Veikon kotisivuopas

Internetin käsikirja teknonaturalistille

1 Alkusanat

Käsissäsi oleva opas on eräänlainen kunnianosoitus internetille, joka on miltei kadonnut vuoden 2023 lopulla, jolloin kirjoitan näitä alkusanoja, internetin, jossa ei ollut Twitteriä tai TikTokia. Sen sijaan henkilökohtaiset asiat piti jakaa halutessaan mitä erikoisimmille kotisivuille. Osa sivuista oli hyvinkin näyttäviä, osa lukukelvottomia, mutta silti jokaisessa niistä oli enemmän persoonallisuutta kuin yhdessäkään Instagram-profiilissa tai YouTube Shortsissa.

Oli toki aika, jolloin sosiaalinen media oli avoin ja mielenkiintoinen. Jossa lähes jokainen meistä jakoi aamupala päivityksiään omalla nimellään ja naamallaan. Kaukana ovat ne ajat nyt. Aina vain harvempi päivittää kuulumisiaan aktiivisesti Facebook-seinälleen ja uuden sisällön tuottavat suurimmalta osin ammattimaiset sisällöntuottajat. Aiemmin niin osallistavasta internetistä on tullut vain seuraava passiiviseen kuluttamiseen rohkaiseva *time sink*.

Unelmanani onkin elvyttää tämän oppaan avulla kotisivu kulttuuria. Tai ainakin tarjota hetken inspiraatioita ja näpertelyn iloa algoritmeilla hypermodattujen syötteiden turruttamille kanssaihmisille.

Kotisivun kehittäminen onkin siis sitä: näpertelyä. Se vie aikaa ja vaatii kärsivällisyyttä sekä visiota. Kannattaakin siis olla itselleen armollinen eikä heti pyrkiä täydellisyyteen. Tärkeintä on, että kotisivusi on nimen omaan sinun sivusi. Sen ei kuulukaan olla heti valmis vaan kehittyä ajan kanssa.

Sen ei tarvitse olla vain käyntikortti tai ilmoitustaulu vaan oma tilasi valtavan internetin meressä. Säilytä siellä listaa vinyylilevyistäsi tai vaikka tietoa hiomakoneesi laikan koosta. Ethän voi koskaan tietää milloin rautakaupassa tulee vastaan tarjous-laikkoja ja haluat varmistaa niiden yhteensopivuuden. Kappas, tietohan löytyy näppärästi internetistä!

Sinä päätät mitä kotisivuillasi teet. Kukaan ei tule kysymättä haastamaan riitaa tai seuraa toimintaasi vain myydäkseen sen eteenpäin mainostajille. Todennäköisesti kukaan ei edes tiedä sivusi olemassa olosta, ellet siitä erikseen mainitse. Ja hyvä niin.

Kotisivun kehitystä kannattaa ajatella kuin mitä tahansa käsityötä. Käytät siihen hetken sieltä, toisen täältä ja pian huomaatkin saaneesi aikaan jotakin ainutlaatuista, jotakin mistä olla ylpeä.

On aika ottaa internet takaisin!

