna 2013, joten se on vielä varsin uusi työkalu markkinoilla sen kilpailijoihin verrattuna (David, 2022).

Kiinnostus aiheen valintaan nousi omasta innostuksesta verkkosivujen tekemisestä kohtaan. Webflow'ta pidetään enemmän ammattilaisille suunnattuna työkaluna ja sen toimintaperiaate on hieman erilainen kuin muilla verkkosivujen rakennustyökaluilla, joten on mielenkiintoista nähdä, miten sivuston toteuttaminen sillä tapahtuu ja mitä mahdollisuuksia se tarjoaa verkkosivuston tekijälle.

Työn teoriaosuudessa tutustaan verkkosivujen suunnitteluprossiin ja verkkosivujen käyttöliittymän kannalta olennaisiin tekijöihin sekä käydään läpi verkkosivuston toteuttamisen eri vaihtoehtoja. Lisäksi tutkitaan miten Webflow eroaa muista verkkosivujen rakennustyökaluista ja sisällönhallintajärjestelmä WordPressistä. Teoriaosuudella haetaan taustoittavaa tietoa tutkimuksen käytännön osuutta varten.

Työn tutkimuskysymykset ovat:

- Mitä tekijöitä tulee ottaa huomioon verkkosivuston toteuttamisessa?
- Miten Webflow eroaa sen kilpailijoista?
- Mitä hyötyjä Webflow-työkalusta on verkkosivuston tekijälle?
- Minkälaisiin projekteihin Webflow-työkalu sopii?

## 2 Verkkosivuston suunnittelu ja rakenne

Tässä luvussa käydään läpi verkkosivuston suunnittelun eri vaiheita ja tekijöitä, jotka on hyvä ottaa huomioon suunnitteluprosessissa. Lisäksi luvussa tarkastellaan verkkosivuston käyttöliittymän kannalta huomioitavia asioita sekä hakukoneoptimointia.

### 2.1 Verkkosivuston suunnitteluprosessi

Dawsonin mukaan (2012, s. 10) verkkosivuston tekoprosessissa on neljä välttämätöntä vaihetta, joista yksi niistä on suunnittelu. Suunnittelu voi olla haastavaa, mutta se on olennainen osa hyvin toteutettua verkkosivustoa (Williams, ei pvm.-a, s. 23).

#### 2.1.1 Tavoitteen määrittäminen

Ensimmäinen vaihe menestyvän verkkosivuston rakentamiseen on ymmärtää mitä sivustolla halutaan saavuttaa. Mikäli tavoite ei ole selkeä, hidastaa se suunnitteluprosessia ja aiheuttaa turhautumista. (Williams, ei pvm.-a, s. 20) Tavoitteen määrittämisen merkittävä osa on kohdeyleisön tunnistaminen. Sivuston sisällön ja käytettävyyden tulisi vastata kohdeyleisön tarpeita. Esimerkiksi jos sivuston käyttäjät eivät ole erityisen diginatiiveja, ei sivustolla tulisi käyttää monimutkaisia widgettejä tai epätavallista käyttöliittymää. Esimerkiksi verkkokaupan on hyvä olla suunniteltu satunnaiselle käyttäjälle. (Tidwell ym., 2020, s. 3) Lisäksi tulisi selvittää mitä käyttäjät olettavat löytävän sivustolta ja mitä käyttäjät sivustolla tekevät. Kohdeyleisön tunteminen on läsnä koko kehitysprosessin ajan ja sivuston suunnittelun apuna voidaan käyttää yksinkertaista tukikysymystä: onko tämä hyödyllistä sivuston kohdeyleisölle? (Williams, ei pvm.-a, ss. 20–23)

Verkkosivuston päätarkoituksen tulee olla selvillä, jotta sivusto voidaan suunnitella tarkoituksen mukaisesti. Tulisi siis selvittää onko sivuston tarkoitus viihdyttää, informoida vaiko esimerkiksi myydä. (Williams, ei pvm.-a, s. 20) Tidwell (Tidwell ym., 2020, s. 7) kuitenkin mainitsee, että on petollisen helppoa hahmottaa kohdeyleisö yhtenä kasvottomana joukkona, jotka käyvät läpi yksinkertaisia käyttötappauksia mielessään yksi tehtäväkeskeinen tavoite. Hyvän suunnittelun tulisi ottaa huomioon myös kohdeyleisön odotukset, arvot, selkäydinreaktiot, mieltymykset, uskomukset ja sosiaalinen ympäristö.

Näiden tekijöistä voi joukosta voi löytyä suunnitellulle kriittinen tekijä, joka tekee sivustosta menestyvän ja vetoavan. (Tidwell ym., 2020, s. 8) On epärealistista ajatella, että sivustolla olisi jokin tyyppillinen käyttäjä tai tyyppillinen käyttöympäristö, joten on hyvä katsoa stereotyyppien ulkopuolelle (Dawson, 2012, ss. 15–16).

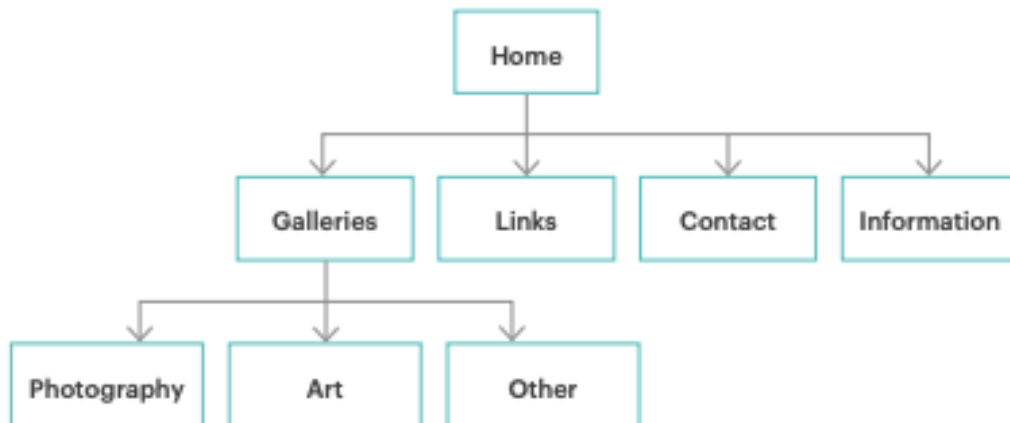
### 2.1.2 Sivustokartan luominen

Tässä osiossa sivustokartalla (engl. sitemap) tarkoitetaan diagrammia, joka kuvaa verkkosivuston hierarkian, eli sivuston rakenteen ja miten mikäkin sivu linkittyy keskenään (Caddick & Cable, 2011, s. 124). Sivustokartalla voidaan tarkoittaa myös hakukoneita varten tehtyä XML-tiedostoa. Sivustokartta toimii perustana jokaiselle hyvin suunnittelulle verkkosivulle ja se antaa suunnittelijalle selkeän kuvan sivuston informaatio arkkitehtuurista (Williams, ei pvm.-a, s. 10). Sivustokarttaa voidaan ajatella koko sivuston sinikopiona ja se toimii myös pohjana rautalankamallille (Babich, 2021).

Sivustokartan luomiseen on eri tapoja ja sopivan tavan valinta riippuu verkkosivuston tyylistä, koosta ja sivuston käyttäjistä. Sivustokartan luomisessa on tärkeää välttää lukkiutumista tiettyyn ulkoasuun, eli olisi hyvä antaa sivustokartan mukailla verkkosivuston tarkoitusta. (Rosenfeld ym., 2015, s. 420) Mikäli tehdään suurempi sisällönhallintajärjestelmäpainotteinen sivustokartta, Caddick & Cable (2011, s. 129) suosittelevat taulukkopohjaista sivustokarttaa, jotta kartta olisi helpommin helpommin tulkittavissa. Visuaalinen sivustokartta, joka koostuu viivoista ja laatikoista on sopiva pienemmille sivustoille ja visuaalisille ihmisille. Sivustokartan valinta riippuu siis myös hieman sivustokartan käyttäjistä.

Sivustokartan diagrammi voidaan tehdä joko vaaka-tai pystysuuntaisesti. Vaakasuuntaisen diagrammin (Kuva 1) tarkoituksena on näyttää korkeatasoinen kuva verkkosivustosta ja ilmaista väylät eri sivujen välillä. Pystysuuntaisessa diagrammi on käytännössä samanlainen, mutta diagrammia luetaan vasemmalta oikealle. Pystysuuntaisen diagrammin käyttö sopii parhaiten kuvaamaan jotain tarkempaa osiota kuten käyttäjävirtaa tai jotain verkkosivuston tiettyä ominaisuutta. (Williams, ei pvm.-a, s. 40)

Kuva 1. Vaakasuuntainen yksinkertainen sivustokartta (Williams, ei pvm.-a, s. 10).



### 2.1.3 Rautalankamallin luominen

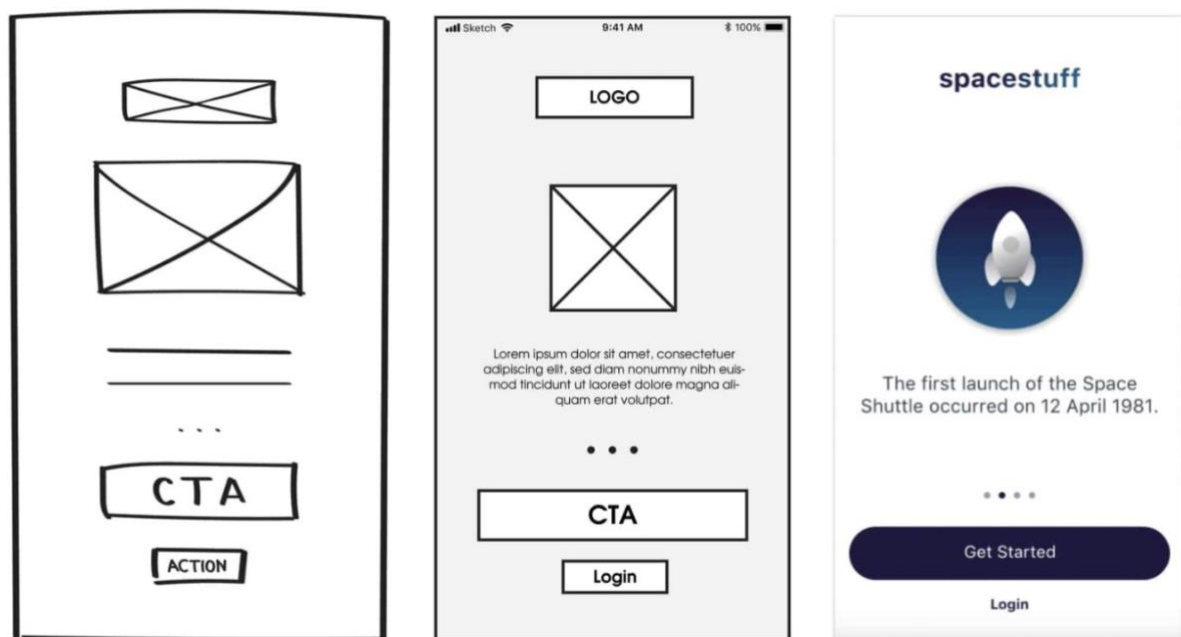
Sivustokartan luomisen jälkeen voi siirtyä rautalankamallin (engl. wireframe) tekoon.

Samaan tapaan kuin sivustokartat, myös rautalankamalli kuvaa hierarkiaa, mutta ainoastaan yhden sivun hierarkian (Williams, ei pvm.-a, s. 42) Rautalankamalli esittää sivuston käyttöliittymän ulkoasun ennen suunnittelun toteutusta. Rautalankamalla voidaan verrata talon pohjapiirustukseen ja rautalankamallin käyttö on tärkeää, jotta sivuston idea ja vaatimukset itse varsinaista suunnittelua varten ovat selkeitä. (Canziba, 2018, ss. 175–176) Rautalankamalla on helpompi muokata kuin valmista sivustoa, joten rautalankamalliin on hyvä kokeilla eri ideoita. Rautalankamalli luodaan tavanomaisesti verkkosivuston tärkeimmille sivuille ja sivuille, jotka ovat monimutkaisia tai uniikkeja. Tarkoitus ei välttämättä ole luoda rautalankamalla jokaisesta sivusta sillä yhden rautalankamallin luominen voi toimia myös mallina muille sivuille. (Rosenfeld ym., 2015, ss. 407–408) Jokaisella rautalankamallin sivulla tulee olla jokin tarkoitus sivuston käyttäjille ja mikäli sivuja suunnitellessa huomataan, että selvää tarkoitusta ei löydy, on suositeltavaa miettiä sivusta luopumista (Babich, 2021).

Rautalankamallit voidaan jakaa kolmeen eri kategoriaan niiden tarkkuuden (Kuva 2) perusteella: matalaan, keskikokoiseen ja korkeaan. Matala malli on usein tehty käyttäen kynää ja paperia ja se auttaa hahmottamaan sivuston kokonaiskuvan, joten se ei sisällä tarkkoja yksityiskohtia. Se saattaa olla hyvin yksinkertainen ja ruma, mutta helppo tehdä ja nopeasti muokattavissa. (Canziba, 2018, s. 180) Mikäli verkkosivusto tehdään asiakkaalle, on

tärkeä selventää minkä tyyppinen rautalankamalli heille esitetään (Williams, ei pvm.-a, s. 43). Malleihin on myös hyvä lisätä kommentteja esimerkiksi painikkeiden toiminnallisuuksista tai tietyistä visuaalisista elementeistä, jotta malli on paremmin ymmärrettävissä (Babich, 2021). Matalan tarkkuuden rautalankamallin etuna on, että asiakkaalta saa nopeasti palautetta. Jokaisella verkkosivuston parissa työskentelevällä henkilöllä on myös sama ajatus sivuston ulkoasusta heti ensimmäisestä rautalankamallista lähtien, joten päämäärä on helpommin hahmotettavissa. (Canziba, 2018, s. 182)

Kuva 2. Rautalankamallien eri tarkkuudet matalasta korkeaan (mentormate, n.d.).



Useimmat rautalankamallit ovat keskitason tarkkuisia (Rosenfeld ym., 2015, s. 411). Tämän tyyppin rautalankamalliin käytetään tyypillisesti jotain suunnittelutyökalua. Keskitason tarkkuudessa mallissa sivuston rakenne näkyy jo selvästi, yksityiskohtia on enemmän ja ulkoasu on parempi. Korkean tarkkuuden malli saattaa näyttää jo lähes lopulliselta versioltta verkkosivustosta ja siinä käytetään oikeita kuvia, fontteja ja värejä, kun taas keskitasoisessa näitä ei vielä käytetä. Jos vertaa eri tarkkuuksia keskenään, niin korkea ja keskitasoinen ovat lähempänä toisiaan, kuin matala ja keskitasoinen. (Canziba, 2018, ss. 184–185) Korkean tarkkuuden mallia ei tulisi tehdä ennen kuin tiedetään miltä lopullinen muotoilu tulee näyttämään ja se tehdään usein vasta kun muotoilua on kokeiltu matalammilla tarkkuuksilla (Osman, 2021). Korkean rautalankamallin käyttöä voi suositella, mikäli resurssit riittävät,

testataan monimutkaisempaa käyttöliittymää tai halutaan tehdä vaikutus ulkopuoliseen tahoon (Babich, 2021).

## 2.2 Käyttöliittymä

Tässä kappaleessa esitellään verkkosivuston käyttöliittymän kannalta olennaisia tekijöitä ja mitä niiden käytössä kannattaa ottaa huomioon.

