

Pekka Tapio Aalto



FRONT END

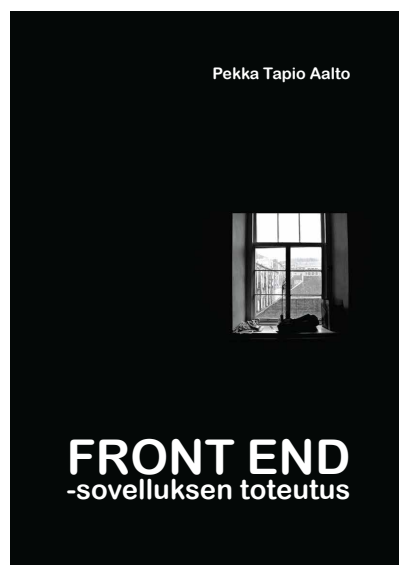
-sovelluksen toteutus

Front End -sovelluksen toteutus

Pekka Tapio Aalto

Tämä kirja on toteutettu seuraavilla työvälineillä:

- \LaTeX ja memoir-luokka (<http://www.ctan.org/pkg/memoir>).
- R (<http://www.r-project.org/>) ja RStudio (<http://www.rstudio.com/>)
- bookdown (<http://bookdown.org/>) ja memoir (<https://ericmarcon.github.io/memoir/>)



Pekka Tapio Aalto
Huittisten ammatti- ja yrittäjäopisto
SASKY koulutuskuntayhtymä
<http://www.sasky.fi>

SISÄLLYS

JOHDANTO

Tämä kirja on tarkoitettu ensisijaisesti KOODAAJA FRONT END -koulutuksen opiskelumateriaaliksi. Osa kirjassa käsitellyistä asioista on sidoksissa koulutuksen toteutustapaan ja se tekee tämän kirjan käyttämisen itseopiskelumateriaalina haasteelliseksi.

KEHITYKSEN TYÖVÄLINEET

Sujuvassa Front End -sovellusten kehityksessä tarvitaan melkoinen määrä erilaisia työkaluja, kuten esimerkiksi

- komentorivi (PowerShell),
- kehitysympäristön työkaluohjelmat (Node.js),
- koodaamiseen soveltuva tekstieditori (Visual Studio Code),
- lähdekoodin hallinta eli versiohallinta (Git).

Seuraavaksi asennamme nämä työvälineet yksi kerrallaan ja testaamme yksinkertaisella testillä, että niiden asennus on onnistunut.

1.1 Komentorivi

Ensimmäinen käyttöön otettava työväline on komentorivi. Ennen graafisia käyttöliittymiä tietokoneita käytettiin merkkipohjaisen komentorivin kautta. Sekä Front End että Back End -sovellusten kehityksessä on hyvä osata komentorivin perusteet.