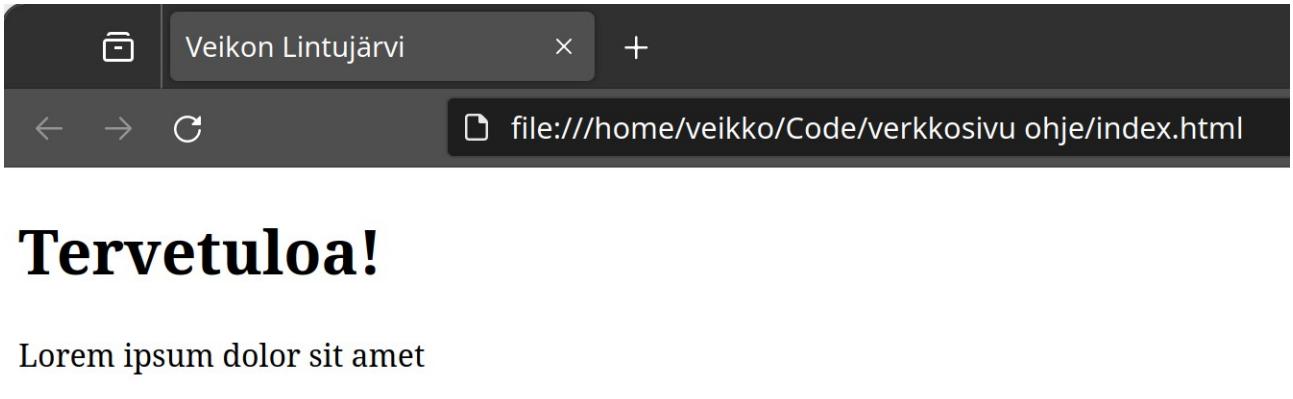
Sisällysluettelo

2 Mitä verkkosivut ovat?

Jokainen verkkosivu rakentuu HTML (Hyper Text Markup Language) merkintä kielen varaan. Sen tarkoituksena on kertoa mitä haluat näyttää selaimelle sen avatessa sivusi. Esimerkiksi alla seuraava *Koodi 1* muodostaa *Kuva 1*:n mukaisen näkymän.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Veikon Lintujärvi</title>
    </head>
    <body>
        <h1>Tervetuloa!</h1>
        <p>Lorem ipsum dolor sit amet</p>
    </body>
</html>
```

Koodi 1: Yksinkertainen HTML-koodin pätkä



Kuva 1: Yksinkertainen HTML dokumentti luettuna paikallisesti

Selaimen osoitepalkissa (*Kuva 1*) oleva teksti *file:///home/veikko/Code/verkkosivu ohje/index.html* tarkoittaa, että olen avannut koodin tiedostosta *index.html* tietokoneellani hakemistossa /*home/veikko/Code/verkkosivu ohje*. Normaalisti koodi kuitenkin haetaan Internetin välityksellä joltakin toiselta tietokoneelta esimerkiksi osoitteesta *veikko.lintujarvi.fi*. Tämä tietenkin vaatii, että osoite on asetettu osoittamaan kyseiseen tietokoneeseen, ja että siellä on käynnissä sovellus, joka jakaa valitsemaasi *index.html* tiedostoa.

Selatessamme internetiä availemme siis tiedostoja siihen tarkoitetulla sovelluksella, joka sitten piirtää ne nähtävillemme niiden jakajan haluamalla tavalla. Tämä ei oleellisesti eroa mistään muusta tietokoneen käytöstä oikeastaan muuten kuin, että tiedostot usein sijaitsevat jollain toisella koneella.

2.1 HTML-koodin ymmärtäminen

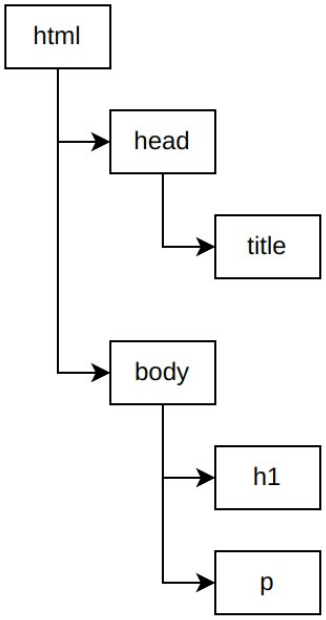
Kaikki grafiikka Internetissä ei ole HTML-koodin sisällä. Voit halutessasi laittaa kuvan tai vaikka PDF-tiedoston jakoon valitsemaasi verkko-osoitteeseen. Tällöin osoitteessa ei kuitenkaan ole muuta kuin tämä yksinäinen tiedosto ilman sen kummempaa logiikkaa tai yhteyttä muihin tiedostoihin tai sivuihin. HTML:n hienous piileekin sen kyvyssä jäsentää ja linkata tietoa paikasta toiseen, tietenkin grafiikan piirtämisen lisäksi.

2.2 Elementit

HTML:n kyky jäsentää tarkoittaa, että kaikki HTML-dokumentin sisältämä tieto on säilötty elementtien sisälle. Yksittäin elementti koostuu aloitusmerkistä (esim. *<h1>*) ja lopetusmerkistä (esim. *</h1>*) Näiden kahden merkin (eng. *tag*) välissä voi olla tekstiä tai muita elementtejä.

Elementtien laittaminen toistensa sisään luo dokumenttiin eräänlaisen puurakenteen. Aiempi HTML koodiesimerkki (*Koodi 1*) näyttäisi puurakenteena *Kaavan 1* mukaiselta. Näin elementit eivät ole yhdessä kasassa vaan niiden sijainti puussa kertoo niiden paikan verkkosivulla.

Elementtien aloitusmerkkeihin voi myös lisätä ominaisuuksia (eng. *properties*) esimerkiksi näin *<h1 id=”main-title”>*, jossa ominaisuus *id*:n arvoksi on asettu *main-title*. Mahdollisia ominaisuuksia on elementistä riippuen iso valikoima, joista joihinkin tulemme palaamaan tässä oppaassa myöhemmin. Niiden olemassa olo on kuitenkin hyvä pitää jatkossa mielessä.



Kaava 1: Yksinkertainen HTML-dokumentti esitetynä puurakenteessa

2.3 Linkkaus

“HTML” lyhenteen “HT” *HyperText* sisältää yhden sen oleellisimmista oivalluksista. Dokumentti voi sisältää linkkauksia (ks. Hyperlink) toisiin dokumentteihin, joiden ei edes tarvitse olla HTML-dokumenttejä. Tämä helpottaa suuresti isomman tietomäärän hallintaa, mutta linkeistä lisää myöhemmin.

2.4 Grafiikka

Selain käyttää oletuksena tiettyjä värijä ja fonteja, mutta toisella oleellisella koodityökalulla pääsemme muuttamaan näitä ja oikeastaan mitä tahansa muutakin graafista ominaisuutta sivulla. Tästä lisää osiossa *Tyylittelyä CSS:llä*.

2.5 Pari sääntöä

Kielenä HTML on hyvin anteeksiantavainen ja selaimet tulkisevat miltei minkä mallista HTML:ää jotakuinkin halutusti. Ne harvat asia, jotka vaativat tarkkuutta ovat *<!DOCTYPE html>* ja pari erityis elementtiä, joista alla lisää.

2.5.1 DOCTYPE

<!DOCTYPE html> on HTML-merkki, joka kertoo selaimelle tiedoston olevan HTML koodia (huom. *.html* tai mikään muukaan pääte ei lähtökohtaissesti kerro mitään itse tiedoston sisällöstä). Tämä tulee siis olla jokaisen HTML-dokumentin alussa.

2.5.2 Html, head ja body

Lisäksi kaiken HTML-koodin tulee olla *html* elementin sisällä, joka taas jaottuu kahteen aliosioon: *head* ja *body*. *Head*issa sijaitsee ns. dokumentin metadata, joista tärkeimpiä ovat sivun otsikko *title* ja tyylit *style*. *Body* taas sisältää varsinaisen esitettävän tekstin ja linkit.

3 Tyylittely CSS:llä

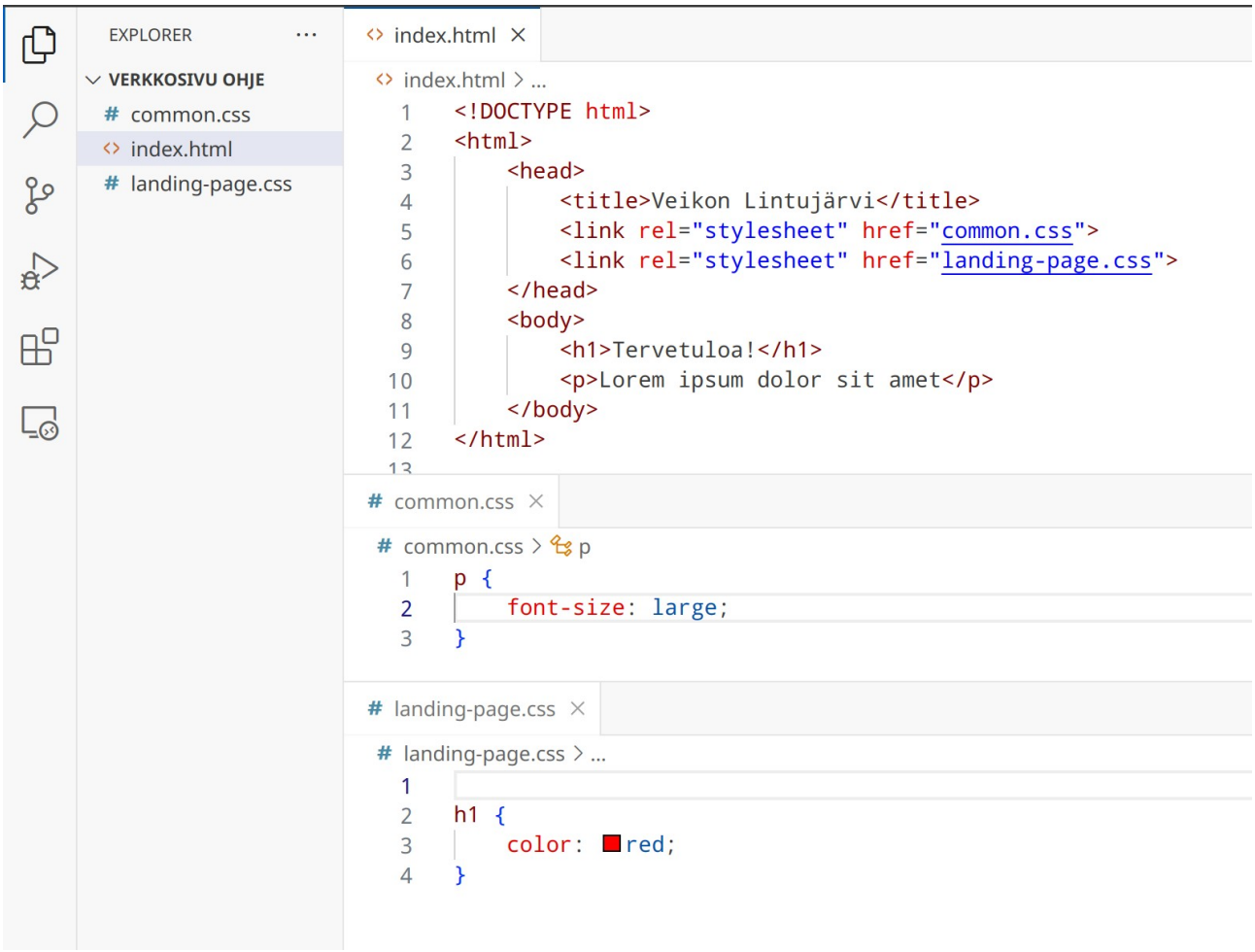
CSS eli *Cascade Styling Sheet* on nimensä mukaisesti ylhäältä alaspäin luettava lista tyylittelyjä. Listassa siis alempana olevat ylikirjoittavat niitä ennen tulleet tyylit mikäli ne ovat ristiriidassa (ks. *Koodi 2*). Tyylit kuuluvat *style* elementtiin HTML-dokumentin *head* osiossa.

index.html > ...

```
1   <!DOCTYPE html>
2   <html>
3       <head>
4           <title>Veikon Lintujärvi</title>
5           <style>
6               h1 {
7                   color: red;
8               }
9
10              h1 {
11                  color: green;
12              }
13          </style>
14      </head>
15      <body>
16          <h1>Tervetuloa!</h1>
17          <p>Lorem ipsum dolor sit amet</p>
18      </body>
19  </html>
20
```

Koodi 2: CSS-koodia style-elementin sisällä, kyseinen koodi muuttaa kaikki dokumentin h1 elementit vihreiksi

Tämän lisäksi tyylittelyjä voi asettaa suoraan elementin ominaisuuteen nimeltä *style* tai linkata toisesta tiedosta. Tyylien siirtäminen erillisiin tiedostoihin onkin suositeltavaa, sillä tällöin voit jaotella yleiset koko sivuston kattavat tyylit omassa tiedostossaan ja sivu kohtaiset omassaan kuten *Koodissa 3*.



Koodi 3: CSS-koodi siirrettynä erillisiin tiedostoihin selkeyden vuoksi. Yleisille tyyleille on oma tiedostonsa ja sivukohtaisille omansa

3.1 Valitsimet

Mitä jos haluammekin tyylitellä jonkin tietyn *h1* elementin niiden kaikkien sijasta. Tällöin pitää ottaa avuksi valitsimet. Valitsimia on pitkä lista, mutta esittelen alla kaksi käytetyintä, joilla pärjää suurimman osan ajasta:

3.1.1 Luokat ja tunnisteet

Luokkien (koodissa *class*) ja tunnisteiden (koodissa *id*) ero on siinä, että luokilla valitaan useampaa ja tunnisteilla yhtä tiettyä elementtiä. Kuten HTML myös CSS on monen asian suhteen salliva eikä näiden kahden käyttäminen sekä yksittäisten että useiden elementtien valitsemiseen riko sivustoa, mutta tämä erottelu on selkeyden vuoksi suositeltavaa.

Luokkien ja tunnisteiden asettaminen tapahtuu elementin *id* ja *class* ominaisuuksissa (ks. *Koodi 4*).