### 2.2.1 Verkkosivujen typografia

Typografia on prosessi ja taiteenmuoto, jonka tarkoitus on saada kieli näkyvään muotoon (Cullen, 2012, s. 12). Typografia sisältää tekstin ja kirjainten järjestämistä ja muotoilua ja pienilläkin muutoksilla muotoiluun voi olla suuri vaikutus kommunikaatioon (Leaning, 2017). Yksi typografian tärkeimmistä merkityksistä on saada teksti luettavaksi, sillä teksti on keskeinen osa jokaista käyttöliittymää. Huono typografia käännäyttää vierailijat sivustolta pois, joten verkkosivuston typografiaan kannattaa keskittyä. Typografia on erittäin laaja alue, mutta siitä hyvä kuitenkin tietää perusteita web-suunnittelussa onnistumiseksi. (Babich, 2017; Hogan, 2009, s. 63)

Web-suunnittelijan näkökulmasta fontit voidaan jakaa kolmeen eri kategoriaan; päätteellisiin (serif), päätteettömiin (sans-serif) ja tasalevyisiin (monospace) (Hogan, 2009, ss. 62–63). Pääosin verkkosivulla käytetään päätteellisiä ja päätteettömiä fontteja, joista varsinkin päätteettömät ovat olleet jo pidempään suosittuja verkkosivuilla (Williams, ei pvm.-b, s. 55). **Päätteellisten** fonttien kirjaimien päistä löytyy pienet päätteet ja päätteettömien fonttien kirjaimista nämä puuttuvat. Päätteellisiä fontteja löytyy suuri määrä, joista yksi kuuluisimmista on Times New Roman. (Beird & George, 2014, s. 139) Serif-fontteja käytetään enemmän tulostettuun materiaaliin sekä esimerkiksi otsikoihin tai logoihin. Ne sopivat kontrastin luomiseksi ja niillä voi kiinnittää hyvin huomiota (Juviler, 2022). **Päätteettömät** eli sans-serif-fontit sopivat paremmin näytöllä luettavaksi, joten niiden käyttöä suositellaan verkkosivujen päätetekstiksi (Hogan, 2009, s. 64). Ne ovat hieman luettavampia ja käytännöllisempiä, kuin serif-fontit (Beird & George, 2014, s. 142).

**Tasalevyisissä** eli monospace-fonteissa kaikki kirjasimet ovat yhtä leveitä ja sitä käytetään esimerkiksi koodieditoreissa ja komentoliittymissä (McKay, 2021).

Verkkosivuston fonttien määrä tulisi pitää minimissä, sillä liian monen fontin käyttäminen tekee sivustosta sekavan ja epäammattimaisen. Fonttien määrän olisi hyvä olla yhden ja kolmen fontin välillä. Mikäli käytetään useampaa kuin yhtä fonttia, olisi suositeltavaa käyttää samantyyllisiä fontteja. Vaikka fontteja on paljon tarjolla, kannattaa kuitenkin yleisesti käyttää standardifontteja. Riippuu aina myös hieman kontekstista, joskus erikoisen fontin käyttö voi olla toimiva ratkaisu. (Babich, 2017)

Fonttien valinnassa tulee myös huomioon fonttien toimivuus verkkosivuilla. On olemassa niin sanottuja "web-safe"-fontteja, joilla tarkoitetaan laitteella valmiiksi asennettuja fontteja, mutta näiden fonttien määrä ei ole kovin suuri. On kuitenkin tarjolla myös muita keinoja fonttien käyttöön, jolloin fontteja saa laajemmin käyttöön. (Beaird & George, 2014, s. 125) CSS-tyylikielille on tarjolla @font-face ominaisuus, joka lataa fontin verkkosivulle. Eli kun @font-face lisätään CSS-tiedostolle, se kertoo selaimelle mistä palvelimelta fontti löytyy. Vaikka ominaisuuskin olisikin käytössä, on kuitenkin hyvä olla myös web-safe fontteja varalle, jotka latautuvat tarvittaessa. (Borowska, 2013) Fontit voidaan isännöidä joko omalta palvelimelta tai käyttää yksinkertaisempaa ja suositumpaa tapaa, eli fonttipalvelua kuten Google Fonts tai Typekit -palveluita, joiden palvelimelta fontit löytyvät (Beaird & George, 2014, ss. 128–129).

Lisäksi tulee ottaa huomioon eri päätelaitteet, eli tekstin pitäisi olla luettavissa myös esimerkiksi mobiililaitteilla. Tätä varten tulee valita fontit, jotka toimivat useassa eri koossa. Kursivoidut fontit ovat usein huono valinta, sillä niitä on vaikea lukea pienemmällä näytöllä. Usein fonteissa saattaa olla myös samanlaisia kirjaimia, kuten I- ja L-kirjaimet, joten tämä on hyvä ottaa tietyissä tilanteissa huomioon. (Babich, 2017)

**Fonttien koon** määrittämiseksi käytetään usein relatiivisia yksiköitä, mutta fontin koon voi asettaa muillakin keinoin kuten esimerkiksi vw, vmax tai vmin -yksiköillä. Kunkin yksikön käyttö perustuu kuitenkin usein kehittäjän henkilökohtaisiin mieltymyksiin. (Hilal, 2020; Suzanne, 2022)

Jos puhutaan fontin koosta pikseleinä, niin kaiken verkkosivustolta löytyvän tekstin olisi suositeltavaa olla minimissään 16 pikseliä eli 1 em tai rem-yksikköä. Fontin kokoon liittyy myös hierarkia, eli otsikoiden koon tulisi aina olla muuta tekstiä isommat hierarkian saavuttamiseksi (Juviler, 2022). Hierarkia helpottaa verkkosivuilla navigoimista ja tekstien lukemista. Yhtä lailla, myös riviväli ja rivien pituus vaikuttavat tekstin luettavuuteen. Hyvän luettavuuden takaamiseksi rivivälin olisi hyvä olla noin 30 % enemmän, kuin kirjaimien korkeus. Rivien pituuksien tulisi olla enintään 60 merkkiä ja mobiililaitteilla 30–40 merkkiä. (Babich, 2017)

### **2.2.2 Värit verkkosivuilla**

Väreillä on suuri merkitys verkkosivujen suunnittelussa. Värien avulla voi luoda tunnelmaa ja herättää tunteita tai esimerkiksi ohjata verkkosivuston käyttäjää tiettyyn toimintaan. Jokaisella värillä on oma tarkoituksensa ja ihmiset reagoivat väreihin eri tavalla. Väri on yksi ensimmäisistä asioista joihin verkkosivuston käyttäjä kiinnittää huomiota, joten oikeiden värien käyttämiseen tulee kiinnittää huomiota. (Canziba, 2018, ss. 141–142) Väärän värin valinta voi aiheuttaa käyttäjässä hämmennystä tai epämieluisan reaktion (Hogan, 2009, s. 38).

**Väripsykologialla** tarkoitetaan värien herättämiä tuntemuksia. Kun tiedetään mitä tunteita värit edustavat (Kuva 3), voidaan verkkosivuston värejä valita tiettyjen tunteiden herättämiseksi. (Gillis, 2021)

Kuva 3. Värien herättämät tuntemukset (usertesting.com, 2019).



Väripsykologian tunteminen auttaa tekemään parempia päätöksiä suunnittelun suhteen ja sivuston värimaailma voidaan valita yrityksen teemaan sopivaksi (Gillis, 2021). Tulee kuitenkin huomioida, että värien herättämiin yleisiin tuntemuksiin ei pidä luottaa liikaa, mutta joidenkin värien herättämät yleiset tuntemukset on kuitenkin hyvä tiedostaa (Beaird & George, 2014, s. 54). Värejä tulkitaan myös eri kulttuureissa eri tavalla, joten se tulisi ottaa huomioon, mikäli suunnitellaan kansainvälistä sivustoa (Hogan, 2009, s. 40).

Värien valitsemisessa on tärkeää pitää mielessä, että tekstin tulisi olla aina helposti luettavissa. Tiettyjen väriyhdistelmien käyttöä tekstin kanssa tulisi välttää ja tätä varten voi hyödyntää **väriympyrää** (Kuva 4) (Canziba, 2018, s. 148). Esimerkiksi oranssin tekstin laittaminen siniselle taustalle ei ole suositeltavaa, sille oranssi ja sininen ovat toistensa vastavärit, eli väriympyrän vastakkaisilla puolilla. Tällaisen tekstin lukeminen vaatii silmiltä pinnistelyä. (Tidwell ym., 2020, ss. 258–259) Vastavärien yhdistäminen muissa tilanteissa voi kuitenkin olla usein suositeltavaa, sillä ne aiheuttavat kontrastivaikutuksen, eli kumpikin väri korostaa toistaan. (Beaird & George, 2014, s. 75) Kontrastilla on tarkoitus vangita käyttäjän huomio tai tehdä jokin elementti erottuvammaksi (Canziba, 2018, s. 164). Mitä suurempi kontrasti, sitä helpommin elementti erottuu joukosta (Beaird & George, 2014, s. 26). Jos kontrasti taas on liian alhainen, voi elementtien erottaminen toisistaan olla hankalaa (Khazanova, 2021).

Kuva 4. Väriympyrä (Tidwell ym., 2020, s. 259).



Väriympyrä auttaa näkemään suhteet eri värien välillä ja yhdistämään värejä, joten se on hyödyllinen työkalu verkkosivun värimaailman valinnassa (Hogan, 2009, s. 41). Vastavärien lisäksi väriympyrän avulla voidaan muodostaa myös muita väripaletteja, joita voi käyttää apuna verkkosivuston värien suunnitteluun. Väreillä kokeilu voi joskus olla myös kannattavaa, mutta useimmiten käytetään väriympyrästä muodostettuja väripaletteja (Cartwright, 2021). Muita väripaletteja ovat esimerkiksi monokromaattinen väripaletti, analoginen väripaletti (Kuva 5), täydentävä väripaletti ja kolmikkoväripaletti (Beaird & George, 2014, s. 65). Tarkkaan mietitty väripaletti tekee sivuston suunnittelusta tasapainoisen ja hyvin toimivat värit luovat myös mieluisan käyttäjäkokemuksen (Khazanova, 2021).

Kuva 5. Analoginen väripaletti (Cartwright, 2021).



Värien käyttämisessä verkossa on myös tärkeä tietää, että verkossa käytetään RGB- tai HSL-värimalleja. Väriä voi ilmoittaa myös Hex-arvona, eli heksadesimaalisena arvona, joka RGB-värimallia. Esimerkiksi valkoinen väri on RGB-arvona `rgb(255, 255, 255)`, heksadesimaalisena arvona `#ffffff` ja HSL-arvona `hsl(0,0%,100%)`. (Frain, 2012, s. 20; Williams, ei pvm.-b, ss. 35–36)

### 2.2.3 Kuvat

Värit olisi hyvä olla mietitty valmiiksi ennen sivuston tyylin luomista, sillä ne vaikuttavat kuvien ja kuvituksien valintaan. Käyttäjät haluavat nähdä sivustolla kuvia, sillä kuvat tekevät sivustosta järjestelmällisemmän ja käytettävämmän. (Williams, ei pvm.-a, s. 15-16)

Ennen kuin valitsee verkkosivustolle sopivia kuvia tai kuvituksia, tulisi pohtia ovatko ne oleellisia sivuston sisällön ja persoonallisuuden kannalta. Sopivien kuvien valinta auttaa sivuston sisältöä tulemaan paremmin esiin ja auttaa käyttäjiä muistamaan sisällön paremmin. Kuvilla on tärkeä rooli viestin välittämisessä. Sisällön ja kuvien välillä olisi siis hyvä olla yhteys, mutta se ei saisi olla tärkein asia, sillä muuten sivustosta saattaa tulla liian kliseinen. Joskus voi olla parempi rikkoa yleistä käytäntöä, mutta kuvan tulisi kuitenkin olla siinä tapauksessa mielenkiintoinen ja vetoava. (Beaird & George, 2014, ss. 160–162) Sivuston kohdeyleisö ja teema kannattaa pitää mielessä kuvien valinnassa, sillä ihmisten mieltymykset ovat erilaisia. Kuvien on tarkoitus olla esteettisesti ja henkisesti houkuttelevia, kun taas vääränlaiset kuvat voivat tehdä sivustosta tai brändistä luotaantyöntävän. (Beaird & George, 2014, s. 161)

**Kuvien formaatit** voidaan jakaa kahteen eri kategoriaan: rasteri- ja vektorigrafiikkaan.

Rasterigrafiikkaan kuuluvat formaatit PNG, JPEG, GIF sekä WebP, joista JPEG ja PNG ovat verkkosivustoilla käytetyimmät muodot. Rasterigrafiikka muodostuu pikseleistä, kun taas vektorigrafiikka muodostuu viivoista, pisteistä ja monikulmioista. Poiketen rasterigrafiikasta, pysyy vektorigrafiikan laatu aina samana siihen suurentamalla. Verkossa käytetään vektorigrafiikan formaattia SVG, joka tulee sanoista Scalable Vector Graphics. Sen käyttäminen sopii esimerkiksi logoihin, taulukoihin tai kuvakkeisiin. Rasterigrafiikan käyttö sopii yksityiskohtaisiin kuviin kuten valokuviin. Tulee kuitenkin huomioida, että jokaisella formaatilla on omat käyttötarkoituksensa ja ominaisuudet sekä on myös oleellista optimoida ne tarvittaessa kompressoimalla. (Bermes, 2015, ss. 138–140; Robbins, 2012, s. 533)

#### **2.2.4 Responsiivisuus**

Responsiivinen web-suunnittelu on menetelmä, jonka tarkoitus on saada verkkosivusto käytettäväksi ja hyvin näkyväksi jokaisella päätelaitteella ja resoluutiolla, eli esimerkiksi puhelimilla, tableteilla tai tietokoneella. Mobiililaitteiden käyttö kasvaa jatkuvasti, joten responsiivisuus on tärkeä osa web-suunnittelua, jotta käyttäjät eivät lähde sivulta, joka ei ole responsiivinen. (Williams, ei pvm.-a, s. 16) Responsiivinen web-suunnittelu tarkoittaa, että verkkosivustosta ei luoda esimerkiksi erillistä mobiiliversiota, vaan suunnitellaan ulkoasu, joka muokataan eri ruuduille sopivaksi CSS-tyylikielen avulla. Tietyissä tilanteissa pelkkää responsiivista menetelmää ei siis kannata käyttää, jos halutaan verkkosivuista poikkeava käyttökokemus mobiililaitteilla. (Beaird & George, 2014, s. 42; Frain, 2012, s. xxvii)

Marcotten (2014, s. 11) mukaan responsiivinen käyttöliittymä koostuu kolmesta pääosasta, jotka ovat mukautuva ruudukkopohjainen pohja (engl. grid-based layout), mukautuvat kuvat sekä CSS-mediakyselyt (engl. media queries). Responsiivisessa suunnittelussa käytetään myös relatiivisia mittayksikkö kuten prosentteja tai rem- ja em-yksikköjä, jolloin sivuston elementit kuten fontit ja kuvat skaalautuvat näytön koon mukaan.

### 2.3 Hakukoneoptimointi

Hakukoneoptimoinnilla tarkoitetaan sivuston sisällön luomista ja kehitystä hakukoneita ajatellen. Verkkosivustolle luodun sisällön tulisi olla optimoitu hakukoneille, sillä sisältö toimii ikään kuin houkuttimena hakuroboteille. Jotta hakukoneet löytävät verkkosivuston helpoiten, sisällön tulisi olla laadukasta ja sen tulisi sisältää oleellisia hakusanoja ja lauseita. Sivustolle sopivia hakusanoja ja lauseita kannattaa siis tutkia, jotta niitä voi hyödyntää sivuston sisällössä. Tulee kuitenkin pitää mielessä, että sisältö ei kuitenkaan saa olla täynnä hakukoneille oleellisia hakusanoja tai lauseita, jotta sisällöstä ei tule hämmentävää.

(Williams, ei pvm.-a, s. 13, 55, 88)

### 3 Verkkosivuston toteutustavat

Verkkosivuston toteuttamiseen on karkeasti jaoteltuna kolme vaihtoehtoa. Sivuston voi luoda kokonaan itse ohjelmoimalla, käyttää sisällönhallintajärjestelmää tai verkkosivujen rakennustyökalua. Tässä luvussa käydään tarkemmin lävitse kutakin vaihtoehtoa.

#### 3.1 Verkkosivujen rakennustyökalut

Verkkosivujen rakennustyökaluilla (engl. website builder) tarkoitetaan verkkoalustaa, jonka avulla voi luoda verkkosivuston ilman ohjelmoinnin osaamista. Useat työkalut tarjoavat valmiita ulkoasumalleja (engl. template) sivuston rakentamiseen. Työkalut toimivat usein raahaa ja pudota -tekniikalla, jonka avulla voi valita sivustolle eri elementtejä ja järjestellä niitä. (Hernandez, 2022) Sivuston ulkoasun näkee siis kokoajan samalla, kun elementtejä lisää tai järjestelee (Benjamins, 2022a). Rakennustyökalujen käyttäminen sopii siis etenkin aloittelijoille, jotka haluavat luoda nopeasti ensimmäisen verkkosivuston, sillä luomisen helppouden lisäksi esimerkiksi verkkotunnuksen rekisteröinti, sivuston isännöinti ja hakukoneoptimointi onnistuu yhtä alustaa käyttämällä (Hernandez, 2022).