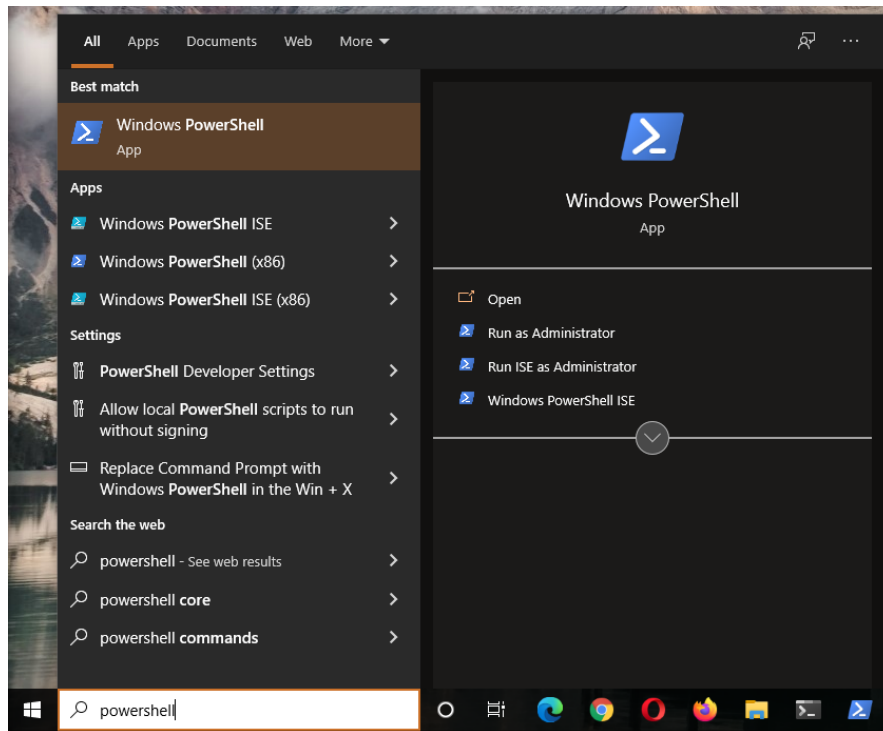
Komentorivin käynnistäminen

Windows

Windowsissa komentoriviä voi käyttää useammalla tavalla. Komentokehote (Command Prompt) perustuu DOS-käyttöjärjestelmän toimintaan ja se on osana nykyistä Windowsia yhteensopivuussyistä. Alkuperäisen komentokehoteen korvaajaksi on kehitetty PowerShell, joka toi Windows-maailmaan Unix-ympäristöissä pitkään olleita ominaisuuksia. Tämän koulutuksen aikana käytämme PowerShellia.

1. KEHITYKSEN TYÖVÄLINEET

Helpoiten PowerShellin voi käynnistää kirjoittamalla Windowsin hakuikkunaan teksti **powershell** ja valitsemalla tarjotuista vaihtoehdoista **Windows Powershell**.



Kuva 1.1: Powershellin käynnistäminen

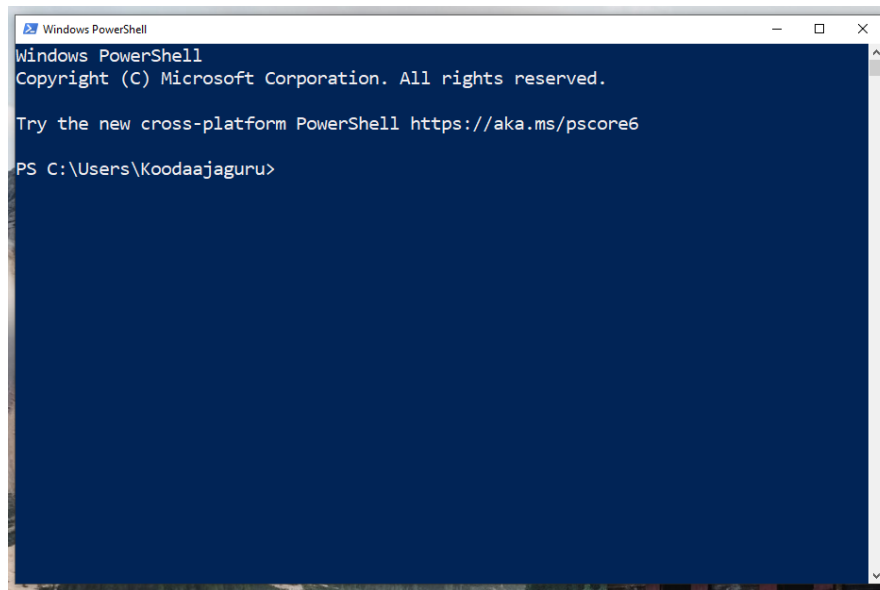
Ruudulle avautuu pienen hetken päästä PowerShell-ikkuna, jossa vilkkuu kursori sen merkiksi, että sille voi syöttää suoritettava komento.

Voit sulkea PowerShell-ikkunan, jota kutsutaan myös istunnoksi (session) joko klikkaamalla ikkunan oikeassa yläkulmassa olevaa rastia tai antamalla komento `exit`.

Windowsissa voi käyttää myös Windows Terminal -sovellusta, joka löytyy ilmaiseksi Microsoft Storesta. Se tukee välilehtiä ja sen kautta pystyy käynnistämään PowerShell-istunnon, komentokehotteen (Command Prompt) tai halutessaan sille voi määritellä SSH-yhteyden omaksi valinnakseen.

Mac

Macissä on oma komentorivinsä, jonka voi käynnistää klikkaamalla *LaunchPad*-ikonia, kirjoittamalla hakukenttään **Terminal** ja valitsemalla sen jälkeen **Terminal**. Lisäohjeita löydät Macin [käyttöohjeista](#).