```
<h1 id="main-title">Tervetuloa!</h1>
<p class="info-text">Lorem ipsum dolor sit amet</p>
<p class="info-text">consectetur adipiscing elit</p>
```

Koodi 4: Otsikko elementin (<h1>) tunnisteeksi on asetettu "main-title" ja molempien kappaleiden (<p>) luokaksi "info-text"

Hauska knoppi liittyen CSS-tunnisteisiin on, että voit lisätä *#tunniste* minkä tahansa hyperlinkin perään (esim. v*eikko.lintujarvi.fi#main-title*) ja selain kelaa sivun yläreunan sijasta sen elementin kohdalle, jossa tuo tunniste on käytössä.

Tunniste valitaan myös CSS-koodin puolella laittamalla nimen eteen “#”. Luokkat taas valitaan pisteellä (ks. *Koodi 5*).

```
#main-title {
    color: red;
}
.info-text {
    font-size: large;
}
```

Koodi 5: Tässä CSS-koodissa "main-title" tunnisteen kirjasinväri on asetettu punaiseksi ja "info-text" luokan kirjasin koko suureksi

Kuten mainittua jo aiemmin, valitsimia on paljon lisää, mutta niistä ja monesta muusta käytännönläheisemmästä seuraavissa osioissa.

3.2 Tekstieditori

Koodaaminen tapahtuu pitkälti tekstieditorissa tai IDEssä (Integrated Development Environment). Jollet ole varma mitä käyttää (huom. Word ei ole tekstieditori), suosittelen lataamaan *Visual Studio Code* nimisen ohjelmiston. Se löytyy osoitteesta:

`https://code.visualstudio.com`

Seuraavat kehitysympäristön pystytys esimerkit tulevat olemaan *Visual Studio Code* spesifejä.

3.3 Yhteyden ottaminen palvelimeen

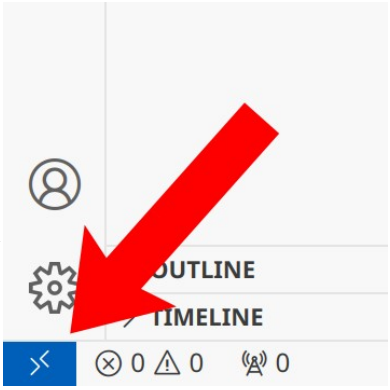
Seuraavassa käyn läpi helpoimman tavan päästä käsiksi tarjoamaani kehitysympäristöön.

Saattaa olla, että joudut yrittämään tätä useamman kerran, joten laitan lyhyen muistilistan tähän alle, johon voit palata tarvittaessa uudelleen:

1. Yhteyden lisääminen	Lisää palvelinyhteys VS Coden lisäosan valikkoon
2. Yhteydenlisäyskomento	`ssh käyttätunnuksesi@lintujarvi.fi -A`
3. Yhteyslisäosan uudelleen avaaminen	Löytyy samasta paikasta kuin yhteyttä lisätessä
4. Uusi ikkuna avautuu	
5. Hyväksy uusi yhteys	“has fingerprin SHA...” jne. Tarvitaan vain ensimmäisellä kirjautumisella
6. Syötä salasana	
7. Avaa kotisivu hakemisto	“Explorer” → “Open Folder” → “/home/käyttäjätunnus/kotisivu”
8. Syötä salasana	
9. Muokkaa tiedostoja	

Jotta voimme lähteä rakentamaan verkkosivua, meidän pitää päästä käsiksi tietokoneeseen, josta haluamme jakaa verkkosivun HTML-dokumenttia. Tämän tyyppisiä tietokoneita kutsutaan usein *palvelimiksi* ja sillä nimellä aion itsekin sitä tästä eteenpäin kutsua.

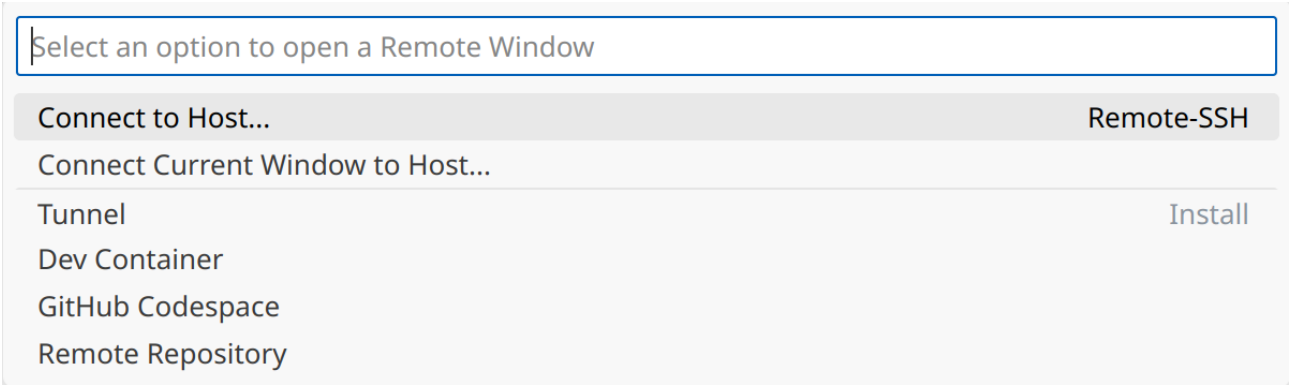
Helpoin tapa yhdistää palvelimmelle on avata *SSH*-yhteys (SSH tulee sanoista Secure Shell). SSH:n avulla voimme luoda suojatun kanavan palvelimelle. Tämä tapahtuu helposti *VS Codessa* olevalla lisäosalla.



Kuva 2: VS Coden palvelinyhteyslisäosan pikanäppäin

Klikkaa *Kuvan 2* osoittamaa painiketta *Visual Studio Coden* vasemmassa alakulmassa. Tämän pitäisi avata *Kuvan 3* mukainen valikko ikkunan ylälaitaan. Valitse tästä “Connect to Host…” ja seuraavasta “ + Add New Host…”. Kirjoita seuraavaan syötteeseen seuraava komento (huom. Korvaa “käyttäjätunnuksesi” Veikolta saamallasi käyttäjätunnuksella):