Verkkosivujen rakennustyökalut toimivat tilauspohjaisella maksulla, eli käyttäjät maksavat työkalusta valitsemansa tilauksen mukaisesti. Maksusta puhutaan suunnitelmana (engl. plan), joita on tarjolla eri tarkoituksiin. Kuukausimaksut vaihtelevat usein 8–40 euron välillä, mutta ilmainenkin jäsenyys voi olla tarjolla, joskin se on hyvin paljon riisutumpi versio maksullisista versioista. Yleensä on myös tarjolla Enterprise-malleja räätälöityimpiin tarpeisiin, jotka voivat olla kalliimpia tai toimivat tarjouspyyntö -hinnoittelulla. (Carney, 2022)

Rakennustyökalun valinnassa kannattaa ottaa huomioon käsite toimittajaloukku (engl. vendor lock-in), eli kuinka helposti sivustolle ladatun sisällön ja datan saa tarvittaessa pois, jos haluaa esimerkiksi vaihtaa palvelua. Verkkosivujen rakennustyökalujen käyttö voi tuntua myös rajoitetulta, sillä tyypillisesti verkkosivuston kustomointia ei voi tehdä yhtä laajasti, kuin jos tekisi sivuston itse ohjelmoimalla. (Hernandez, 2022) On kuitenkin huomioitava, että työkaluja on suuri määrä, joten eroavaisuuksia löytyy työkalujen välillä.

Suosituimpia verkkosivujen rakennustyökaluja ovat Wix ja Squarespace sekä verkkokaupoille suunnattu Shopify. W3techs.com -sivuston tekemän tutkimuksen mukaan Wix on käytössä 2,3 % kaikista verkkosivustoista, Squarespace 2,0 % ja tämän tutkimuksen aiheena oleva Webflow on käytössä 0,6 % kaikista verkkosivustoista. Webflow'n prosenttimäärä on kasvanut noin 0,2 % marraskuusta 2021 lähtien. Verraten sen kilpailijoihin, sitä käytetään myös enemmän korkean kävijämäärän omaavilla sivustoilla. (w3techs, 2022)

### **3.2 Sisällönhallintajärjestelmät**

Verkkosivujen rakennustyökaluja ja sisällönhallintajärjestelmiä pidetään usein samana, joten niiden yleisimmät eroavaisuudet on hyvä hahmottaa (Pawlak, 2021).

Sisällönhallintajärjestelmä (engl. content management system) on järjestelmä, jolla voi luoda verkkosivuston rakenteen ja hallita verkkosivuston sisältöä kuten sivuja, kuvia, dataa tai esimerkiksi käyttäjiä. Olennaista sisällönhallintajärjestelmissä on, että sisältö, sivuston suunnittelu ja ohjelmointi ovat keskenään erotettavissa. Sisällön luominen ja muokkaaminen onnistuu siis ilman web-tekniikoiden osaamista, ja jaottelu sivuston tekijän ja sisällön luojaan välillä on varsin yleinen käytäntö. (Hauschildt, 2010, ss. 30–31)

Perinteinen sisällönhallintajärjestelmä koostuu kahdesta kerroksesta eli CMA:sta (Content Management Application) ja CDA:sta (Content Delivery Application). CMA on yksinkertaisesti sisällönhallintajärjestelmän frontend, eli se antaa ylläpitäjien muokata sivuston sisältöä ja selainpuolen ulkoasua ilman web-kehittäjän apua. CDA on sisällönhallintajärjestelmän backend, joka yhdistää ja säilyttää kaiken sisällön sekä päivittää itse verkkosivuston. (Heslop, 2018) Perinteisten sisällönhallintajärjestelmien lisäksi löytyy Headless CMS -toteutustapa, joka tarjoaa enemmän joustavuutta, sillä se erottelee sisällön (backend) ja sisällön esitystavan (frontend), eli kehittäjät voivat hyödyntää sisältöä Headless-mallin tarjoaman rajapinnan avulla jokaiseen sovellukseen tai laitteeseen, sekä käyttää haluamiansa web-tekniologioita ulkoasukerroksen kehittämiseen. (Heslop, 2018; O'Connell, 2022). Headless CMS on käytössä enemmän isommilla yrityksillä, joilla on tarve käyttää sisältöä rajapinnan kautta esimerkiksi älylaitteissa. Perinteinen sisällönhallintajärjestelmä on kuitenkin yleisempi, helpompi ja kustannustehokkaampi ratkaisu varsinkin pienemmille sekä keskisuurille yrityksille. (Nitro, 2022)

Suosituin sisällönhallintajärjestelmä on ylivoimaisesti WordPress, jonka markkinaosuus on yli 43 % kaikista verkkosivustoista (w3techs, 2022). WordPressistä puhuttaessa tulee huomioida, että on olemassa WordPress.com ja WordPress.org -sivustot. Yleisesti ottaen tarkoitetaan .org -sivustoa, kun puhutaan WordPressistä sisällönhallintajärjestelmänä. WordPress on avoimen lähdekoodin ohjelmisto, jonka käyttäjä lataa tietokoneelle ja sivuston julkaisemiseksi lataa sivuston palvelimelle. WordPressin käyttö ja sen lataaminen on ilmaista, mutta verkkotunnuksen ja hosting-palvelusta pitää maksaa erikseen. WordPress.com tarjoaa esimerkiksi verkkosivujen rakennustyökalun ja hosting-palveluita. (Thibodeau, 2022) Muita suosittuja perinteisiä sisällönhallintajärjestelmiä ovat esimerkiksi Drupal ja Joomla, mutta niiden markkinaosuus jää kuitenkin huomattavasti alle WordPressin (w3techs, 2022).

### 3.3 Verkkosivuston ohjelmointi

Verkkosivustojen yleinen rakenne koostuu kolmesta komponentista, jotka ovat sisältö, esitystapa ja käyttäytyminen. Sisältö muodostuu HTML-merkintäkielestä, esitystapa CSS-tyyliekielestä ja sivuston käyttäytyminen Javascript-ohjelmointikielestä. (Dean, 2018, s. xiv) Kun verkkosivustoa luodaan, olisi kunkin kielen tiedostot hyvä olla eroteltuna (Eygi, 2019). Verkkosivuston julkaisemiseksi vaaditaan kyseisten kielen osaamisen lisäksi tekstieditori, verkkoselain, verkkosivustojen isännöintipalvelu sekä verkkotunnus. (Dean, 2018, s. xiv,3; Fitzgerald, 2022).

**HTML** tulee sanoista Hypertext Markup Language ja lyhykäisyydessään se kertoo selaimelle mitä verkkosivustolla näytetään (Robbins, 2012, s. 8, 207) Verkkosivustojen sisältö koostuu eri HTML-elementeistä, kuten otsikoista ja kappaleista, jotka muodostetaan HTML-tunnisteiden avulla. Elementti koostuu useimmiten (Kuva 6) alkutunnisteesta, sisällöstä ja lopputunnisteesta. (Dean, 2018, s. 7)

Kuva 6. Esimerkki pääotsikon HTML-elementistä ja HTML-tunnisteista.

```
<h1> Tämä on HTML-elementti </h1>
```

**CSS**, joka on lyhennetty englannin kielen sanoista Cascading Style Sheets, määrittää verkkosivuston ulkoasun, eli miltä HTML:llä luotu sisältö näyttää. CSS toimii muodostamalla sääntöjä HTML-elementtien ulkoasuun. Selain lukee säännöt dokumentilta ja määrittää sivuston ulkoasun niiden mukaisesti. CSS-säännöt koostuvat valitsimista, eli nimensä mukaisesti niillä valitaan HTML-elementit, joita halutaan muotoilla. Valitsin on aina ensimmäinen osa CSS-säännön muodostamisesta. Valitsimen hakasulkujen (Kuva 7) sisälle kirjoitetaan deklaraatio, eli mitä elementtiä halutaan muokata, ja ominaisuus, eli esimerkiksi valitun elementin väri. (Robbins, 2012, ss. 8, 210–211)

Kuva 7. Esimerkki CSS-säännöstä ja luokkavalitsimesta (web.dev, n.d.).

```

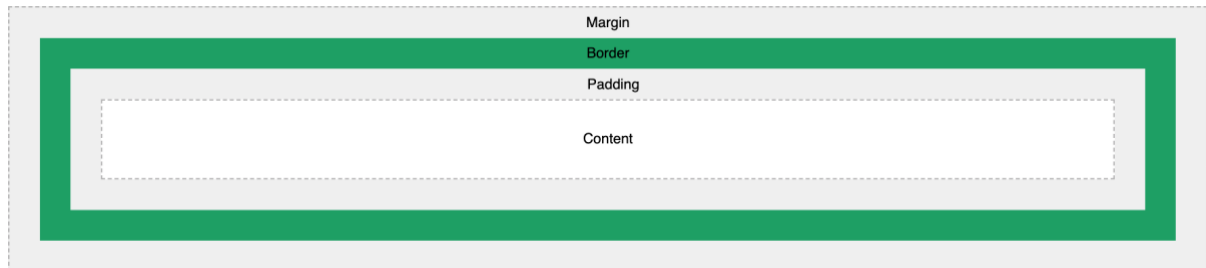
Selector
↓
.my-css-rule {
  background: red;
  color: beige;
  font-size: 1.2rem;
}
Property      Value
  ↑           ↑
  
```

Valitsimia löytyy muutamia erilaisia, mutta yksi suosittu tapa valitsimien tekoon on luokkavalitsimet (engl. class selector), joita käyttämällä HTML-elementti voidaan määrätä käyttämään tiettyä tyyliä ja samaa luokkavalitsinta voidaan käyttää useampaan elementtiin. (Dean, 2018, s. 79) Webflow'n toimintaperiaate perustuu myös luokkavalitsimen käyttöön.

Laatikkomallin (engl. box model) toimintaperiaatteen ymmärtäminen on keskeinen osa CSS-tyylikieltä. CSS:ää käytettäessä on olennaista muistaa, että kaikkia HTML-elementtejä voidaan ajatella laatikoina ja laatikkomalli on laatikko, joka rakentuu HTML-elementin ympärille. Laatikkomalli (Kuva 8) rakentuu HTML-elementin sisällöstä, täytteestä,

reunuksesta ja marginaalista. Laatikkomallin avulla voidaan esimerkiksi määrittää tilaa elementtien välille tai lisätä reunuksen elementin ympärille. (w3schools, n.d.; *Web.dev*, n.d.)

Kuva 8. CSS:n laatikkomalli (w3schools, n.d.).



**JavaScript** on ohjelmointikieli, jonka avulla voi luoda verkkosivustolle interaktiivisuutta ja funktionalisuutta. Esimerkiksi valikon laajentamiseen tai lomakkeen validointiin voi hyödyntää JavaScriptiä.

## 4 Webflow

Tässä luvussa esitellään Webflow'n ominaisuuksia ja tehdään vertailua muihin verkkosivujen rakennustyökaluihin. WordPressin ja Webflow'n eroja käsitellään laajemmin alaluvussa.

### 4.1 Webflow'n esittely

Webflow on selaimella käytettävä pilvipohjainen alusta verkkosivustojen luomiseen. Webflow kuvailee työkaluaan visuaalisena pohjana (engl. canvas), jolla voi rakentaa verkkosivuja HTML:n, CSS:n ja JavaScriptin avulla. Kun suunnittelupohjalle lisätään eri elementtejä, luo Webflow puhdasta, W3C-standardia noudattavaa koodia, jonka voi halutessaan ladata itselleen ja esimerkiksi siirtää erilliseen isännöintipalveluun. Koodin lataaminen on kuitenkin mahdollista ainoastaan tietyissä maksullisissa tilauksissa. Useat verkkosivujen rakennustyökalut tuottavat sotkuista koodia, mikä vaikuttaa sivustojen nopeuteen ja hakukoneiden näkyvyyteen. Hakukoneiden indeksointirobotit tykkäävät puhtaasta koodista, jota Webflow tuottaa, joka mahdollistaa paremman näkyvyyden hakukoneissa. (Webflow, n.d.; McSweeney, 2022)

Webflow'ta voi käyttää ilmaiseksi rajatuin ominaisuuksin tai maksullisia versioita, jolloin ominaisuudet lisääntyvät hinnan kasvaessa. Maksullisten versioiden hinnat vaihtelevat 14 dollarin kuukausimaksusta räätälöityyn tarjoukseen asti, joka on suunnattu suuremmille yrityksille. (Webflow, n.d.)

Kuten muutkin verkkosivujen rakennustyökalut, ei Webflow'n käyttöön vaadita ohjelmointitaitoja, mutta ohjelmointia osaava henkilö voi tarvittaessa tehdä kustomointeja koodin avulla. Verrattuna muihin työkaluihin, Webflow tarjoaa kaikista eniten muokattavuutta sen tarjoaman mukautetun koodin lisäämisen mahdollisuuden vuoksi. Wixin kehittämä Editor X -rakennustyökalu sisältää myös integroidun kehitysympäristön, mutta Webflow'n kehitysympäristö sopii kuitenkin hieman paremmin vaativampaan ohjelmointiin ja esimerkiksi CSS-tyyliin uudelleen käyttäminen on helpompaa Webflow'ssa. Lisäksi Editor X -työkalussa ei ole erillistä näkymää sivuston sisällön muokkaukseen. Editor X soveltuu paremmin verkkokaupoille halvemman hinnan vuoksi sekä tiimikäyttöön, sillä sivustoa voi muokata useampi henkilö samanaikaisesti. (Benjamins, 2022b; Graw, 2022) Walsh (2022) ei

myöskään suosittelen Webflow'n valitsemista verkkokaupoille, sillä verkkokaupoille on tarjolla parempia verkkokaupoille suunnattuja vaihtoehtoja.

Webflow on ainoa verkkosivujen rakennustyökalu, joka tarjoaa täyden sisällönhallintajärjestelmän, josta on hyötyä esimerkiksi suurille blogisivustoille tai sivustoille, jotka sisältävät paljon dataa. Sisällönhallintajärjestelmän tietokannan kapasiteetti on kuitenkin rajallinen, joten ennakkoon on hyvä miettiä, onko se riittävä. (Graw, 2022; Walsh, 2022) Poikkeuksena on Enterprise-tilaus, josta voi pyytää erikseen tarjousta. (Webflow, n.d.)