Kuva 1.2: Powershell käynnistetty

Linux

Jokaisesta Linuxista löytyy joku terminaali, haasteellisempaa on löytää se, miten se käynnistetään, koska käynnistäminen vaihtelee jakeluversiosta ja käytettävästä graafisesta ympäristöstä riippuen. Tyypillisesti se löytyy syöttämällä hakusanoiksi *terminal*, *command*, *prompt* tai *shell*. Useimmissa järjestelmissä komentorivi on nimeltään Terminal tai xterm. Monissa jakeluversiossa toimii myös pikanäppäinyhdistelmä *Ctrl+Alt+T*. Lisäohjeita löydät mm. Ubuntun [käyttöohjeista](#).

Komentorivin käyttäminen

Kansiorakenteessa liikkuminen

Riippumatta siitä, mitä käyttöjärjestelmää käytät, on komentorivin peruskäyttö hyvin samanlaista. Aivan ensimmäiseksi kannattaa opetella liikkumaan hakemistorakenteen sisällä. Lähtökohtaisesti komennolla `dir` saat listattua nykyisessä kansiossa olevat tiedostot ja hakemistot. Esimerkiksi Windowsissa komennon tulos näyttäisi seuraavanlaiselta:

```
PS C:\Users\Koodaajaguru> dir
```

```
Directory: C:\Users\Koodaajaguru
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-r---	1.9.2020	7.16	3D Objects
d-r---	1.9.2020	7.16	Contacts
d-r---	9.3.2021	13.40	Desktop
d-r---	7.4.2021	14.34	Documents
d-r---	7.4.2021	14.34	Downloads
...			

1. KEHITYKSEN TYÖVÄLINEET

d-r---	1.9.2020	7.16	Searches
d-r---	5.9.2020	13.24	Videos
-a----	1.9.2020	12.26	293 .gitconfig

Edellisessä listauksessa kaikki d-kirjaimella alkavat rivit (d-r—) ovat hakemistoja ja loput ovat tavallisia tiedostoja. Alihakemistoon pystyt siirtymään cd-komennolla. Esimerkiksi Windowsissa voit siirtyä käyttäjän Tiedostot-kansioon (Documents) komennolla

```
cd Documents
```

Tämän seurauksena komentokehoite muuttuu niin, että se ilmaisee uuden sijainnin.

Windows on hieman kieroutunut kansion nimien suhteen. Jos tätä kansiota katsotaan Resurssienhallinnan kautta, niin se näkyy suomenkielisessä Windowsissa suomennettuna eli nimellä Tiedostot. Todellisuudessa tämä kansio levytasolla on nimellä Documents ja tätä alkuperäistä nimeä on käytettävä komentorivillä siihen viitattaessa.

```
PS C:\Users\Koodaajaguru\Documents>
```

Macissä ja Linuxissa voit kokeilla siirtymistä mihin tahansa omassa tiedostolistauksessasi olevaan kansioon. Hakemistopuussa liikkuminen on turvallista, se ei sekoita tietokonetta.

Kansiossa, johon edellisessä vaiheessa siirryit, voit tehdä jälleen samat toiminnot eli listata kansiossa olevat tiedostot komennolla `dir` ja siirtymällä kansiossa olevaan alikansioon komennolla `cd alikansio`.

Välillä tulee tilanne, että olisi tarve siirtyä yhtä kansiota ylemmäs. Tämä onnistuu helpoiten komennolla

```
cd ..
```

Tämän seurauksena käyttäjä on siirtynyt kansiotasoa ylemmäs eli malliesimerkissä takaisin `C:\Users\Koodaajaguru` -kansioon.

Kahdella pisteellä on oma erityismerkityksensä hakemistopoluissa. Se viittaa aina kansion ns. äitikansioon eli siihen pääkansioon, jonka alla kansio sijaitsee. Kansion ja sen alikansion voi niputtaa yhteen käyttämällä erottimenä hakemistoerotinta eli `/-`-merkkiä. Esimerkiksi komento `cd ../../` siirtyisi `C:\Users\Koodaajaguru\Documents` -kansioista `C:\Users`-kansioon eli kahta kansiotasoa ylemmäs. Vastaavasti komento `cd ../../Downloads` siirtyisi `C:\Users\Koodaajaguru\Documents` -kansioista `C:\Users\Koodaajaguru\Downloads`-kansioon eli käyttäjän `Lautaukset`-kansioon.

Jos olet eksynyt hakemistopuuhun ja haluat palata takaisin käyttäjän kotikansioon, niin se onnistuu helpoiten komennolla