```
ssh käyttätunnuksesi@lintujarvi.fi -A
```

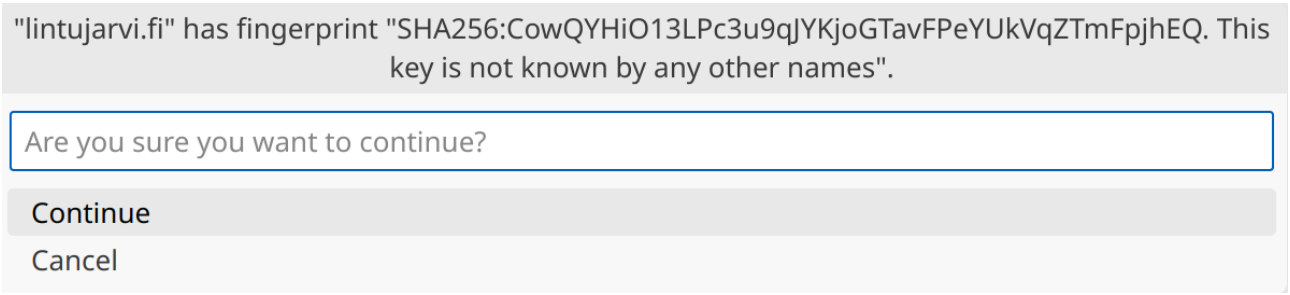


Kuva 3: Yhteystyyppivalikko VS Codessa

Jos *VS Code* vielä tämän jälkeen kysyy “Select SSH configuration file to update…” vastaa ensimmäinen vaihtoehto valikosta.

Nyt olet lisännyt *VS Coden* etäyhteyslistalle palvelimen niin, että voit jatkossa kirjautua sinne vain parilla klikkauksella.

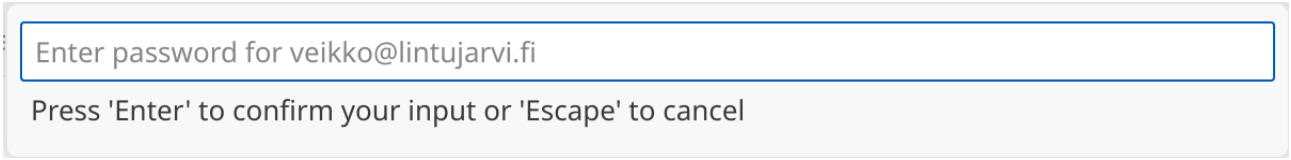
Jotta pääsemme kirjautumaan palvelimelle yhteyden lisäämisen jälkeen meidän pitää avata *Kuvan 3* valikko jälkeen uudelleen *Kuvan 2* osoittamasta napista, valita jälleen “Connect to Host…” ja tällä kertaa valita listalle ilmestynyt juuri luomamme “lintujarvi.fi”. Valitse se ja *VS Code* avaa uuden ikkunan, jossa se kysyy suurinpiirtein seuraavaa:



Kuva 4: SSH pitää yllä tunnettujen etäyhteyksien listaa ja varoittaa, mikäli yhteys jota koetat muodostaa ei ole entuudestaan tuttu

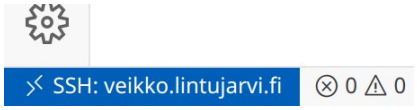
Vastaa tähän “Continue”. Kyseinen tiedustelu tulee vain ensimmäisellä kirjautumiskerralla.

Viimein seuraavassa kohdassa sinun tulee syöttää salasana, jonka olet saanut Veikolta *Kuvan 5* mukaiseen valikkoon.

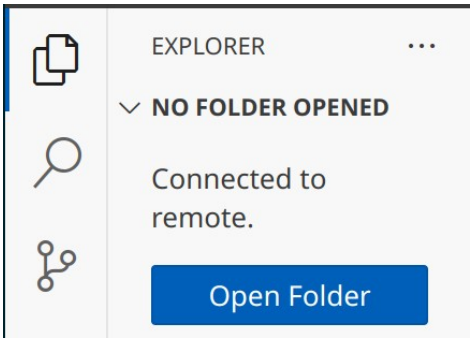


Kuva 5: Salasanan tiedustelu (huom. "veikko" on mikä tahansa käyttäjätunnus, jolla koetat kirjautua)

Nyt olet muodostanut onnistuneesti yhteyden palvelimelle, josta tulevaa kotisivuasi tullaan jakelemaan. Voit varmistaa, että yhteys toimii katsomalla, että vasemmassa alakulmassa lukee jotakuinkin näin:



Viimeiseksi sinun tulee valita kotisivusi hakemisto. Avaa oikeasta yläkulmasta *VS Coden Explorer Kuvan 6* mukaisesti. Tämän jälkeen klikkaa “Open Folder” painiketta, valitse itsesi valikosta “kotisivu” hakemisto ja klikkaa “Ok”. Joudut syöttämään salasanasi vielä kerran. Jos sinulta kysytään “Do you trust the authors of the files in this folder?” voit hyvillä



Kuva 6: VS Coden Explorer, näppärä lisäosa hakemistojen ja tiedostojen selaamiseen

mielin vastata “Yes, I trust the authors”. Tämän jälkeen oletkin onnistuneesti perillä. Olen lisännyt kansioon verkkosivupohjan ja voitkin viimein alkaa rakentamaan kotisivuasi!

3.4 Kotisivun kehitys

Kaikki *kotisivu* kansiosi tiedostoihin tekemäsi muutokset tulevat välittömästi näkyville kotisivuillesi, joten onkin varmasti helpointa pitää toisessa ikkunassa auki koodiasi (johon otimme yhteyden edellisessä osiossa) *VS Codessa* ja toisessa selainta, johon olet avannut kotisivusi. Näin näet saman tien, miltä tekemäsi koodimuutokset näyttävät kotisivuilla.

3.4.1 Developer Tools

Yksi näppärä työkalu kaikissa moderneissa selaimissa on *Developer Tools*. Siihen pääset käsiksi joko oikealla hiiren näppäimellä, klikkaamalla jotain kohtaa sivusta ja valitsemalla “Inspect” tai painalla F12 näppäimistöstäsi. Näin saat näkyville koodisi suoraan selaimeen ja voit jopa muokata sitä. Muutokset, joita teet eivät kuitenkaan tallennu ja sivu palaakin ennalleen kun lataat sen uudelleen. *Developer Tools* on kuitenkin varsin kätevä apu esimerkiksi CSS-tyylittelyitä hienosäätäessä.