Webflow saattaa olla haastava käyttää aloittelijalle, joten web-suunnittelun ja -tekniikoiden kuten HTML:n ja CSS:n perusteiden ymmärtäminen voi olla suositeltavaa sen käyttämiseksi (Benjamins, 2022b). Varsinkin, jos CSS-tyylikielen laatikkomalli ei ole entuudestaan tuttu, vaatii Webflow'n oppiminen aikaa. Webflow ei toimi täysin samalla "raahaa ja pudota"-tekniikalla, kuten useat muut työkalut, sillä elementtejä ei voi suoraan pudottaa sivustolla mihin haluaa, vaan niiden sijainti on säädettävä erikseen. (Walsh, 2022) Walsh sanookin, että aloittelijan ei kannata valita Webflow'ta, jos haluaa luoda verkkosivuston mahdollisimman pienellä vaivannäöllä. Webflow'n tarjoaa kumminkin erinomaiset koulutus- ja tukimateriaalit, joiden avulla Webflow'n käytön oppii kohtuullisen lyhyessäkin ajassa. Oppimisen vaikeuden lisäksi Walsh listaa Webflow'n huonoiksi puoliksi myös muun muassa hämmentävän hintarakenteen, puhelinasiakaspalvelun puuttumisen sekä sen, että sivustoa ei voi muokata mobiilisovelluksella. Webflow ei kuitenkaan yritä piilottaa web-kehityksen monimutkaisuutta, vaan ennemminkin käyttää sitä myyntivalttina. Johnson (n.d.) mainitsee Webflow'n verkkosivuilla, että Webflow soveltuu erityisesti frontend-kehittäjille ja listaa muutamia eri syitä miksi kehittäjän kannattaisi valita Webflow:

- Rakennettu ammattilaisia ajatellen
- Tyyli luokat CSS-tyylikielen mukaisesti
- Täysi hallinta sivuston ulkoasusta modernien työkalujen avulla kuten CSS-gridin
- Sivuston julkaisu tapahtuu parilla klikkauksella
- Joustava ja tehokas sisällönhallintajärjestelmä

- Työkalu animaatioiden ja interaktiivisten elementtien tekoon

## 4.2 Webflow'n ja WordPressin eroavaisuuksia

Yksi merkittävä tekninen ero on, että WordPress on avoimen lähdekoodin alusta, joten esimerkiksi sivustojen teemojen koodia pääsee muokkaamaan, mutta Webflow'ssa koodia ei itse pääse muokkaamaan, vaan ainoastaan lisäämään. WordPressissä sivuston isännöinti ja verkkotunnus ostetaan erikseen, kun taas Webflow'ssa ne kuuluvat palveluun. WordPressin vapaus valita isännöintipalvelun suhteen voi tarjota kuitenkin parempia vaihtoehtoja. (Paterson, 2022; Steele, 2022)

WordPress perustettiin vuonna 2003 blogien julkaisemispalveluksi, minkä johdosta se on tänä päivänä yksi parhaista työkaluista blogin tekemiseen ja tarjoaakin enemmän ominaisuuksia kuin Webflow'n CMS-työkalu. Webflow'n CMS-työkalua kuitenkin kehitetään jatkuvasti. Tähän liittyen, WordPressin käyttäjien määrää ei ole rajattu samalla tapaa kuin Webflow'ssa. Esimerkiksi Webflow'n CMS-tilaus sallii kolmen käyttäjän lisäämisen, jotka voivat muokata sivuston sisältöä ja kallein tilaus sallii 10 käyttäjää. Itse sivuston sisällön muokkaaminen onnistuu taas Webflow'n puolella hieman WordPressiä helpommin. (Paterson, 2022)

WordPressiä käyttäessä pitää itse huolehtia sivuston tietoturvasta pitämällä WordPressin ja lisäosien päivitykset ajan tasalla, kun taas Webflow hoitaa tietoturvan käyttäjän puolesta. Varsinkin kolmannen osapuolen lisäosat saattavat olla haavoittuvaisia hakkeroinnille, joten niiden päivityksien suhteen pitää olla tarkkana. Webflow tarjoaa SSL-suojauksen kaikille sivustoille, ja WordPressissä SSL-suojaus pitää hankkia erikseen. Webflow'n tietoturvaa tarkastetaan myös kolmannen osapuolen puolelta, eli heiltä löytyy tyyppin I ja II SOC 2-sertifikaatit. Webflow sivustot isännöidään Amazonin (Amazon Web Services) palvelimella, joten riski sivuston kaatumiseen on pieni. Lisäksi sivustot toimivat nopeasti Fastlyn ja Amazon Cloudfront sisällönjakeluverkkojen avulla. (Paterson, 2022; Webflow, n.d.) Yksi isoimmista eroista on siis se, että Webflow on käyttövalmiimpi heti alusta alkaen, toisin kuin WordPress, jossa sivustoja voidaan parantaa lisäosien avulla. Tällöin käyttäjällä on kumminkin laajemmin valinnanvaraa, mutta vaatii enemmän eri lisäosiin perehtymistä. (Newcomer, 2022). WordPress on suunniteltu lisäosien käyttämistä varten, joten lisäosia

löytyy luonnollisesti runsaasti. Lisäosien asentaminen onnistuu myös suoraan WordPressin käyttöliittymästä, verraten Webflow'hun, jossa integraatioiden käyttäminen ei onnistu suoraan Webflow'sta. (Paterson, 2022)

Mikäli haluaa käyttää sivustoon valmista mallipohjaa, tarjoaa WordPress mallipohjia enemmän kuin Webflow (Paterson, 2022) Webflow'n päätarkoituksena on kuitenkin sivuston suunnittelun vapaudessa, joten keskittyminen mallipohjiin ei ole tärkeyslistan kärjessä (Walsh, 2022).

Kummatkin tarjoavat kattavat työkalut hakukoneoptimoidun sivuston saavuttamiseksi. Webflow luo automaattisesti sisällönhallintajärjestelmän sisällön metaotsikot sekä tuottaa käyttäjän puolesta XML-sivukartan, jonka voi lähettää Googlelle, jotta Googlen hakukonerobotit tietävät miten indeksoida verkkosivustoa. Käyttäjä voi itse muokata hakukoneoptimoinnille olennaisia tietoja, kuten otsikoita, sisällön kuvauksia, avainsanoja ja kuvien sisällön kuvaustietoja, eli Alt-tekstejä. (Steele, 2022) Huuskon (2022) mukaan hakukoneoptimoinnin tekemiseen on Webflow'ssa tarkat säädöt ja se on tehty helpoksi. Webflow'ssa hakukoneoptimoidun sivuston tekeminen on nopeampaa kuin WordPressissä, mutta WordPress voi antaa tarkempia säätömahdollisuuksia lisäosien avulla (Holcombe, 2022).

Valitseminen WordPressin ja Webflow'n väliltä riippuu myös hyvin pitkälti verkkosivuston tavoitteesta sekä verkkosivuston tekijästä ja tekijän kokemuksesta. Patersonin (2022) mukaan Webflow' soveltuu käyttäjälle, jolla on vähäinen kokemus ohjelmoinnista tai ei ollenkaan, kun taas WordPress saattaa sopia paremmin kokeneemmalle kehittäjälle. Webflow'n eduksi kääntyy kuitenkin huolettomuus, sillä päivityksistä ei tarvitse erikseen huolehtia. Webflow markkinoi työkaluun myös suoraan parempana ja helpompana vaihtoehtona WordPressille. (Webflow, n.d.)

## 5 Verkkosivuston luominen Webflow'n avulla

Jotta verkkosivuston toteuttamisesta Webflow-työkalulla saadaan parempi kuva, tässä osiossa toteutetaan verkkosivusto sitä käyttäen. Tutkimuksessa otettiin käyttöön Webflow'n opiskelijoille tarjoama ilmainen CMS-tilaus, joka maksaisi normaalisti 29 dollaria kuukaudessa tai 23 dollaria, mikäli maksaa kerralla koko vuoden tilauksen. Tarkoituksena on testata Webflow'n ominaisuuksia mahdollisimman laajasti, mutta kuitenkin niin, että ne sopivat tekemälleni sivustolle. Ensiksi käydään läpi lyhyesti suunnitteluprosessia, jonka jälkeen tutkitaan Webflow'n ominaisuuksia verkkosivuston kehittämisessä. Luku keskittyy pääosin Webflow'n käyttämiseen ja sen ominaisuuksien tutkimiseen.

### 5.1 Sivuston suunnittelu

Tässä kappaleessa kerrotaan mitä suunnittelun vaiheita toteutetaan verkkosivuston luomista varten. Aluksi esitellään sivuston tavoite ja sivuston aiheen rajausta. Tämän jälkeen esitellään sivuston rakenne ja rautalankamalli sivuston etusivusta. Lopuksi kerrotaan sivustolle valituista väreistä sekä fonteista.

#### 5.1.1 Tavoitteen määrittäminen

Ensimmäisenä vaiheena tulee miettiä, mikä on verkkosivuston tavoite ja ketkä ovat sivuston kohdeyleisö. Sivuston tyyli on kuvitteellinen portfoliosivusto verkkosivuja kehittäväälle yritykselle, joten päällimmäinen tarkoitus on näyttää osaaminen asiakkaille, joten sivustolta tulee löytyä omia projekteja sekä tietoa yrityksestä. Hyvä portfoliosivusto on yksinkertainen, mutta kumminkin näyttävä ja varsinkin käyttäjäystävällinen. Sivuston sisältö on kuitenkin tämän opinnäytetyön aihealueen ulkopuolella, joten sivustolta löytyy esimerkiksi jonkin verran täytetextiä ja kuvat on ladattu ilmaisista kuvapankeista. Hakukoneoptimointiin ei tämän vuoksi keskitytty, mutta sitä on käsitelty teoriaosuudessa.

#### 5.1.2 Sivuston rakenne ja rautalankamalli

Verkkosivusto jakautuu kokonaisuudessaan neljään osaan, jotka ovat etusivu, tietoa meistä, projektit, sekä ota yhteyttä -osio. Sivustolle ei tule erillisiä alisivuja, vaan kaikki osiot ovat

samalla sivulla. Navigaatiovalikko tulee sivuston oikeaan yläreunaan, josta löytyy osiot: Tietoa meistä, projektit, sekä ota yhteyttä. Osiota klikkaamalla sivu rullautuu aina oikeaan kohtaan. Sivuston vasempaan yläkulmaan tulee paikka yrityksen logolle. Sivuston osioiden suunnittelun jälkeen siirryin Figman puolelle, jolla tein keskitason tarkkuuden rautalankamallin (Kuva 9).

Kuva 9. Figmalla luotu rautalankamalli sivuston etusivusta.



### 5.1.3 Väripaletti ja typografia

Värejä valitsiessa lähdin ensin katsomaan värien herättämiä tuntemuksia. Sininen väri kuvastaa luotettavuutta ja ammattimaisuutta, joten valitsin sen ja lähdin miettimään väripalettia (Kuva 10). Hyödynnän sivustolla monokromaattista väripalettia, eli käytän sinisen värin eri sävyjä. Monokromaattisessa väripaletissa kaikki värit sopivat keskenään yhteen ja sen avulla voi luoda visuaalisesti yhtenäisen ilmeen. Käytän myös keltaista väriä tuomaan sivuille kontrastia ja valkoista väriä joihinkin sivun teksteihin.

Kuva 10. Sivustolle valittu väripaletti.



Webflow’ssa tallensin värit värienhallinta -ominaisuudella (Kuva 11), josta haluamani värin voi aina valita eri elementeille.

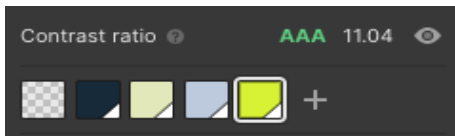
Kuva 11. Webflow’n paneeli värien hallintaan.



Typografian tulisi olla yksinkertaista ja helposti luettavaa. Webflow tarjoaa valmiita fontteja ja mahdollisuuden ladata suunnittelutyökaluun Googlen fontteja, mukautettuja fontteja sekä Adoben fonttien integroinnin. Valitsin Titillium Web -Google fontin ja sen eri paksuudet. Fontin käyttöönotto onnistui muutamalla klikkauksella.

Tekstiä lisätessä sivulle voi aina myös tarkistaa miten tekstin väri sopii taustaväriin, eli kontrastisuhteen, joka tarkoittaa näytön suurimman ja pienimmän kirkkauden välistä eroa. Esimerkiksi sivuston etusivun otsikon keltaisen tekstin (Kuva 12) kontrastisuhte on 11.04 ja WCAG-luokitus on AAA, joka tarkoittaa tehostettua kontrastisuhdetta. Mikäli kontrastisuhte on liian alhainen, ilmoittaa Webflow punaisella tekstillä ”fail”, joten on vaikea vahingossakaan valita huonoa kontrastisuhdetta omaavia värejä.

Kuva 12. Kontrastisuhteen tarkistus Webflow'ssa.



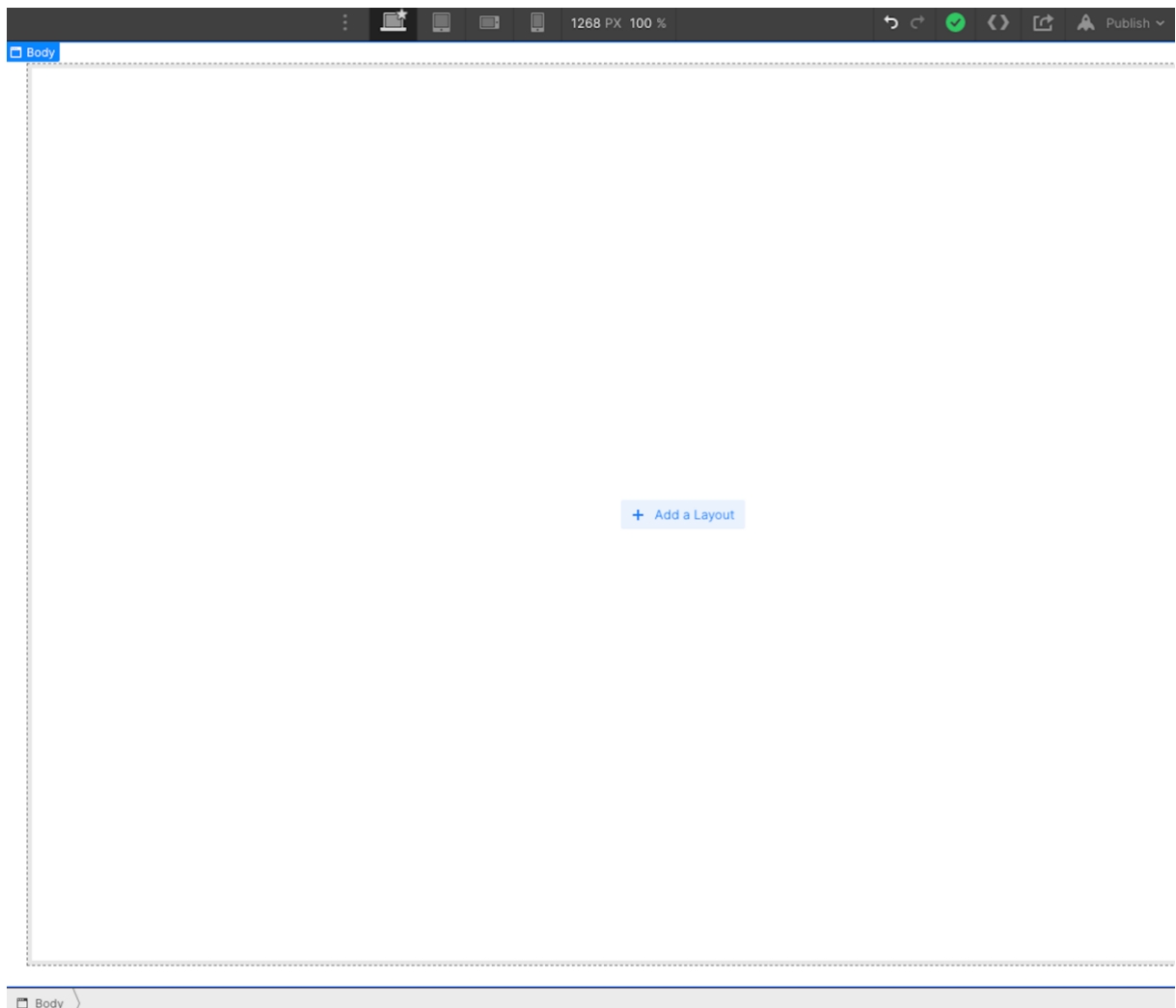
## 5.2 Verkkosivuston luominen

Sivuston luominen tapahtuu pääasiassa Webflow'n visuaalisella työkalulla, joka on nimeltään Designer. Designerin käyttäminen ei onnistunut puhelimelta, eli se vaatii aina vähintään 900 pikselin levyisen ruudun. Graafinen ulkoasu muistuttaa hyvin paljon Photoshop-kuvankäsittelyohjelmaa ja rakentuuakin hieman samalla tavalla.

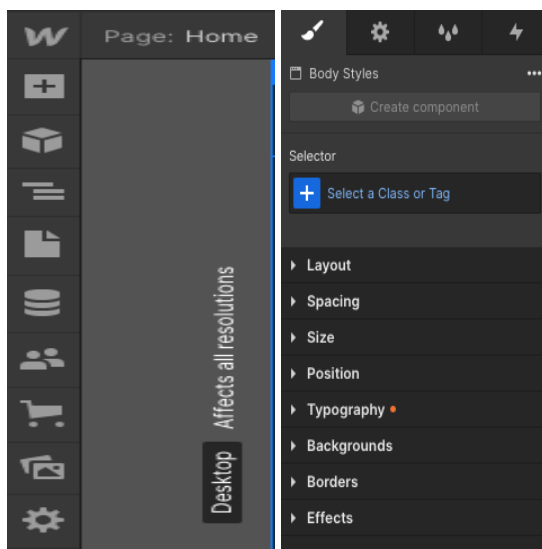
Designer rakentuu kolmesta pääosasta, joista keskellä on suunnittelupohja (engl. canvas) (Kuva 13), jonka koko määrittyy yläpalkista valitun päätelaitteen mukaan.