```
cd ~
```

Alikansion luominen

Sen lisäksi, että osaat liikkua kansiorakenteessa, saatat joutua luomaan uusia alikansiota. Tämäkin tapahtuu peruspiirteissään samalla tavalla riippumatta siitä, että käytätkö Windowsia, Linuxia tai macOSia. Uusi alikansio luodaan `mkdir`-komennolla, joka on lyhenne sanoista *make directory*.

Seuraavaksi luomme kotihakemistoon alikansion, jonne keräämme koulutuksen aikana toteutettavat projektit.

Windows Siirrytään ensin käyttäjän Tiedostot-kansioon komennolla

```
cd ~/Documents
```

ja luodaan Tiedostot-kansioon uusi *projektit*-niminen alikansio komennolla

```
mkdir projektit
```

Jatkossa pääset seuraavalla komennolla projektit-kansioon, aina kun käynnistät PowerShellin uudestaan.

```
cd ~/Documents/projektit
```

Linux ja macOS Unix-pohjaissa käyttöjärjestelmissä kotikansiomalli on erilainen kuin Windowsissa. Tästä johtuen luomme projektit-kansion käyttäjän kotikansion alle. Siirrytään ensin käyttäjän kotikansioon komennolla

```
cd ~
```

ja luodaan uusi *projektit*-niminen alikansio komennolla

```
mkdir projektit
```

Jatkossa pääset seuraavalla komennolla projektit-kansioon, aina kun käynnistät komentorivin.

```
cd ~/projektit
```

Voit halutessasi nimetä projektit-kansion jollain toisella nimellä, mutta muista käyttää tätä antamaasi nimeä jatkossa, kun materiaalissa käytetään projektit-kansiota.

Edellä on todellisuudessa aivan pieni pintaraapaisu siitä, mitä komentorivillä voi tehdä. Tässä vaiheessa riittää, että osaat liikkua hakemistopuussa ylös- ja alaspäin sekä osaat luoda uusia alikansiota. Samoja asioita käsitellään myös [Koodauksen työvälineet - Komentorivi](#) -videossa.

1.2 Node.js

Toinen jatkossa paljonkin käytettävä työväline on *Node.js*-ympäristö. Se on ajoympäristö, joka mahdollistaa JavaScriptillä toteutettujen ohjelmien suorittamisen omalla koneella. Node.js on ladattavissa yleisimmille käyttöjärjestelmille. Toisin sanoen sille löytyy valmiit asennuspaketit Windowsille, Linuxille ja macOSille.

Node.js:n asennus

1. Mene selaimella osoitteeseen <https://nodejs.org/en/>.



Kuva 1.3: Node.js-sivusto

2. Klikkaa **Current**-vaihtoehtoa, se on uusin ja viimeisin käännetty Node.js-versio. LTS-vaihtoehto on vakaampi versio, joka on ensisijaisesti tarkoitettu tuotantokäyttöön tai tilanteisiin, jossa uusimman version ominaisuuksia ei ole vielä tuettu.

Sivusto tunnistaa automaattisesti käyttämäsi käyttöjärjestelmän ja tarjoaa sitä vastaavaa asennuspakettia. Jos näin ei kuitenkaan tapahdu, niin klikkaa silloin sivun yläreunasta Downloads-valinta ja valitse sieltä käyttöjärjestelmääsi vastaava asennuspaketti.

3. Käynnistä lataamasi asennuspaketti tuplaklikkaamalla sitä. Seuraavat ohjeet noudattavat Windows-paketin asennusvaiheita, jos asennat ympäristöä toiselle käyttöjärjestelmälle, niin asennuksen kulku on silloin erilainen.

- Klikkaa *Welcome to Node.js Setup Wizard* -näkylässä **Next**.
- Hyväksy *End-User License Agreement* -näkylässä lisenssi valitsemalla **I accept the terms in the License Agreement** ja klikkaa **Next**.
- Hyväksy tarjottu asennuskansio *Destination Folder* -näkylässä klikkaamalla **Next**.
- Hyväksy asennettavat paketit *Custom Setup* -näkylässä klikkaamalla **Next**.
- Tässä koulutuksessa ei ole tarvetta kääntää moduuleita, joten *Tool for Native Modules* -näkylässä ei tehdä tarjottua valintaa, vaan klikataan ainoastaan **Next**.
- Hyväksy lopulta *Ready to install Node.js* -näkylässä Node.js-ympäristön asennus tekemilläsi valinnoilla klikkaamalla **Install**. Asennusohjelma kysyy pääkäyttäjän oikeuksia, jotka sinun tulee myöntää.
- Asennusohjelma asentaa kaikki tarvittavat tiedostot sekä määrittelee polkuasetuksen. Lopulta asennusohjelma päättyy *Completed the Node.js Setup Wizard* -näkylään, jossa voit klikata **Finish** päättääksesi asennusohjelman.

4. Windowsin Node.js hyödyntää taustalla PowerShell-skriptejä, joiden suorittaminen on oletusarvoisesti estetty. Jotta nämä skriptit toimivat ongelmitta, on sinun tehtävä seuraava määrittely koneellesi PowerShellin kautta:

```
Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
```

Tämä komento sallii paikallisten PowerShell-skriptien suorittamisen käyttäjällä, jolla olet kirjautunut sisälle. Ladatut skriptit vaativat jatkossa ns. allekirjoituksen. Komento varmistaa vielä, että olet tietoinen muutoksen sivuvaikutuksista. Vastaa tähän kysymykseen **Y** eli **Yes**.

Komennon suorituksen jälkeen komentorivi näyttää seuraavanlaiselta:

```
PS C:\Users\Koodaajaguru> Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
```

```
Execution Policy Change
```

```
The execution policy helps protect you from scripts that you do not trust. Changing the
execution policy might expose you to the security risks described in the
about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
```

Asennuksen onnistumisen tarkistaminen

Seuraavaksi tarkistetaan, että Node.js-ympäristö toimii niin, että voimme sitä jatkossa käyttää. Tulosta asennetun Node.js:n versionumero komennolla

```
node -v
```

Tämä komento tulostaa asennetun ympäristön versionumeron, tässä materiaalissa asennetulla versiolla tulostuu

```
v17.3.1
```

Asentamasi versio on todennäköisesti eri. Oleellista on, että komento tulostaa versionumeron, joka on sama tai uudempi kuin esimerkissä. Tarkistetaan myös, että npm-pakettihallinta on asentunut oikein syöttämällä komento

```
npm -v
```

Komento tulostaa asennetun npm-sovelluksen versionumeron, joka on materiaalin asennuspaketilla

```
8.3.0
```

Tässäkin tapauksessa versionumero voi poiketa. Oleellista on, että versionumero on sama tai uudempi kuin esimerkissä.

Nämä samat asiat käsitellään myös [Koodauksen työvälineet - Node.js](#)-videossa sillä erolla, että videossa asennetaan vanhempi versio Node.js-ympäristöstä.

1.3 npm

Node.js-ympäristön mukana asennettiin *npm*-niminen sovellus, jolla voi asentaa, päivittää ja poistaa npm-kirjastossa julkaistuja paketteja. npm tulee sanoista Node Package Management ja sen kirjasto sisältää uskomattoman määrän eri käyttötarkoituksiin toteutettuja JavaScript-sovelluksia, jotka auttavat joko sovelluskehityksessä, ylläpidossa tai niitä voi hyödyntää osana omaa sovellusta.

Suoritus ilman asentamista

npm-pakettien suorittaminen vaatii, että niiden sovelluskoodit löytyvät suorittavalta koneelta. Aikaisemmin tämä vaati paketin asentamista, uusimmissa versioissa suorittaminen onnistuu ilman asentamista erillisellä *npx*-ohjelmalla. Taustalla *npx* käy lataamassa väliaikaisesti tarvittavat koodit, suorittaa sen ja sen jälkeen poistaa suoritukseen ladatut koodit.

Esimerkiksi seuraava komento käy lataamassa cowsay-nimisen npm-paketin, suorittaa sen antamalla sen syötteen komentorivin lopussa olevan tekstin ja poistaa lopuksi ladatut (ja väliaikaisesti) asennetut tiedostot.

```
npm cowsay "Nyt lähtee homma käsistä!"
```

Komento tuottaa seuraavan tulostuksen. Mahdolliseen varmistusviestiin voi vastata **y**.