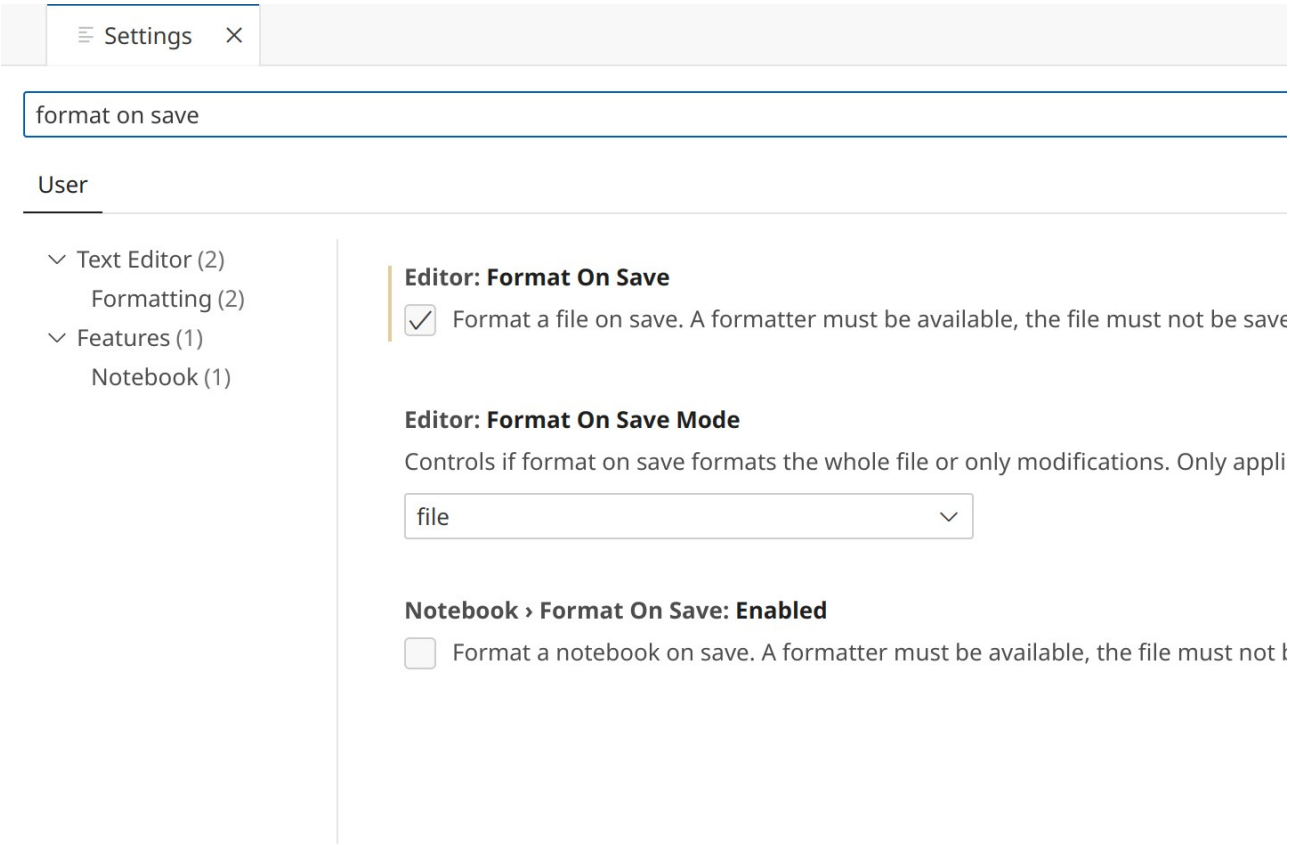
Kuva 7: Developer toolsin saa auki painamalla F12:sta. Voit nähdä elementin margin ja padding ominaisuuden näytöllä vielämällä kursorin elementin päälle. Eri tyylejä voi myös kokeilla täältä.

3.4.2 VS Coden käyttö

Pari hyödyllistä asiaa *VS Codesta* ennen kuin alat käyttämään sitä.

3.4.2.1 Formatointi

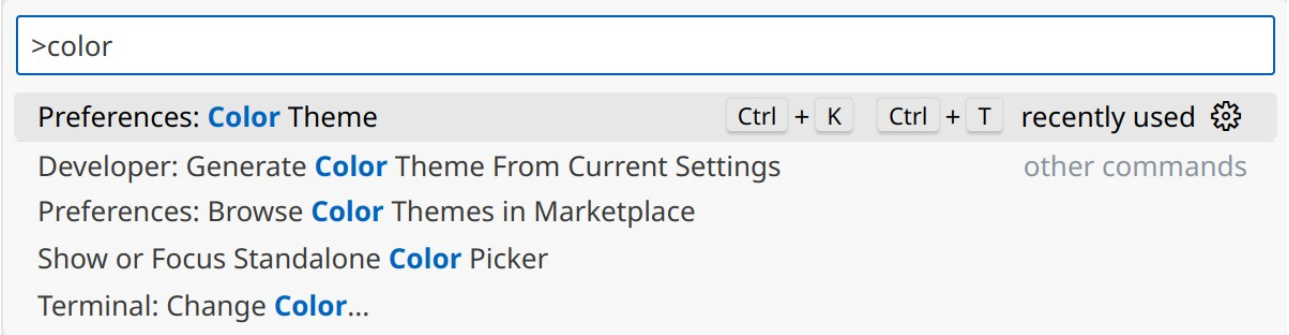
Jotta koodisi pysyy helposti järjestyksessä kannattaa ottaa automaattinen formatointi käyttöön asetuksista *Kuvan 8* mukaisesti.



Kuva 8: "Editor: Format on save" kannattaa laittaa päälle asetuksista. Näin ollen aina kun tallennat (Crtl+S) myös koodisi siistitään automaattisesti.

3.4.2.2 Ctrl+Shift+P

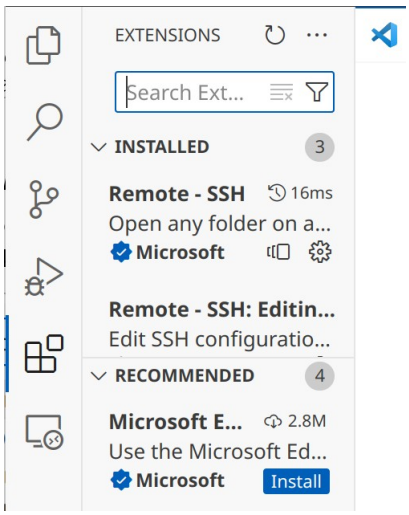
Näppäinyhdistemä *Crtl+Shift+P* avaa pikatoimintovalikon, jossa on vapaa sanahaku. Tästä valikosta löytyy paljon kaikenlaista hyödyllistä, kuten väri-teeman valinta (ks. *Kuva 9*).



Kuva 9: Pikatoimintovalikosta voit muuttaa väriteemaa. Lisää väriteemoja voi ladata Extensions välilehdeltä.

3.4.2.3 Extensions

Extensions eli lisäosat löytyvät *VS Coden* sivupalkista (ks. *Kuva 10*). En osaa näin alkuun suositellaoikeastaan mitään tiettyä, mutta lisäosa valikoimaan kannattaa ihan mielenkiinnosta selailla. Kaikki lisäosat ovat ilmaisia.



Kuva 10: Lisäosavalikko (eng. Extensions). Täältä löytyy koodaustyökaluja melkein mihin vain tarpeeseen.

3.4.2.4 Snippets

Snippets eli leikkeet ovat näppäriä koodinpätkiä, joita voit lisätä helposti editoriisi painamalla *Ctrl+Space*, valitsemalla leike ja painamalla *Enter*. Erityisen hyödyllinen meidän tapauksessamme on “HTML Sample” niminen leike. Tämän avulla sinun ei tarvitse joka kerta kirjoittaa kaikkea HTML:n vaatimaa koodia uudestaan (ks. Osio *2.5 Pari sääntöä*).

Kuva 11: Generoi usein käytettyä koodia Ctrl+Space pikakomennolla. Voit ladata lisää leikkeitä Extensions valikosta.

4 Koodausvinkkejä

Nyt kun olet saanut kehitysympäristön valmiiksi. Tässä osiossa annan neuvoja ja vinkkejä koodaamiseen.

4.1 W3Schools

W3Schools on oma ensisijainen ja oikeastaan ainoa lähteeni mitä tulee HTML tai CSS ongelmiin.

https://www.w3schools.com/

Mikäli haluat tietää jostakin HTML-elementistä tai CSS-tyylistä lisää tämä on paras paikka saada nopea vastaus.

4.2 Tärkeimpiä HTML-elementtejä

HTML-elementtejä on kaikkiaan kaikkiaan yli sata, mutta alla niistä muutama tärkein.

4.2.1 Kuvat

Kuvia lisätään <*img*> merkeillä. Jokaisen *img* elementin tulee lisältää *src* ja *alt* ominaisuudet. Kuvatiedostoon osoitetaan esimerkiksi *src=”kuva.jpg”*, jonka mukaan kuvatiedoisto sijaitsee samassa kansiossa. Mikäli haluat laittaa kuiva tidostot erilliseen kansioonsa (mikä on erittäin suositeltavaa), voit osoittaa niihin esimerkiksi näin *src=”kuvat/kuva.jpg”*.