Suunnittelupohjaan voi tuoda erilaisia elementtejä vasemmalta löytyvästä toisesta pääosasta, eli työkaluvalikosta (Kuva 14). Kaikki mitä suunnittelupohjalla näkyy, näkyy myös julkaistulla sivustolla joitakin poikkeuksia lukuun ottamatta. Oikealla olevat paneelit ovat käyttöliittymän kolmas pääosa (Kuva 14), joka on tarkoitettu elementtien ulkoasun sekä niiden toiminnallisuuden säätämiseen. Paneelit jakautuvat neljään eri osaan, jotka ovat Style panel, Element setting panel, Style Manager ja Interactions Panel.

Kuva 13. Designerin suunnittelupohja.

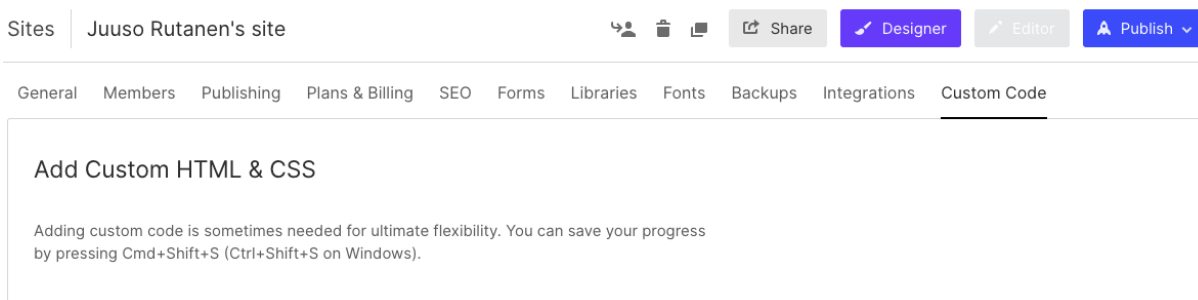


Kuva 14. Webflow Designerin vasemman puolen työkaluvalikko ja oikean puolen paneelit elementtien muokkaamiseen.



Toinen pääosa Webflow'n käyttöliittymää on Dashboard-osio (Kuva 15), josta löytyy yleisiä asetuksia ja esimerkiksi oman koodin lisäämisen mahdollisuus.

Kuva 15. Webflow'n Dashboard-osio.



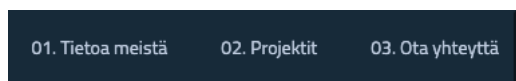
## 5.3 Sivustolla käytetyt ominaisuudet

Tässä kappaleessa kerrotaan sivustolla käytetyistä Webflow'n ominaisuuksista sekä havainnollistetaan ominaisuuksien testausta käytännön esimerkeillä.

### 5.3.1 Webflow'n NavBar-elementti

Aloitin sivuston tekemisen navigaation, eli sivuston valikon (Kuva 16) lisäämisellä hyödyntäen Webflow'ssa valmiiksi löytyvää navbar-elementtiä. Elementtien lisäämisen nopeuttamiseksi pystyy käyttämään pikanäppäintä, jolloin halutun elementin tyylin voi kirjoittaa hakuun ja raahata suunnittelupohjalle sitä kautta. Koska tekemälleni sivustolle ei tule alasivuja, eli kaikki sisältöosiot löytyvät yhdeltä sivulta, pitää sivusto jakaa osiin. Tämän voi tehdä Webflow'ssa esimerkiksi Section-elementtejä hyödyntämällä, ja laittamalla navigaation teksteihin linkityksen section-elementteihin. Navigaatiovalikon linkkejä painamalla sivu rullautuu automaattisesti animaationa oikeaan sisältöosiin. Myös mobiilinnavigaation toiminallisuus tuli navbar-elementin avulla, mutta vaatii kuitenkin vielä pientä hienosäätöä. Navigaatiosta on hyvä luoda myös komponentti (engl. component), joka tarkoittaa Webflow'n puolella uudelleen käytettävää osiota. Esimerkiksi jos tekisin sivustolle useita eri sivuja, voisin lisätä jo luomani navigaatiovalikon siirtämällä navigaatiokomponentin jokaiselle sivulle.

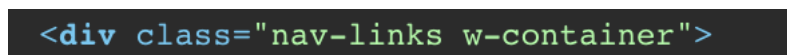
Kuva 16. Sivustolta löytyvä navigaatiovalikko.



### 5.3.2 Webflow'n Div Block-elementti

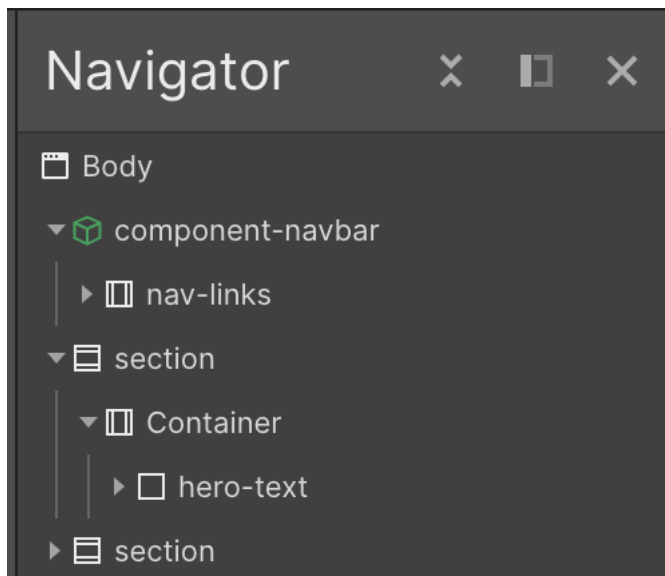
Webflow'ta käyttäessä on hyvä ymmärtää, että suurin osa suunnittelupohjaan lisätyistä elementeistä muodostuu laatikoista, joita Webflow kutsuu Div Blockeiksi. Div Block-elementti vastaa HTML:n div-tunnistetta, johon on määritetty luokka. Webflow luo automaattisesti div-tunnisteen, kun lisää elementtejä suunnittelupohjalle. Tunnistetta voi halutessaan vaihtaa asetuksista, jos haluaa käyttää semanttista HTML:ää saavutettavuuden luomiseksi. Tässä työssä div-tunniste kuitenkin riittää, joten en lähtenyt vaihtamaan tunnisteita elementteihin. Kuvassa (Kuva 17) hahmottuu Webflow'n luoma div-tunniste, jonka class-attribuuttiin on määritetty luokka "nav-links w-container".

Kuva 17. Webflow'n koodieditorissa näkyvä div-tunniste.



Div Block -elementti löytyy erikseen ja muut elementit ovat ennalta määritettyjä tyyliä Div Blockeista. Esimerkiksi Container-elementti on Div Block, jonka maksimileveys on määritetty valmiiksi 940 pikseliin ja elementti sijoittuu automaattisesti sivuston keskelle. Section-elementti taas on koko sivun levyinen elementti, ja sen rajoituksena on, että sitä ei voi sijoittaa eri elementtien sisään, jolloin puhutaan termistä periytyminen (engl. nesting). Vaikka section-elementin nimi voisi viitata HTML:n `<section>` -tunnisteeseen, on se kuitenkin Webflow'n muodostamassa koodissa div-tunniste. Näiden eri Div Block-elementtien variaatioiden tarkoitus on nopeuttaa verkkosivuston toteuttamista ja niitä on hyvä käyttää luettavan rakenteen (Kuva 18) muodostamiseksi.

Kuva 18. Sivuston rakennetta Webflow'n suunnittelutyökalussa.



### 5.3.3 Webflow'n Grid-elementti

Grid-elementti luo sivustolle valmiin ruudukon, joka on sama kuin CSS-tyylikielen Grid Layout. Yhteen ruutuun voi käyttää aina ainoastaan yhtä elementtiä, mutta ruutuihin voi lisätä myös Div block-elementin, jolloin yhteen ruutuun saa tarvittaessa enemmän eri elementtejä. Grid-elementtiä käyttämällä oli helppo asetella elementtejä ja säätää niiden kokoa. Hyödynsin Grid-elementtiä tekemällä sivustolle alatunnisteen (engl. footer). Kuvasta (Kuva 19) näkyy, kuinka eri sisältöosiöt ovat ruutujen sisällä ja voisin tarvittaessa lisätä uusia ruutuja sekä säätää ruutujen kokoja.

Kuva 19. Grid-elementin hyödyntäminen alatunnisteessa.



### 5.3.4 Tiedostojen lisääminen

Tiedostoja kuten esimerkiksi kuvia ja kuvituksia voi lisätä Assets-paneelin kautta raahaamalla tiedostoja omasta kansioista tai Assets-paneelin upload-painikkeella. Kompressoitin ensin käyttämäni kuvat tinypng-palvelua käyttämällä, jonka jälkeen raahasin kuvat Webflow'n suunnittelutyökaluun. Mikäli haluaa käyttää webgrafiikkaa, pitää ne lisätä myös erikseen. Työssäni käytin muutamia svg-tiedostoja, kuten esimerkiksi sosiaalisen median painikkeita.

### 5.3.5 CMS-työkalun käyttäminen

CMS-työkalulla saadaan rakennettua sisällönhallintajärjestelmistä tuttuja kustomoituja sisältötyyppejä, joiden avulla sivujen sisältöä voidaan monipuolisesti tallentaa tietokantaan ja esittää sivuilla. Käytin Webflow'n tarjoamaa CMS Collections -työkalua projektit-osion luomiseen. CMS-työkalua käyttämällä pystyin luomaan ulkoasun haluamani mukaiseksi, mutta tietojen päivitys ja uusien projektien lisääminen onnistuu hieman helpommin hyödyntämällä CMS-työkalua.

Aloitin osion luomisen samalla tavalla kuin muidenkin osioiden, eli lisäsin section- ja container-elementit sivulle. Tämän jälkeen lähdin käyttämään CMS Collections-työkalua, jonne loin Projektit-nimisen kokoelman (Collection), mihin lisäsin kolme projektia. Kokoelmat tallennetaan tietokantaan, josta niitä voidaan monipuolisesti hakea ja esittää sivuilla. Tiedot voisi tuoda myös CSV-tiedostolta ja tietokantojen tiedot saa tarvittaessa myös ulos CSV-tiedostona. Kokoelmaa luodessa pitää valita kokoelman nimi, tietojen säilytyksen osoite sekä kokoelman kentät (Kuva 20). Kentät voi joko luoda kokonaan itse tai hyödyntää valmista Webflow'n tarjoamaa pohjaa esimerkiksi blogia varten.

Kuva 20. Tietokantaan valitut kentät.

Collection fields ? [+ Add New Field](#)

**Basic info**

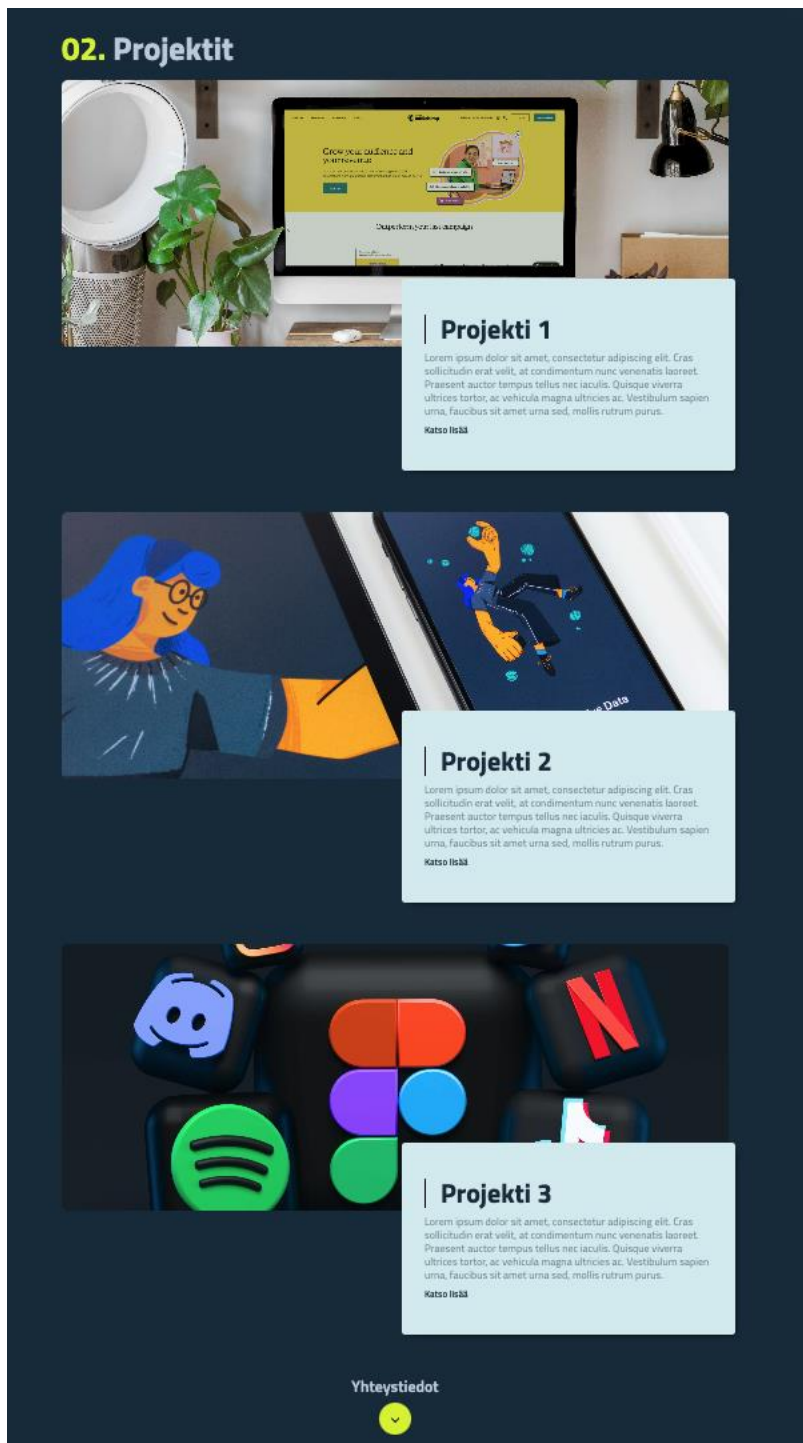
T	Projektin nimi (Plain text)	Required Field
T	Slug (Plain text)	Required Field

**Custom fields**

T	Projektin kuvaus (Plain text)	
🖼️	Projektin kuva (Image)	
🎨	Projektin väri (Color)	
☑️	Projektin tyyppi (Option)	
T	Projektin vuosi (Plain text)	
T	Asiakkaan nimi (Plain text)	
📄	Projektin tiedot (Rich text)	
🔗	Projektin linkki (Link)	
🖼️	Projektin kuvat (Multi-image)	
+	Add Field	

Kuvassa (Kuva 21) näkyvässä valmiissa lopputuloksessa tekstit, kuvat ja värit tulevat suoraan tietokannan tiedoista ja niiden päivittäminen oli mielestäni helppoa. Designerin puolella voi itse valita mitä sisältökenttiä käyttää ja päätin käyttää vain muutamaa. Asetuksista voi myös käydä poistamassa julkaisun näkyvistä, jolloin se jää kuitenkin talteen ja sen voi laittaa halutessaan uudelleen näkyviin.

Kuva 21. Esimerkki CMS-työkalun hyödyntämisestä.



Mikäli haluaisi antaa sivuston käyttäjälle mahdollisuuden suodattaa sisältöä eri kategorioiden perusteella, voisi sen tehdä integraation avulla tai itse ohjelmoimalla.

### 5.3.6 Sisällön päivittäminen

Webflow’sta löytyy erillinen Editor-ominaisuus, joka on tarkoitettu sivuston sisällön muokkaamiseen. Tätä ominaisuutta voisi käyttää, jos vaikkapa jakaa rooleja sivuston tekijän ja sisällön päivittäjän kesken. Kokeilin ominaisuutta lisäämällä itselleni Content Editor-roolin toiselle sähköpostilleni. Sain sähköpostiini kutsun, jota kautta pääsin käyttämään Editor-ominaisuutta erillisen käyttäjän näkökulmasta. Sivuston tekstien ja kuvien päivittäminen tuntui olevan selkeää ja sisällönhallintajärjestelmään pystyi lisäämään nopeasti uutta sisältöä. Mitenkään laajemmin sivua ei pysty muokkaamaan Editorissa, eli ainoastaan sivuston sisältöä. Poikkeuksena oli JavaScriptillä luotu itsestään kirjoittuva teksti, jota ei päässyt muokkaamaan. Sivulle voi lisätä myös salasanasuojauksen, jolloin sisällön päivittäjä pääsee päivittämään sivua ainoastaan tietämällä salasanan. Kokeilin Webflow’n Editorin käyttöä myös puhelimella ja ainakaan tällä hetkellä tietojen päivittäminen tai sivuston muokkaus ei onnistunut puhelimella.