```
<img src=”hastur.png alt=”Leikkivä kissa”>
```

Alt ominaisuus antaa kuvalle meta-selitteen. Tämä ominaisuus on saavutettavuus syistä tarpeellinen (ks. *4.2.9 Saavutettavuus*). *Alt*-tekstin tulee kertoa mitä kuvassa on sisältämättä kuitenkaan sanaa “kuva”.

Huono *Alt*-teksti: “Kuva mustasta kissasta”

Hyvä *Alt*-teskti: “Musta kissa”

Kuva elementti ei vaadi päätemerkkiä (esim. <*/img*>) kuten muut HTML-elementit.

4.2.2 Sivuston otsikko

Sivuston otsikon voi asettaa <*title*> elementillä HTML-dokumentin <*head*>-osiossa. Sen sisältämä teksti näkyy sivun nimenä hakukoneissa ja selaimen välilehdillä.

4.2.3 Otsikot tekstissä

Eri tason otsikoita voi lisätä tekstiin elemeillä <*h(jokin numero)*> esim. <*h3*>. Mitä suurempi numero, sitä pienempi otsikko.

4.2.4 Div

Div tulee sanasta “division”. *Diviä* käytetään silloin kun sivun sisältö halutaan jakaa usempaan osaan kuitenkin ja mitään semanttisista elementeistä (ks. *4.2.5 Semanttiset elementit*) ei voida käyttää.

4.2.5 Semanttiset elementit

Semanttisillä elementeillä ei ole sinänsä toiminnallisia ominaisuuksi kuten vaikkapa *<img>:*lla, vaan niitä käytetään koodin luettavuuden parantamiseksi. Näihin kuuluvat muun muassa *<nav>*, *<header>* ja *<section>*. W3Schoolsissa on erinomainen artikkeli tästä aiheesta:

https://www.w3schools.com/html/html5_semantic_elements.asp

Semanttisten elementtien käyttö erottaa HTML-amatöörit HTML-mestereista.

4.2.6 Kappaleet

Kaiken perustekstin tulisi olla *p* eli “paragraph” elementtien sisällä. Oikeastaan ainoat sallitut paikat tekstille on *<p>*, *<a>* (ks. *4.2.7 Linkit*), *<h>* (ks. *4.2.3 Otsikot tekstissä*) ja *<li>* (ks. *4.2.8 Listat*) elementtien sisällä.

4.2.7 Linkit

Linkkejä luodaan käyttämällä *<a>* elementtiä. *A* tulee sanasta “anchor”, joka todennäköisesti on peräisin sanomalehtien taitosta (kuten niin moni muukin asia HTML:ssä). Linkki vaatii *href* ominaisuuden ja useinm miten joko alielementtejä ja tai tekstin.

Tämä elementti:

```
<a href=”https://veikko.lintujarvi.fi”>Veikon kotisivu</a>
```

tuottaa seuraavanlaisen linkin:

Veikon kotisivu

Linkeillä voi näppärästi luoda navigointipalkin kotisivuillesi.

```
<nav>
      <ul>
            <li><a href="/">Etusivu</a></li>
            <li><a href="linux.html">Linux</a></li>
            <li><a href="warhammer.html">Warhammer</a></li>
            <li><a href="tietokoneet.html">Tietokoneet</a></li>
      </ul>
</nav>
```

Huomaa, että saat linkin etusivullesi asettamalla *hrefin* arvoksi “ / ”.

4.2.8 Listat

Listat koostuvat varsinaisesta lista elementistä *<ul>* tai *<ol>* ja näiden sisällä olevista lista riveistä *<li>*.

Listoja tulisi käyttää aina kun listaat asioita, halusit ne alekkain tai viereikkäin. Onkin hieman hämäävää, että listat piirretään oletuksena alekkain.

```
<ul>
      <li>Tietokonepelaaminen</li>
      <li>Miniatyyrien maalaaminen</li>
      <li>Pyöräily</li>
      <li>Tietokilpailujen järjestäminen</li>
      <li>Koodaaminen ja tietokoneiden ropailu</li>
</ul>
```

tuottaa ilman CSS-tyylittelyjä seuraavan lopputuloksen

- Tietokonepelaaminen
- Miniatyyrien maalaaminen
- Pyöräily
- Tietokilpailujen järjestäminen
- Koodaaminen ja tietokoneiden ropailu

Tarkemmin listoihin kannattaa jälleen tutustua W3Schoolsin artikkelin kautta:

https://www.w3schools.com/html/html_lists.asp

4.2.9 Saavutettavuus

Verkkosivujen tulisi olla mahdollisimman monen käytettävissä. Osa ihmisistä on väliaikaisesti tai pysyvästi näkö-, liikunta- tai kuulokyvyttömiä. Esimerkkinä autoa ajavan käyttäjän tulee saada tarvittava informaatio sivulta mahdollisimman nopeasti. Sokea taas saattaa käyttää ruudunlukuohjelmistoa internetin selaamiseen, joka kirjaimellisesti lukee koodin käyttäjälle soveltuvin osin.

Tämän vuoksi myös verkkosivuja tulee kehittää niin, että kaikki tarvittava tieto on esitettynä selkeästi, mutta myös järkevästi tekstimuodossa.

Alla lista muutamasta helposta tavasta parantaa sivusi saavutettavuutta:

1. Lisää <*img*> elementteihin *alt* ominaisuus (ks. *4.2.1 Kuvat*)
2. Käytä semanttisia elementtejä (ks. *4.2.5 Semanttiset elementit*)
3. Vältä *divejä* (ks. *4.2.4 Div*)
4. Valitessasi värejä, suosi konstrastia

W3Schoolsilla on aiheesta erinomainen tutoriaali:

https://www.w3schools.com/accessibility/

Usein saavutettavan sivun koodi
on myös ymmärrettävää koodia.

4.3 Tyylittelyvinkit

CSS on ehdottomasti haastavin osa verkkosivujen kehittämistä ja vaikka olenkin käyttänyt itse sitä jo vuosia, jotkin asiat saattavat edelleen tuottaa hankaluuksi. Ole siis kärsivällinen ja hae apua internetin keskustelupalstoilta, jos jäät jumiin johonkin. Todennäköisesti et ole ongelmasi kanssa yksin.

Seuraavaksi listaan muutamia asioita, joita olisin itse toivonut tietäväni kun aloitin verkkosivujen kehittämisen.

4.3.1 Selaimen developer tools

Mainitsin *Developer Toolsin* jo aiemmin, mutta en voi tarpeeksi painottaa kuinka näppärä työkalu se on. Voit siis avata verkkosivusi ja klikata hiiren oikealla painikkeella mitä tahansa kohtaa sivusta, valita “Inspect element” tms. ja selain avaa *Developers Toolsin* siitä kohti HTML:ään. Tämän jälkeen voit muokata kyseisen elementin CSS-tyylittelyjä näkymän samalla muuttuessa. Kun olet muokannut tyylit sopiviksi voit kopioda ne tekstieditoriisi ja tallentaa.

Kehotankin opettelemaan heti alkuun *Developer Toolsin* käytön. Vaikka se alkuun vaikuttaakin monimutkaiselta, säästät jatkossa paljon vaivaa.

4.3.2 Elementin koko

Jokaisella elementillä on *width* ja *height* ominaisuutensa. Voit käyttää näitä asettamaan esimerkiksi kuvasi haluamasi kokoiseksi.

Näitä useammin neuvoisin kuitenkin käyttämään ehdollisia mittoja kuten *max-width* tai *min-height*. Tällöin voit varautua eri kokoisiin näyttöihin (tietokoneen näyttö vs. kännykkä).

4.3.3 Mittayksiköt

Perusmitta CSS:ssä on *px* eli pikseli. Nimestään huolimatta pikseli ei aina ole oikeasti pikselin kokoinen vaan saattaa vaihdella laitekohtaisesti. Pikseleitä kannattaakin käyttää harkiten.

Pikseliä parempia mittayksiköitä ovat *rem*, *em* ja *%*. Nämä ovat niin sanottuja *suhteellisia mittayksiköitä*. Ne ovat siis riippuvaisia jostakin ulkoisesta tekijästä kuten kirjasinkoosta. Täten ne skaalautuvat paremmin kokoisilla näytöillä.

Nyrkkisääntönä voisikin sanoa, että *margin* ja *padding* (ks. *4.3.4 Marginaalit*) sekä kirjasin koot kannattaa asettaa *rem* yksiköllä ja vain hyvin pienet yksityiskohdat pikseleillä.

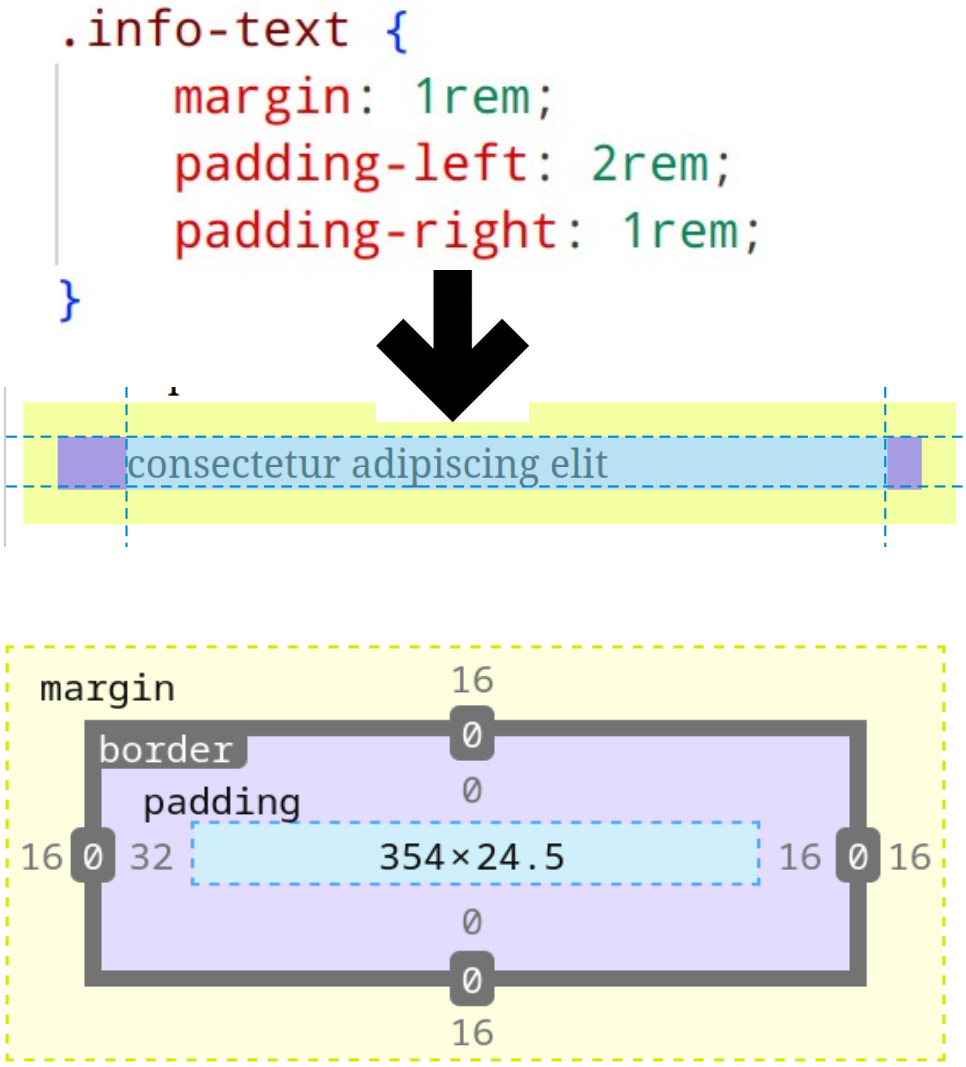
Semanttisten elementtien (ks. *4.2.5 Semanttiset elementit*) ja *divien* (ks. *4.2.4 Div*) kanssa taas suosin prosentuaalisia mittoja.

Kannattaa kuitenkin lukea lisää mittayksiköistä W3Schoolsista:

https://www.w3schools.com/cssref/css_units.php

4.3.4 Marginaalit

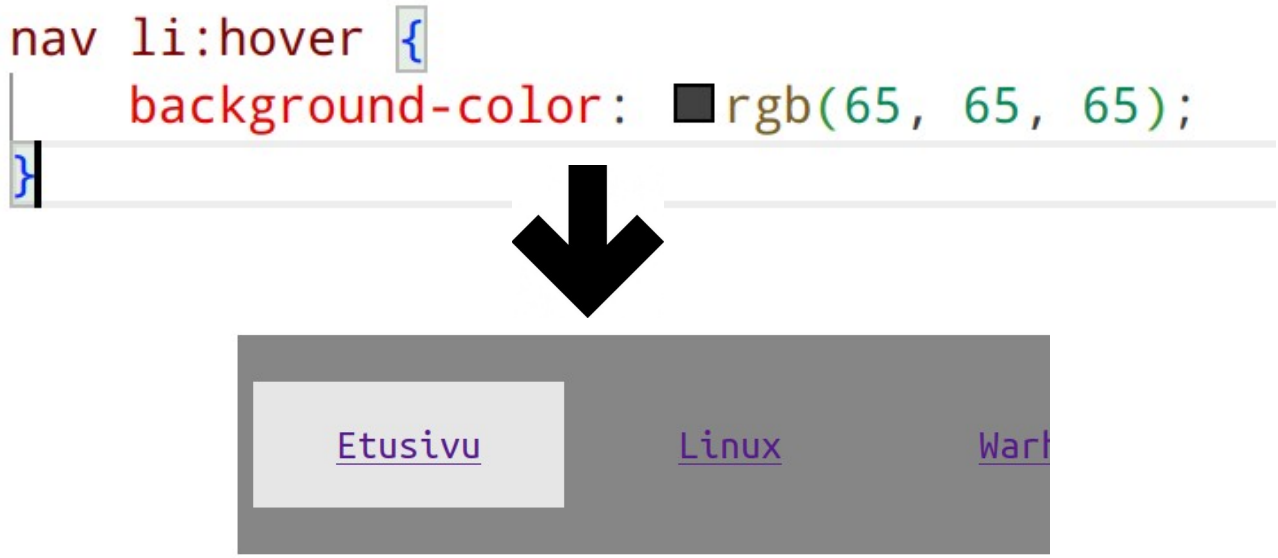
Marginaaleilla saat luotua sivulle keveyttä ja parannat samalla luettavuutta. Marginaaleja on kahdenlaisia: *margin* joka tarkoittaa elementin ulkopuolista (alla olevassa esimerkissä vaalea) ja *padding* joka tarkoittaa elementin sisäistä marginaalia (alla olevassa esimerkissä tummempi).



Kuva 12: Margin tai padding CSS-ominaisuus ilman laitan määrittelyä (esim. margin: 1rem;) asettaa kaikki laidat samaksi, mutta jos jokin laidoista on asetettu erikseen (esim. padding: 2rem;), muut laidat asetetaan nollaan.

4.3.5 Kursorin liikkeet

Tyylittelyjä voi asettaa myös reagoimaan hiiren liikkeisiin. Esimerkiksi *:hover* valitsimella voit asettaa elementille eri tyylit silloin kun kursori on elementin päällä.



Tällä tavoin saat helposti hienon painike efektin. Muista kuitenkin, että *:hover* ei vaikuta mitenkään kosketusnäytöillä kuten kännyköillä.

4.3.6 Responsiivisuus

Responsiivisuus tarkoittaa, että sivu toimii eri kokoisilla näytöillä. Tätä voinee pitää eräänlaisena saavutettavuutena (ks. *4.2.9 Saavutettavuus*). Tyypillisesti riittää, että sivun kehittäjä varautuu leveään tietokoneen näyttöön ja kapeaan kännykän näyttöön.

Responsiivisuuden säätäminen on usein ärsyttävää näpertämistä, joka on kuitenkin ennemmin tai myöhemmin pakko tehdä.

Alla on lista asioista, joita käyttämällä teet sivustasi responsiivisen:

3.6.1.1 Flexbox-wrap

Jos käytät *flexboxia* aseta *flex* elementin *flex-wrap: wrap;* (ks. 4.3.7 Flexbox).

4.3.6.2 Suhteelliset yksiköt

Suosi suhteellisia yksiköitä kuten *%* tai *rem* absoluuttisten sijasta kuten *px*.

4.3.6.3 Viewport asetus

Lisää HTML-dokumenttisi <*head*>-osioon seuraava *meta*-elementti:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

4.3.6.4 Kuvien suhteelliset mitat

Img elementtien mitat (width ja height) kannattaa asettaa muuttumaan ylemmän elementin koon mukaan. Voi tehdä tämän asettamalla joko *width* tai *height 100%*:aan ja toisen arvoksi *auto* (ks. Koodi 6).

```
img {
    width: 100%;
    height: auto;
}
```

Koodi 6: Kuvan leveys on asetettu riippumaan ylemmän elementin koosta ja korkeus seuraamaan leveyttä rikkomatta kuitenkaan kuvasuhdetta. Tämä on tehokas tapa varmistaa, että kuva näkyy hyvin erikokoisilla näytöillä.

4.3.6.5 Media kyselyt

Usein yllä listatut keinot eivät riitä kuitenkaan, jolloin on pakko muokata tyylejä yksi kerrallaan. Ainoa tapa tehdä se on käyttää media kyselyjä (eng. *media query*). Media kyselyillä voi asettaa ehdollisia tyylejä sen mukaan onko laite läppäri, kännykkä tai printteri. Oleellisin media kysely on:

```
@media screen and (max-width: 600px)
```

Tällä voit valita tyylin niille tilanteille, joissa näytön leveys on alle 600px eli käytännössä kaikkien kännyköiden näytöt. Esimerkiksi *Koodissa 7 headerin* marginaalin leveyttä muutetaan riippuen siitä ollaanko kännykän vai läppärin näytöllä.