### 5.3.7 Webflow’n kehitysympäristö

Webflow’n itse luomaa koodia ei pääse muokkaamaan, eli koodia pääsee ainoastaan itse lisäämään eri tavoilla. Webflow tukee JavaScriptiä, CSS:ää ja HTML, eli serveripuolen kieliä kuten PHP:ta ei voi kustomoinneissa hyödyntää. Yksi tapa koodin lisäämiseen on HTML-embed-elementtiä käyttämällä. Kokeilin toimintoa lisäämällä Google Maps-sijainnin (Ohjelmakoodi 1) sivustolle. Lisäsin ensiksi Div Block -elementin, johon lisäsin Embed-elementin. Google Mapsin sivuilta saa kartan osoitteen, jonka voi lisätä koodieditoriin ja koodia voi muokata sopivaksi.

Ohjelmakoodi 1. Webflow’n koodieditoriin lisätty Google Mapsin upotettu koodi.



```
HTML Embed Code Editor
Custom Code:
1 <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d127248.38826825371!2d24.87860094868
2 width="100%" height="100%" style="border:0;"
3 allowfullscreen="" loading="lazy"
4 referrerpolicy="no-referrer-when-downgrade"></iframe>
```

Tässä tapauksessa muokkasin ainoastaan leveyttä ja korkeutta. HTML-embed-elementin etuna on, että voin itse valita suunnittelupohjalta, mihin kohtaan haluan kustomoidun koodin. Embed-elementtiä käyttämällä lopputuloksen näkee myös heti suunnittelupohjalla (Kuva 22), lukuun ottamatta JavaScriptillä tehtyä koodia.

Kuva 22. Valmis lopputulos HTML-embed-elementillä lisätystä kartasta.

**03. Ota yhteyttä**

Kiinnostuitko? Laita meille viestiä, niin jutellaan lisää. Tai tule pyörähtämään toimistollamme.

**Nimi**

**Sähköposti \***

**Yritys**

**Puhelinnumero**

**Viesti**  
 Sana on vapaa

**Lähetä**

**Helsinki**  
 Helsinki  
 Reittiohjeet  
 Näytä suurempi kartta

linna  
 Junkkiranra

Google  
 Pikanäppäimet Karttatiedot ©2022 Google Käyttöehdot Ilmoita karttavirheestä

Halusin myös kokeilla, miten voisin lisätä itse tekemäni CSS-luokan, vaikka Webflow'n tyylipaneelin avulla tämän voisi luoda nopeammin. Tätä varten piti myös lisätä HTML-embed-elementti suunnittelupohjalle ja lisäsin koodieditoriin CSS-valitsimen style-tunnisteiden sisälle (Ohjelmakoodi 2). Loin yksinkertaisen logon, jonka sijoitin sivuston vasempaan yläkulmaan. Logosta pystyi tekemään myös komponentin, eli se on helposti käytettävissä tarvittaessa muissakin osioissa.

Ohjelmakoodi 2. Webflow'n sisäiseen koodieditoriin tehty syntaksi.

```

HTML Embed Code Editor

Custom Code:

1  <style>
2  .laatikko-esimerkki {
3  width: 5rem;
4  height: 4rem;
5  background: #dbf40b;
6  margin: auto;
7  color: #163042;
8  display: flex;
9  justify-content: center;
10 align-items: center;
11 font-family: 'Titilium Web', sans-serif;
12 font-size: 1rem;
13 font-weight: 400;
14 }
15 </style>
16
17 <h1 class="laatikko-esimerkki">Logo </h1>

```

Mikäli HTML Embed -koodieditoriin lisää CSS-tyylikieltä, ei se tule kuitenkaan näkyviin Webflow'n luomaan CSS-tyylitiedostoon, vaan se on upotettuna HTML-tiedostoon, kuten embed (engl. upottaa) sanakin viittaa. Jos elementtiä kuitenkin muokkaa Webflow'n Designerissä, ilmestyy luokka tämän jälkeen näkyviin myös Webflow'n CSS-tiedostoon. Halutessaan elementtiä voi siis muokata sekä itse CSS-tyylikielillä tai Webflow'n tyylipaneelissa.

Lisäksi Dashboard-osiosta löytyy editori oman koodin lisäämiselle ja sama tapa löytyy myös suoraan Designerin asetuksista. Dashboardin ja Designerin koodieditoreissa on kuitenkin olennainen ero, sillä Dashboardin koodieditoriin tehty koodi vaikuttaa koko sivustolle ja Designerin puolella tehty koodi vaikuttaa aina vain valittuun sivuun.

Halusin luoda sivulle myös tekstiä (Kuva 24), joka kirjoittuu sivulle automaattisesti scriptin (Ohjelmakoodi 3) avulla. Asetukset löytyvät editorit erikseen <head>- ja </body> -tunnisteille ja JavaScript laitetaan usein ennen </body> -tunnistetta, jotta selain lukee sen HTML:n jälkeen. Käytin JavaScript ohjelmointikielen Typing Animation-kirjastoa Typed.js ja muokkasin scriptiä haluamani mukaiseksi. Strings -kohtaan voi asettaa sanat merkkijonona,

josta sanat kirjoittuvat sivulle ja silmukka (engl. loop) toistaa sanat aina uudelleen. Scripti hyödyntää ulkoista JavaScript-tiedostoa.

Ohjelmakoodi 3. JavaScript-koodin lisääminen Webflow'n editoriin.

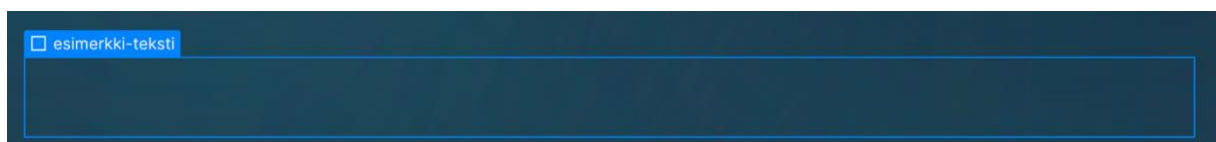
```

Before </body> tag
1  <script src="https://cdn.jsdelivr.net/npm/typed.js@2.0.9"></script>
2
3  <script>
4      var typed4 = new Typed('.esimerkki-teksti', {
5          strings: ['Uniikit', 'responsiiviset', 'nopeat'],
6          typeSpeed: 75,
7          backSpeed: 25,
8          bindInputFocusEvents: true,
9          cursorChar: '_',
10         showCursor: false,
11         loop: true,
12         backDelay: 1000,
13         startDelay: 1000
14
15     });
16 </script>
17

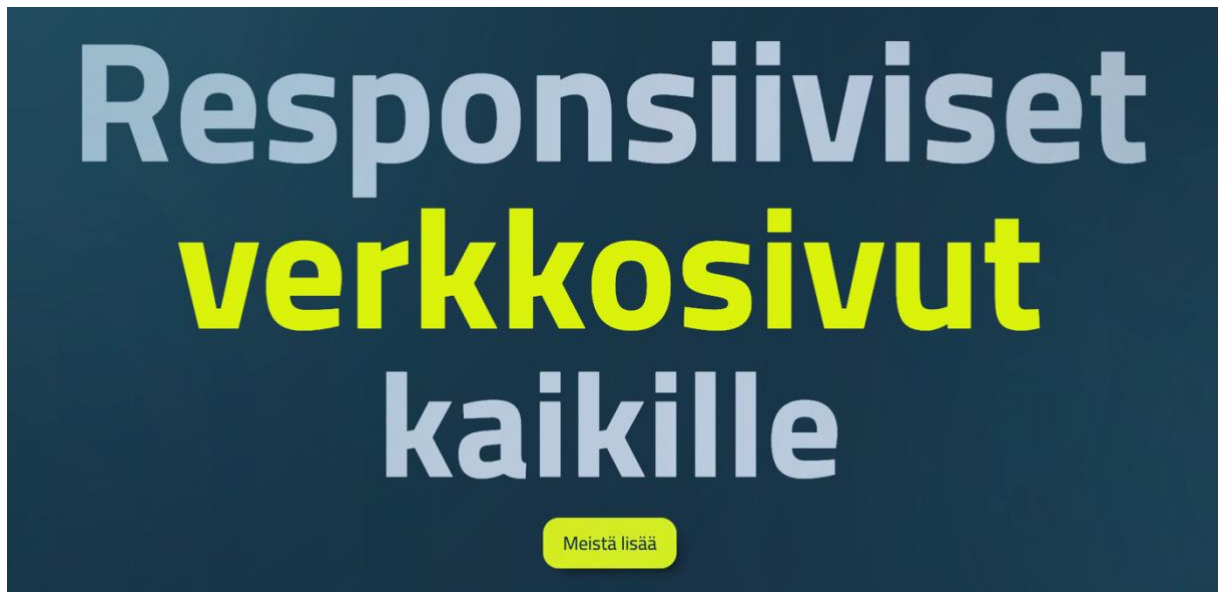
```

Jotta koodi toimii Webflow'n sivustolla, pitää Designerin puolelta lisätä elementti ja nimetä se. Kuvassa (Kuva 23) näkyy elementti Designerin puolelta ja luokan nimi. Pitää myös huomata, että itse lopputulos ei näy tässä tapauksessa suunnittelutyökalun puolella, vaan sivusto pitää ensin julkaista. Elementille piti asettaa kiinteä korkeus, sillä muuten muut elementit liikkuvat samalla, kun teksti kirjoittuu.

Kuva 23. Elementti suunnittelupohjalla ja luokan nimi.



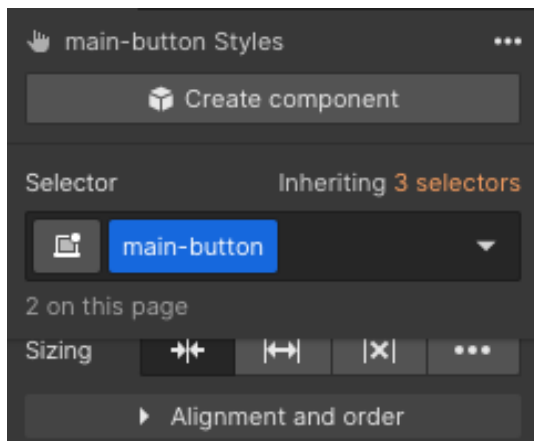
Kuva 24. Valmis lopputulos JavaScript-elementin käytöstä. (Huom. ylimmän rivin teksti kirjoittuu itsestään ja sana vaihtuu)



### 5.3.8 Tyylipaneelin käyttäminen

**Luokkien** käyttäminen on Webflow'ssa tärkeää samalla tapaa kuin CSS-tyylielessä. Webflow'ssa luokka muodostuu aina, kun jotain elementtiä muokataan. Webflow nimeää luokan automaattisesti, mutta nimen voi itse vaihtaa ja se onkin suositeltavaa eri luokkien hahmottamiseksi sekä yhtenäisen rakenteen muodostamiseksi. Luokkien nimeämiseen voi käyttää samoja periaatteita, mitä käytetään CSS-luokkien nimeämisessä ja luokat toimivat muutenkin samalla tyylillä kuin CSS-tyylielessä, mutta ID-valitsinta ei kuitenkaan ollut käytössä valmiina, joten sen käyttäminen vaatisi koodieditorin käyttämistä. Webflow'n käytön aloitettuani en vielä keskittynyt luokkien nimeämiseen, mikä teki Webflow'n Navigator-palkista hieman vaikeasti luettavan ja luokkien nimeäminen jälkikäteen on hieman hankalaa, mikäli luokkia on paljon. Jos normaalisti käyttäisi CSS-tyylikieltä, voisi luokkien nimeäminen olla hieman nopeampaa. Mikäli samaa luokkavalitsinta käytetään toisessa elementissä, käyttää elementtiä siihen määritetyn luokan tyyliä. Kun loin sivuston etusivulle painikkeen, Webflow loi automaattisesti luokan, jonka nimesin itse nimellä main-button ja voin käyttää samaa luokkaa myös muissa sivuston painikkeissa. Kuvista näkyy luokka Webflow'n suunnittelutyökalussa (Kuva 25) ja CSS-valitsimen koodi, kun koodia (Kuva 26) tarkastelee Webflow'n Export code -toiminnon kautta.

Kuva 25. luokkavalitsin Webflow'n suunnittelutyökalussa.

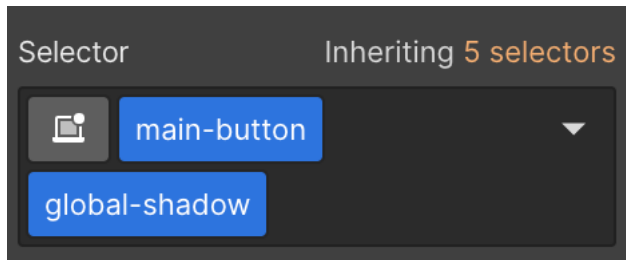


Kuva 26. CSS-valitsimen koodi Webflow'n tuottamassa koodissa

```
.main-button {
  position: static;
  display: inline-block;
  border-radius: 20px;
  background-color: #4f9bdd;
  text-align: left;
}
```

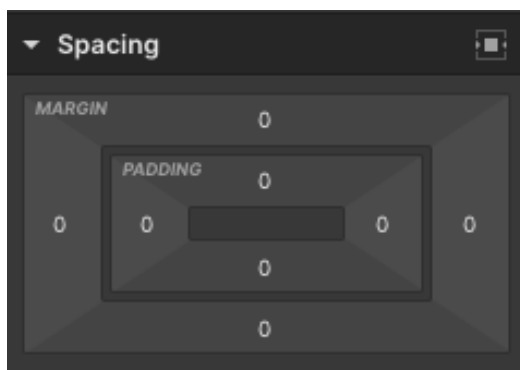
Luokan voi myös kopioida ja luoda uuden luokan käyttäen vanhaa luokkaa pohjana. Jos haluan käyttää samaa luokkaa, mutta pienillä muutoksilla, voi käyttää luokkayhdistelmää (engl. combo class), jolloin luokkayhdistelmä perii pohjaluokan säännöt. Ero luokan kopioimiseen on siis se, että luokkayhdistelmää käyttävä elementti muuttuu, mikäli pohjaluokkaan tehdään muutoksia. Lisäksi on mahdollista käyttää globaaleja luokkia, eli mikäli esimerkiksi haluaisi eri elementeille yhtenäisen varjostuksen, voi siihen tarkoitukseen käyttää globaalia luokkaa. Globaalin luokan käyttämistä varten on hyvä luoda kokonaan uusi elementti, johon CSS-säännöt liitetään. Loin kokonaan uuden sivun, johon lisäsin Div Block-elementin globaalin luokan tekemistä varten ja käytin globaalia luokkaa sivuston painikkeisiin (Kuva 27) sekä projektisivulla oleviin teksteihin.

Kuva 27. Globaalin luokan käyttäminen painikkeen varjostuksessa.



**Spacing-ominaisuudella** (Kuva 28) voi säätää elementtien välistä tyhjää tilaa eli marginaalia (engl. margin) sekä täytettä (engl. padding), eli rajan ja sisällön välistä tilaa. Ominaisuus toimii CSS:stä tutun laatikkomallin mukaan, mutta visuaalisesti. Reunusta (engl. border) voi säätää Webflow'n tyylipaneelin Border-ominaisuuden avulla. Webflow'ta käyttäessä Spacing-ominaisuutta tulee käytettyä useaan otteeseen ja hyödynsin sitä esimerkiksi luomaan tilaa eri sisältöosioiden välille.

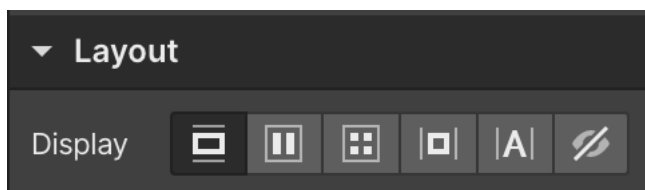
Kuva 28. Spacing-työkalu.



**Layout-työkalu** toimii samalla tavalla kuin CSS:n display-ominaisuus. Display-ominaisuus on suuressa roolissa Webflow'n käytössä, sillä sen avulla voi hallita kuinka elementit näytetään suhteessa toisiinsa. Jos ymmärtää CSS:ää, ei ominaisuuden käyttäminen pitäisi tuottaa vaikeuksia, mutta muutoin voi olla alkuun hankala asetella eri elementtejä. Display-arvot

vasemmalta oikealle (Kuva 29) ovat block, flex, grid, inline block, inline sekä none. Varsinkin flex-arvoa käytin tekemälläni sivustolla useaan kertaan.

Kuva 29. Layout-työkalu.



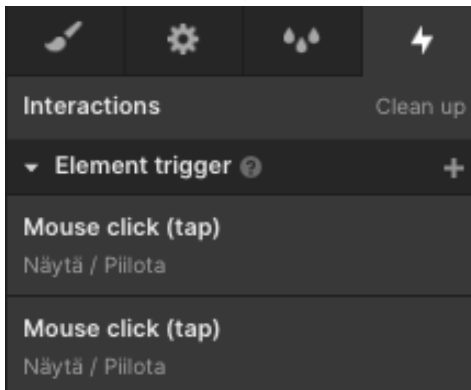
Elementtien sijainnin hallintaan löytyy myös **Position-ominaisuus**, joka toimii CSS:n tavoin. Se tarjoaa neljä eri arvoa, eli absoluuttisen (engl. absolute), relatiivisen (engl. relative), kiinteän (engl. fixed) ja normaalin (engl. static) asettelun. Normaali asettelu on aina vakiona. Position-ominaisuutta hyödynsin esimerkiksi animaatioiden tekemisessä, jolloin pystyin säätämään elementtien Z-indeksiä ja määrittämään kiinteitä sijainteja.

### 5.3.9 Interaktiot-paneelin käyttäminen

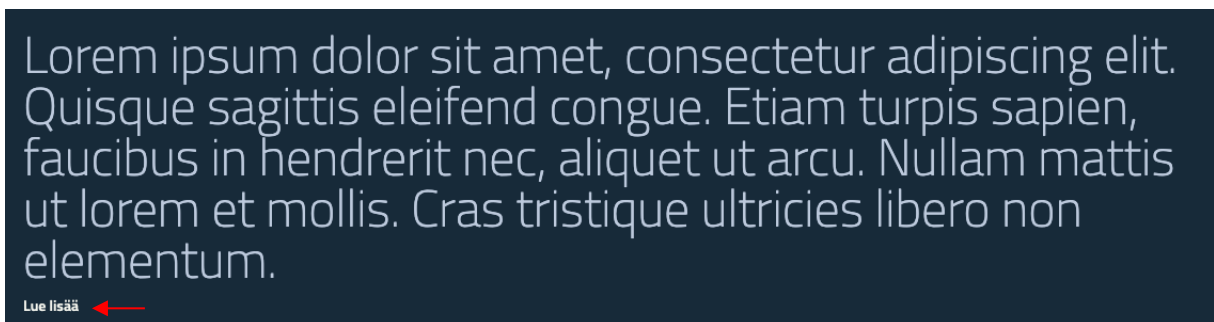
Interaktiivisten elementtien luonti on yksi Webflow'n suurimmista vahvuuksista. Elementteistä voi tehdä interaktiivisia Webflow'n Interactions-paneelin avulla, joka lisää elementteihin JavaScript-koodia, mutta JavaScriptia ei kuitenkaan tarvitse osata paneelin käyttämiseksi. Animaatioita ja interaktioita voi tarvittaessa käyttää myös ainoastaan tietyllä päätelaitteella. Kokeilin Interactions-paneelia luomalla sivustolle yksinkertaisen vuorovaikutuksen, eli toiminnon, jossa elementti vaatii sivuston käyttäjältä toimenpiteen kuten esimerkiksi hiiren klikkauksen. Tein yksinkertaisen "Lue lisää" -tekstin "Tietoa meistä" -osion yhteyteen. "Lue lisää" -tekstiä klikkaamalla aukeaa kappale tekstiä, jonka jälkeen "Lue lisää" -tekstin tilalle ilmestyy "Piilota" -teksti ja sitä klikkaamalla tekstikappale menee piiloon ja "Lue lisää" -teksti on taas näkyvillä.

Toimintoa varten Interactions-osiossa piti luoda kaksi eri elementin laukaisijaa (Kuva 30) ja niihin animaatiot, eli ensimmäiseksi tekstin saamiseksi näkyville (Kuva 31) ja toinen tekstin piilottamiseksi (Kuva 32).

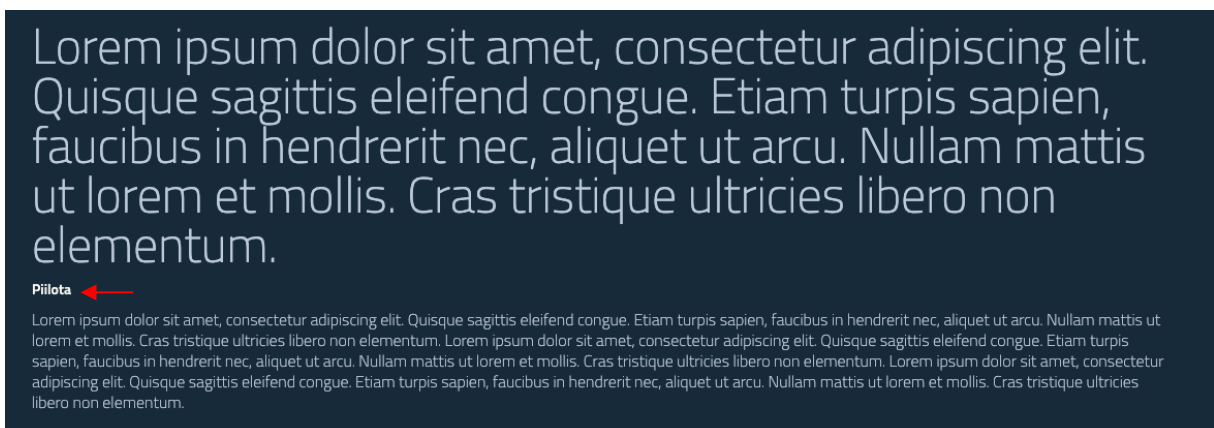
Kuva 30. Elementin laukaisijat.



Kuva 31. Interaktiivisen elementin ensimmäinen klikkaus.

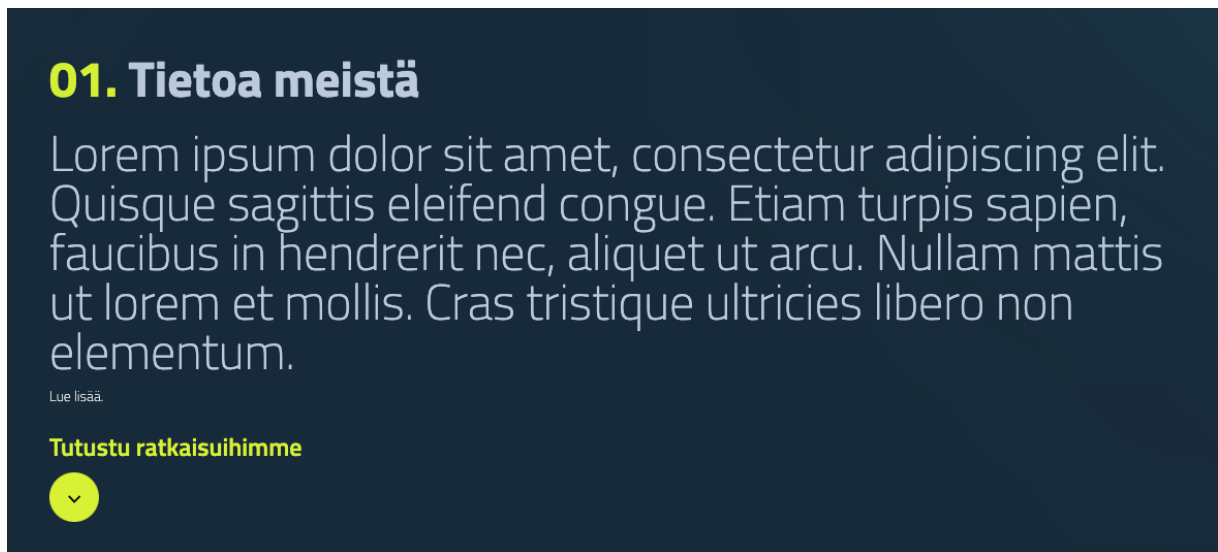


Kuva 32. Interaktiivisen elementin toinen klikkaus.



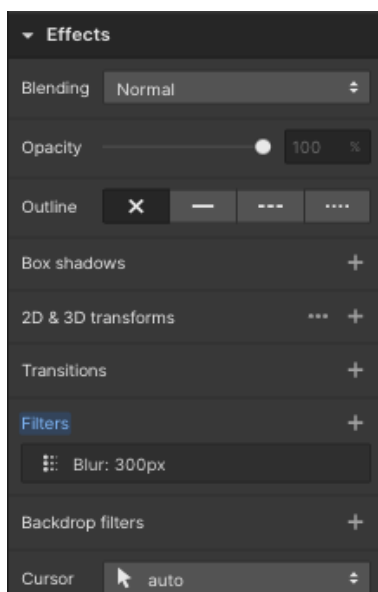
Kuvasta (Kuva 33) näkyy valmis lopputulos Tietoa meistä-osiosta, ennen käyttäjän ensimmäistä klikkausta. Kuvassa näkyvästä painikkeesta pääsee etenemään seuraavaan sisältöosiin.

Kuva 33. Valmis lopputulos Tietoa meistä-osiosta.



Halusin tehdä sivulle myös animaation (Kuva 36), joka toimii toistorakenteena ja aina kun sivusto latautuu. Ideana oli luoda sivustolle hieman lisää visualisuutta, joten loin ensin neljä Div Block-elementtiä, jotka muotoilin ympyräksi ja muokkasin ympyröitä haluamani mukaisiksi. Sumennuksen ympyröihin pystyi luomaan tyylipaneelin **Effects-työkalun** (Kuva 34) Filters-ominaisuuden avulla.

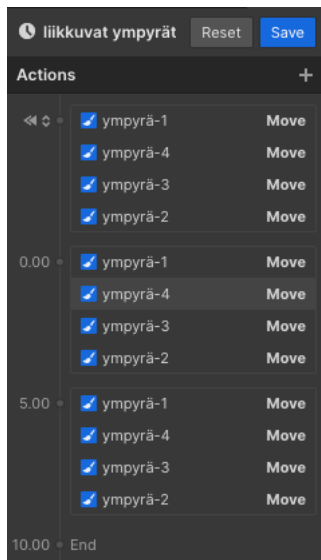
Kuva 34. Effect-työkalun hyödyntäminen.



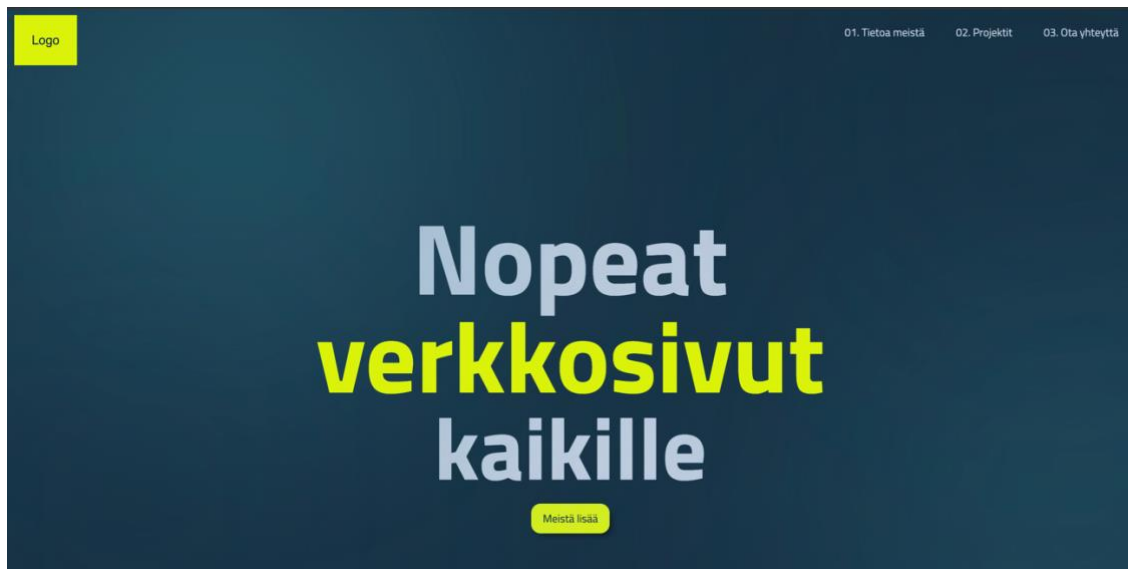
Kun ympyrät oli luotu ja sijoitettu sivuston eri kohtiin, tein jokaiseen ympyrään animaatiot (Kuva 35) käyttäen Interactions-paneelin Page Trigger-ominaisuutta, jolloin animaatio alkaa

sivuston avattaessa. Animaatiot toimivat siten, että sumennetut ympyrät liikkuvat sivustolla eri suuntiin ja animaatio toistuu toistorakenteen avulla. Move-toiminnolla pystyi säätämään, kuinka kauan ja mihin suuntiin ympyrät liikkuvat x- ja y-akseleilla.

Kuva 35. Animaation asetukset liikkuviin ympyröihin.



Kuva 36. Valmis lopputulos animoiduista sumennetuista ympyröistä.

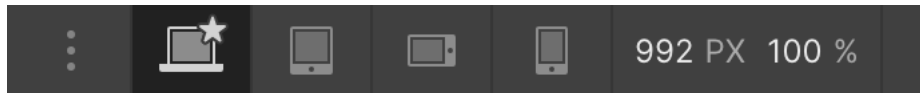


Lisäksi tein sivuston painikkeista interaktiivisia lisäämällä niihin elementin laukaisijan ja käytin tällä kertaa On hover-toimintoa, eli kun käyttäjä vie hiiren painikkeen päälle, vaihtaa painike muotoaan. Kyseisen toiminnon käyttäminen oli hyvin yksinkertaista ja onnistui muutamalla klikkauksella.

### 5.3.10 Pysäytyspisteet

Suunnittelupohjan ylälaidasta (Kuva 37) löytyy vaihtoehdot pysäytyspisteille (engl. breakpoint), jolloin suunnittelupohjan koko määrittyy valitun pysäytyspisteen mukaiseksi ja sivua voi muokata tällöin sopivaksi valitulle pysäytyspisteelle, eli valitun näytön koolle. Pysäytyspisteiden avulla sivuston saa siis parannettua sivuston responsiivisuutta. Kuvassa (Kuva 37) näkyvä tähtisymboli pätee kaikkiin resoluutioihin ja tähdettömät vastaavat CSS:n mediakyselyitä maksimiarvolla.

Kuva 37. Pysäytyspisteiden valinta.



Käytin paljon relatiivisia mittayksiköitä, joten kovin paljoa muutoksia ei tarvinnut tehdä eri pysäytyspisteille. Jotkut elementit tosin tarvitsivat pientä hienosäätöä, mutta se ei tuottanut ongelmia. Tein tämän aivan loppuksi, joten kokemusta Webflow'n käytöstä oli jo kertynyt, mutta aloittelijalle tämä voisi aiheuttaa vaikeuksia. Kuvasta (Kuva 38) näkyy valmis lopputulos etusivun mobiiliversiosta.

Kuva 38. Valmis lopputulos sivuston etusivun mobiiliversiosta.