```
header {
    margin: 1rem;
}

@media screen and (max-width: 600px) {
    header {
        margin-left: 0.5rem;
        margin-right: 0.5rem;
    }
}
```

Koodi 7: Normaalisti headerin margin on joka puolelta 1rem, mutta kännykän näytöille sitä pienennetään 0,5remiin.

4.3.7 Flexbox

Haastavin osa tyylittelyä on saada elementit haluamilleen paikoille. *Flexbox* tehty juuri tätä ongelmaa varten.

Flexboxia tulee käyttää joko semanttisten elementtien (ks. *4.2.4 Semanttiset Elementit*) tai divien (ks. *4.2.4 Div*) kanssa. Voit määritellä sen avulla kuinka elementit *flexbox*-elementin sisällä järjestetään.

Kaikki lähtee siitä, että määrittelet elementin *display* ominaisuuden arvoksi *flexbox* (ks. *Koodi 8*). Se itsessään jo muuttaa alielementtien järjestelyä, mutta voit tämän jälkeen lisätä määrittelyjä, jotta saat elementit haluamaasi järjetykseen.

```
.content {
    display: flex;
    gap: 1rem;
    flex-wrap: wrap;
}
```

Koodi 8: Esimerkki flexbox elementin CSS-luokasta

Alla lista eniten käyttämistäni ominaisuuksista:

1. *align-items*: asettelu pystysuunnassa
2. *justify-content*: asettelu vaakasuunnassa
3. *flex-direction*: suunta johon elementtejä asetellaan (*column* vs. *row*)
4. *gap*: paljonko elementtien välillä on tyhjää
5. *flex-wrap*: siirtää yli menevät elementit uudelle riville

Flexbox saattaa alkuun tuntua hämmentävältä, mutta voit koettaa hahmottaa sitä esimerkiksi *Developer Toolsin* avulla (ks. *4.3.1 Selaimen Developer Tools*). Älä kuitenkaan lannistu mikäli et saa alkuun haluttua lopputulosta.

Parhaan koonnin *flexboxiin* löydät CSS Tricks -sivustolta:

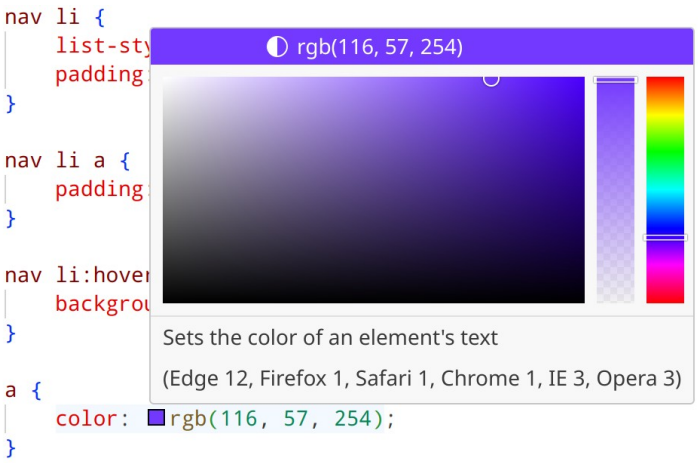
https://css-tricks.com/snippets/css/a-guide-to-flexbox/

4.3.8 Värit

Värien valinta on oleellista ei pelkästään sivun ulkonäön, mutta myös sen saavutettavuuden kannalta (ks. *4.2.9 Saavutettavuus*). Värien kannalta oleellisimmat CSS-ominaisuudet ovat *color,* joka määrittää fontin värin ja *background*-color, joka määrittää taustavärin.

Voit ilmoittaa värin usealla eri tavalla: sanallisesti (“Aquamarine”), RGB:nä (“rgb(127, 255, 212)”), heksadesimaalina (“#7FFFD4”). RGB (Red-Green-Blue) esitykseen on vielä mahdollista lisätä läpinäkyvyys eli *alpha* (“rgba(127, 255, 212, 0.8)”), mutta itse käytän sitä hyvin harvoin.

VS Codessa sekä *Developer Toolsissa* värien valitseminen on tehty kohtalaisen miellyttäväksi siten, että kun viet kursorin värin päälle, sen viereen avautuu värivalitsin (ks. *Kuva 13*), jolla voit kätevästi tummentaa tai vaihtaa väriä toiseen ilman, että joudut muuttamaan parametreja numeraalisesti.



Kuva 13: Värivalitsin VS Codessa säästää paljolta vaivalta värejä säätäessä.

Pari vinkkiä värien valintaan:

1. Älä koskaan käytä täysin mustaa tai täysin valkoista vaan jotain hieman vaaleampaa tai tummempaa.
2. Suosi vahvoja kontrasteja luettavuuden parantamiseksi.
3. Valitse maksimissaan kolme väriä, joita käytät sivustolla. Muuten lopputuloksesta tulee kaoottinen.

Värien valinta on oma taiteen- ja tieteenlajinsa, johon kannattaa tutustua paremmin, jos haluat sivusi erottuvan massasta.

4.3.9 Kirjasintyypit eli fontit

Usein uuden sivuston kehittäminen alkaa fontin valinnalla. Selaimen itsensä tukemat fontit ovat kuitenkin varsin rajallisia ja suosittelenkin kääntymään esimerkiksi *Google Fontsin* suuntaan. Seuraavasta osoitteesta löydät todella laajan valikoiman ilmaisia ja vapaasti käytettävissä olevia fontteja:

https://fonts.google.com/

Google Fontin lisääminen tapahtuu linkkaamalla tarvitsemasi fontit HTML-dokumentin *headissa* (ks. Koodi 9).

```
    <link href='https://fonts.googleapis.com/css?family=Ubuntu Mono' rel='stylesheet'>
    <link href='https://fonts.googleapis.com/css?family=Major Mono Display' rel='stylesheet'>
</head>
```

Koodi 9: Kirjasintyypit "Ubuntu Mono" ja "Major Mono Display" linkattuna HTML-dokumenttiin.

Tämän jälkeen voit asettaa fontin CSS-ominaisuuden *font-familyn* arvoksi (ks. Koodi 10).

```
header {
    background-color: rgb(43, 43, 43);
    margin-top: 2rem;
    padding: 1rem;
    text-align: center;
    font-family: 'Major Mono Display';
}
```

Koodi 10: Kirjasintyyppi "Major Mono Display" asetettuna headerin fontiksi.

Fontti on erittäin oleellinen osa sivun ulkoasua, mutta sen valintaan ei kuitenkaan kannata käyttää loputtomasti aikaa. Jos olet kahden vaiheilla, kannattaa valita yksinkertaisempi.

4.4 Favicon

Pieni mutta tärkeä osa verkkosivun ulkoasua on sen logo. Logon voi asettaa lisäämällä *index.html*-tiedoston kanssa samaan hakemistoon *favicon.ico* nimisen tiedoston. Periaatteessa pelkkä tämän tiedoston lisääminen riittää, mutta jos asiaa haluaa hienosäätää, voi lisätä useamman eri logo-tiedoston. Linkkaamalla ne HTML-dokumentin *headiin* manifesti-tiedoston kera (ks. *Koodi 11*) sivusto valitseen eri kokoisen logon eri käyttökohteisiin.

```
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
```

Koodi 11: Esimerkki favicon-määrittelyjä niin, että esimerkiksi Applen kännykän pikakuvakkeeksi valitaan eri kuva kuin selaimen välilehdelle.

Erinomainen favicon generaattori löytyy osoitteesta:

https://favicon.io/favicon-generator/

5 Loppusanat

Tämä opas vain nopea pintaraapaisu monimutkaiseen ja kiehtovaan verkkosivujen maailmaan. On kuitenkin hyvä muistaa, että loppujen lopuksi kaikki sivustot perustuvat näille edellä esitellyille periaatteille. Jos kiinnostusta riittää, voitkin jatkaa seuraavaksi JavaScriptin pariin. Se on ohjelmointikieli, jolla selaimen saa tekemään mitä erikoisimpia asioita. HTML:llä ja CSS:llä kannattaa kuitenkin aloittaa. Ne ovat ikään kuin verkkosivu-kakun pohja ja kuorrute.