## 6 Johtopäätökset ja pohdinta

Teoriaosuudesta saatu käsitys Webflow'n sopivuudesta ammattilaisten käyttöön osoittautui paikkaansa pitäväksi. Sivuston luominen tuntui myös samalta, kuin olisi luonut sivustoa HTML:llä ja CSS:llä, mutta että pääsee heti rakentamaan itse sivuja ja ilman huolia virheellisestä koodista. Webflow huolehtii myös sivuston html-perusrakenteesta ja sisällyttää tähän kaikki vaadittavat elementit sekä linkittää automaattisesti kustomoidut CSS- ja JavaScript-tiedostot. Webflow'n käytön opetteluun meni hieman enemmän aikaa, kuin olin kuvitellut, joten rajasin hieman sivustolle suunniteltuja toiminnallisuuksia. Lisäksi kokeilin usein eri toimintoja ja päädyin välillä vaihtamaan jo tekemääni ratkaisua, mikä vei myös aikaa. Webflow'n sisäinen versionhallinta nousi tässä kuitenkin esiin, sillä Webflow luo sivustosta automaattisesti varmuuskopioita, mikä oli hyödyllistä, jos halusin palata vanhaan tyyliin. Kokonaisuudessaan projekti onnistui hyvin ja onnistuin tutkimaan Webflow'n eri ominaisuuksia sekä toteuttamaan sillä toimivan responsiivisen verkkosivuston.

Webflow'n mukaan sivuston luominen ei vaadi osaamista web-tekniikoista, mikä toki pitää paikkansa, mutta varsinkin CSS:n ja HTML:n perusteiden osaamisesta on sivustoa luodessa suuri hyöty, mikäli sivuston luo tyhjältä pohjalta. Etenkin ymmärrys CSS:n ja HTML:n säännöistä ja periaatteista on Webflow'n käytössä keskeisessä roolissa, mikäli haluaa saada Webflow'n käytöstä eniten irti. Vaativimpien interaktioiden ja animaatioiden käytön opettelu on myös alkuun aikaa vievää, joten niiden käyttö jäi vähäisemmäksi. Webflow'n tarjoamat opetusmateriaalit olivat kumminkin erinomaisia, joten en kokenut turhautumista eri tekniikoiden opetteluun. Webflow'n käyttöliittymä tuntui myös luontevalta käyttää.

Työn edetessä Webflow'n tarjoamia hyötyjä oli helpompi ymmärtää ja mikäli tekisin nyt uuden sivuston Webflow'lla, olisi sivuston tekeminen huomattavasti nopeampaa ja saisin paremman sivuston aikaiseksi. Webflow sopii etenkin sellaisille verkkosivustoille, joissa sisältö on keskeisessä roolissa. Sisällönhallintajärjestelmän käyttö oli suhteellisen helppoa ja varsinkin uusien tietojen lisääminen tietokantaan. Lisäksi Editor-työkalu tarjoaa helpon tavan päivittää sivuston sisältöä. Sisältöä ei kuitenkaan pystynyt päivittämään mobiililaitteella, mikä voisi olla joskus tarpeen.

Webflow mahdollistaa myös visuaalisesti näyttävien sivustojen luonnin ilman ohjelmoinnin osaamista. Varsinkin interaktiivisten elementtien ja animaatioiden teko oli kätevää, joskin tekniikoiden opettelu vei jonkun verran aikaa. Jos koodin kustomoinnille on kumminkin tarvetta, onnistuu oman koodin lisääminen melko helposti, kuten työssä havainnollistettiin. Mikäli haluaa kuitenkin yksinkertaisen verkkosivun helposti ja nopeasti, voisi Webflow'n sijaan käyttää muita tarjolla olevia vaihtoehtoja. Kunhan Webflow'n käytön kuitenkin ensin oppii, saa sillä varmasti ammattimaisen verkkosivuston tehtyä suhteellisen lyhyessä ajassa. Koin Webflow'n tarjoamat tilausvaihtoehdot myös hieman epäselväksi ja varsinkin Enterprise-tilauksen tarjoamat mahdollisuudet jäivät pimentoon, sillä sen saa ainoastaan tarjouspyynnöllä. Webflow'n valintaa miettiessä verkkosivuston toteuttamista varten kannattaa pohtia tarkasti, mitä sivustolta haluaa. Käytettäviä ominaisuuksia voi olla muutoin helposti liikaa, jolloin saattaa maksaa turhasta. Webflow voisi soveltua hyvin myös freelancereille tai digitoimistoille, jotka kehittävät verkkosivuja asiakkaille. Asiakkaiden on helppo päivittää sivuston sisältöä Editor-työkalun avulla.

## 7 Yhteenveto

Työn teoriaosuuden tutkimuskysymyksinä käsiteltiin verkkosivuston suunnittelua ja verkkosivustojen toteuttamisen eri vaihtoehtoja. Lisäksi tutkimuskysymyksenä oli Webflow'n eroavaisuudet sen kilpailijoihin. Teoriapohjasta saimme selville suunnittelulle tärkeitä tekijöitä sekä verkkosivuston toteutuksen tekniikoita ja toteutuksessa huomioitavia yksityiskohtia. Lisäksi teoriaosuus antoi tietoa Webflow'n eroavaisuuksista muihin rakennustyökaluihin sekä sisällönhallintajärjestelmä WordPressiin. Käytännön osan projekti havainnollisti Webflow-työkalun toimintaperiaatteita ja niiden hyötyjä verkkosivuston toteuttamiseen. Verkkosivuston toteutus Webflow'lla antoi myös käsitystä siitä, minkälaisiin projekteihin Webflow'ta voisi hyödyntää.

Opinnäytetyötä tehdessä opin verkkosivuston suunnittelun periaatteita ja tekniikoita. Lisäksi opin ymmärtämään Webflow-työkalun toimintaa ja sen mahdollisuuksia verkkosivujen toteuttamiseen. Webflow'n käyttäminen auttoi myös hahmottamaan paremmin CSS-tyylikielen käyttöä ja toimintaperiaatteita.

Webflow tarjoaa vielä laajasti tutkimisen aiheita tulevaisuudessa. Opinnäytetyön kirjoituksen hetkellä Webflow'ssa on testausvaiheessa automaatioiden tekeminen Logic-ominaisuudella, joten sen käyttöä voisi tulevaisuudessa tutkia. Lisäksi testausvaiheessa on Webflow'n Memberships-työkalu, jonka avulla voi luoda tilausmallilla toimivia verkkosivustoja, joiden sisältö näkyy vain kirjautuneille/rekisteröityneille käyttäjille erilaisten ehtojen mukaan. Kirjoitushetkellä Webflow julkaisi myös erillisen kehittäjiä varten tehdyn sivun, josta löytyy useita eri kehittäjille tarkoitettuja ominaisuuksia kuten Webhookit ja Webflow'n rajapintaratkaisut. Verkkosivujen rakennustyökalut kehittyvät varmasti tulevaisuudessa, joten jään mielenkiinnolla seuraamaan Webflow'n kasvua ja kehitystä.

## Lähteet

- Babich, N. (2017, kesäkuuta 23). *10 Tips On Typography in Web Design*.  
<https://uxplanet.org/10-tips-on-typography-in-web-design-13a378f4aa0d>
- Babich, N. (2021, tammikuuta 13). *Combining Wireframes & Sitemaps for Creating Effective User Flows*. <https://xd.adobe.com/ideas/process/wireframing/wireframe-map-tips-for-clean-user-flow/>
- Beaird, J., & George, J. (2014). *The Principles of Beautiful Web Design*. Sitepoint Pty Ltd.
- Benjamins, S. (2022a, kesäkuuta 14). *WordPress vs Website Builders*.  
<https://www.sitebuilderreport.com/blog/wordpress-vs-website-builders>
- Benjamins, S. (2022b, marraskuuta 7). *The Best Website Builder*.  
<https://www.sitebuilderreport.com/best-website-builder>
- Bermes, B. (2015). *Lean Websites: Because Web Performance Simply Matters*. Sitepoint Pty Ltd.
- Borowska, P. (2013, tammikuuta 2). *What is @font-face and How to Use It in CSS*.  
<https://designmodo.com/font-face/>
- Caddick, R., & Cable, S. (2011). *Communicating the User Experience: A Practical Guide for Creating Useful UX Documentation*. Wiley.
- Canziba, E. (2018). *Hands On UX Design for Developers, Design, Prototype, and Implement Compelling User Experiences From Scratch*. Packt Publishing.
- Carney, L. (2022, lokakuuta 11). *How to Build a Website 2022, A Step by Step Beginner's Guide*. <https://www.websitebuilderexpert.com/building-websites/>
- Cartwright, B. (2021, kesäkuuta 22). *Color Theory 101: A Complete Guide to Color Wheels & Color Schemes*. <https://blog.hubspot.com/marketing/color-theory-design#seven>
- Clean, W3C compliant HTML and CSS*. (n.d.). Webflow. <https://webflow.com/feature/clean-code-standards-compliant-html-and-css>
- Cullen, K. (2012). *Design Elements, Typography Fundamentals*. Quarto Publishing Group USA.
- David, C. (2022, heinäkuuta 7). *Evolution of website building: From hassle to no-code*.  
<https://www.simple.ink/blog/website-building-evolution>
- Dawson, A. (2012). *Future-Proof Web Design, A Survival Guide*. A John Wiley and Sons, Ltd.
- Dean, J. (2018). *Web Programming With HTML5, CSS, and JavaScript*. Jones & Bartlett Learning.
- Eygi, C. (2019, syyskuuta 20). *7 Important Tips for Writing Better CSS*.

- <https://www.freecodecamp.org/news/7-important-tips-for-writing-better-css/>
- Fitzgerald, A. (2022, elokuuta 8). *Domains and Domain Names: What They Are and 10+ Examples*. <https://blog.hubspot.com/website/what-is-a-domain#what-is-a-domain>
- Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3*. Packt Publishing, Limited.
- Gillis, O. (2021, huhtikuuta 13). *7 Rules for Choosing A Website Color Scheme*. <https://elementor.com/blog/website-color-schemes/>
- Graw, M. (2022, helmikuuta 10). *Editor X vs Webflow*. <https://www.creativebloq.com/features/editor-x-vs-webflow>
- Hauschildt, S. (2010). *CMS Made Simple 1. 6: Beginner's Guide*. Packt Publishing, Limited.
- Hernandez, I. (2022, lokakuuta 21). *The Current State of Website Builders (Overview & Comparisons)*. <https://www.dreamhost.com/blog/state-of-website-builders/>
- Heslop, B. (2018, joulukuuta 18). *History of Content Management Systems and Rise of Headless CMS*. <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms/>
- Hilal, H. (2020, helmikuuta 16). *Rem vs em Units in CSS*. <https://medium.com/@hossam.hilal0/rem-vs-em-units-in-css-96d5ac15878e>
- Hogan, B. P. (2009). *Web Design for Developers—A Programmer's Guide to Design Tools and Techniques*. Pragmatic Bookshelf.
- Holcombe, J. (2022, elokuuta 8). *Webflow vs WordPress: Which One is Better for Your Next Site?* <https://kinsta.com/blog/webflow-vs-wordpress/>
- Huusko, A. (2022, lokakuuta 16). *5 Syytä rakentaa verkkosivu Webflow-alustalla 2022*. <https://www.ailecom.fi/blogi/5-syyta-rakentaa-verkkosivu-webflow-alustalla-2022>
- Johnson, B. (n.d.). *7 reasons front-end developers should use Webflow*. <https://webflow.com/blog/webflow-front-end-development>
- Juviler, J. (2022, tammikuuta 4). *The Beginner's Guide to Typography in Web Design*. <https://blog.hubspot.com/website/website-typography>
- Khazanova, A. (2021, tammikuuta 11). *Color Theory Fundamentals Every Web Designer Should Know*. <https://elementor.com/blog/color-theory-web-design/>
- LaGrone, B. (2016). *Web Design Blueprints*. Packt Publishing.
- Leaning, B. (2017, heinäkuuta 28). *Typography Tutorial for Beginners: Everything You Need to Learn Typography Basics*. <https://blog.hubspot.com/marketing/typography-terms-introduction#whatistypography>
- Low Fidelity Wireframes vs High Fidelity Wireframes. (n.d.). *mentormate.com*. <https://mentormate.com/blog/low-fidelity-wireframes-vs-high-fidelity-wireframes/>
- Marcotte, E. (2014). *Responsive Web Design*. Jeffrey Zeldman.

- McKay, L. (2021, elokuuta 6). *What is a Monospaced Font?*  
<https://creativemarket.com/blog/what-is-a-monospaced-font>
- McSweeney, D. (2022, tammikuuta 25). *Revealed: Which CMS Is Best For SEO In 2022?*  
<https://www.seobility.net/en/blog/best-cms-for-seo/>
- Miksi valita Headless CMS?* (2022, maaliskuuta 23). <https://nitro.fi/blogi/miksi-valita-headless-cms>
- Newcomer, C. (2022, toukokuuta 10). *Webflow vs WordPress: Which Should You Use to Build Your Website?* <https://blog.hubspot.com/website/webflow-vs-wordpress#pros-and-cons>
- O’Connell, B. (2022, elokuuta 5). *A Beginner’s Guide to the Headless CMS.*  
<https://www.fool.com/the-ascent/small-business/cms/articles/headless-cms/>
- Osman, M. (2021, marraskuuta 2). *The Differences in Wireframe Fidelity: From Low to High Fidelity Wireframes.* <https://blog.hubspot.com/website/high-fidelity-wireframe>
- Paterson, K. (2022, lokakuuta 21). *Webflow vs. WordPress: Which is better for your website? [2023].* <https://zapier.com/blog/webflow-vs-wordpress/>
- Pawlak, A. (2021, elokuuta 3). *Website Builder Vs. CMS? What’s Better?*  
<https://dotinum.com/blog/website-builder-vs-cms-whats-better/>
- Ramoser, A. (2022, helmikuuta 24). *Tested: Is Webflow Good for SEO in 2022?*  
<https://www.seobility.net/en/blog/webflow-seo/>
- Robbins, J. (2012). *Learning Web Design: A Beginner’s Guide to Html, Css, Javascript, and Web Graphics.* O’Reilly Media, Inc.
- Rosenfeld, L., Morville, P., & Arango, J. (2015). *Information Architecture For the Web and Beyond.* O’Reilly Media, Inc.
- Steele, H. (2022, syyskuuta 22). *Webflow vs WordPress: Which is Better?*  
<https://superbwebsitebuilders.com/webflow-vs-wordpress/>
- Suzanne, M. (2022, toukokuuta 23). *Fun with Viewport Units.* <https://css-tricks.com/fun-viewport-units/>
- Thibodeau, T. (2022, huhtikuuta 29). *WordPress.com vs WordPress.org: What’s the Difference?* <https://wordpress.com/go/website-building/wordpress-com-vs-wordpress-org/>
- Tidwell, J., Brewer, C., & Valencia, A. (2020). *Designing Interfaces, Patterns for Effective Interaction Design.* O’Reilly Media, Inc.
- Usage statistics and market share of Webflow.* (2022, marraskuuta 27). w3techs.  
<https://w3techs.com/technologies/details/cm-webflow>
- Usage statistics of content management systems.* (2022, kesäkuuta 27). w3techs.  
[https://w3techs.com/technologies/overview/content\\_management](https://w3techs.com/technologies/overview/content_management)
- w3schools. (n.d.). [https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)

Walsh, M. (2022, marraskuuta 22). *Webflow Review—The Key Pros and Cons*.

<https://www.stylefactoryproductions.com/blog/webflow-review>

*Web.dev.* (n.d.). Web.Dev. <https://web.dev/learn/css/box-model/>

*Webflow.* (n.d.). Webflow pricing. Haettu 23.11.2022 osoitteesta <https://webflow.com/pricing>

*Webflow.* (n.d.). Webflow vs Wordpress. <https://webflow.com/vs/wordpress>

*Webflow web hosting: Fast, reliable service.* (n.d.). <https://webflow.com/hosting>

Williams, J. M. (ei pvm.-a). *The Modern Web Design Process*.

Williams, J. M. (ei pvm.-b). *Web Design 101*.

## **Liite 1: Aineistonhallintasuunnitelma**

### **Kehitysprojekti:**

Kehitysprojektin tulokset löytyvät Webflow'n AWS-palvelimelta. Muu aineisto sijaitsee tietokoneeni kovalevyllä sekä ulkoisella kovalevyllä. Aineistoa tullaan säilyttämään vuosi opinnäytetyön valmistumisen jälkeen. Aineisto varten ei tulla keräämään henkilötietoja, joten tietosuojasta ei tarvitse tässä yhteydessä huolehtia