```
Need to install the following packages:
```

```
cowsay
```

```
Ok to proceed? (y) y
```

```
< Nyt lähtee homma käsistä! >
```

```

      \      ^__^
       (oo)\_______
            (__)\       )\/\
                ||----w |
                ||

```

Kuten tulosteesta voi päätellä, cowsay-paketista on loppujen lopuksi varsin vähän todellista hyötyä. npx-komennolla kannattaa ajaa harvoin tarvittavia paketteja, tällöin paketin asennus ei vie turhaa levytilaa ja ajettaessa käytetään aina uusinta versiota paketista.

Paketin asentaminen globaalisti

npm-pakettien iso haittapuoli on niiden tarvitsema levytila. Yksittäisessä projektissa sillä ei ole juuri merkitystä, mutta tilantarve kertaantuu tilanteissa, jossa samaa sovellusta tarvitaan useammassa projektissa. Tämänkaltaisissa tilanteissa npm-paketin voi asentaa ns. globaalisti eli niin, että se on käytettävissä missä tahansa kansiossa. Huomaa kuitenkin, että globaali asennus on käyttäjäkohtainen, se ei tee npm-paketin asennusta näkyväksi kaikille koneen käyttäjille.

Tämän koulutuksen aikana tulemme hyödyntämään tehtävien tarkistuksessa automaattista tarkistusta. Käytännössä tämä tarkoittaa sitä, että tehtävän vastaus ajetaan tarkistustestien lävitse. Jotta voit jatkossa testata, että meneekö tekemäsi tehtävä testistä läbitse, niin kannattaa levytilan kulutuksen vähentämiseksi asentaa tehtävissä käytettävät testauspaketit globaalisti.

TestCafe-paketin asennus

TestCafe on testausympäristö, jolla voi testata HTML-sivun toimintaa. Lisätietoja paketista löydät [paketin kuvauksesta](#). Asennetaan *testcafe*-paketti globaalisti komennolla

```
npm install -g testcafe
```

1. KEHITYKSEN TYÖVÄLINEET

Komentorivillä `install` kertoo, että asennetaan uusi paketti, `-g` ilmaisee, että paketti asennetaan globaalisti ja `testcafe` on asennettavan paketin nimi. Komennon suoritus kestää jonkin aikaa ja ruudulle saattaa tulostua *npm WARN*-alkuisia varoitusrivejä. Asennusvaiheessa tulevat varotukset ovat ihan normaaleita, ne eivät estä sovelluksen asentumista ja suorittamista. Kun sovellus on asentunut, tulostuu ruudulle lopuksi seuraavankaltainen teksti:

```
added 462 packages, and audited 463 packages in 57s
```

```
26 packages are looking for funding  
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

Voit testata, että `tescafe` tuli asennettua ja se käynnistyy komennolla

```
testcafe -v
```

Jos ruudulle tulostuu testcafen versionumero, on kaikki kunnossa.

```
1.18.1
```

Jest-paketin asennus

Jest on erityisesti JavaScriptin testaukseen soveltuva testausympäristö. Lisätietoja Jestistä löydät sen [kotisivuilta](#). Asennetaan samalla tavalla jest-paketti globaalisti komennolla

```
npm install -g jest
```

Asennuksen jälkeen tulostuu seuraavankaltaiset tekstit ilmaisemaan asennuksen onnistumista.

```
added 323 packages, and audited 324 packages in 14s
```

```
26 packages are looking for funding  
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

Asennusta voi testata suorittamalla komento

```
jest -v
```

Jos tulokseksi tuli jestin versionumero, niin asennus onnistui.

```
27.4.7
```

npm-sovelluksen käyttö ja testauspakettien asennus käsitellään myös [Koodauksen työvälineet - npm](#) -videossa.

1.4 Visual Studio Code

Koodauksen yksi tärkeimmistä työkaluista on tekstieditori, jolla muokataan lähdekoodia ja muita tekstitiedostoja. Periaatteessa tähän tarkoitukseen kelpaa melkein mikä tahansa tekstieditori, hätätilanteessa Windowsin Muistio toimii peruseditointiin. Jotta ohjelmakoodin kirjoittaminen olisi tehokasta ja tuottavaa, kannattaa käyttöön ottaa koodaamiseen erityisesti suunniteltu tekstieditori.

Tässä materiaalissa käytetään Visual Studio Code -nimistä tekstieditoria, pääosin seuraavista syistä:

- Se on vapaan lähdekoodin projekti, toisin sanoen se on ilmainen.
- Sille löytyy valmiit asennuspaketit Windowsille, Linuxille ja macOS:lle.
- Se on kooltaan maltillinen ja se käynnistyy kohtuullisen nopeasti.
- Sille löytyy uskomaton määrä erilaisia lisäosia, joilla sen toiminnallisuutta pystyy laajentamaan.
- Siihen on sisäänrakennettu tuki Git-versiohallinnalle.
- Se on käytetyin tekstieditori Front End -kehityksessä.

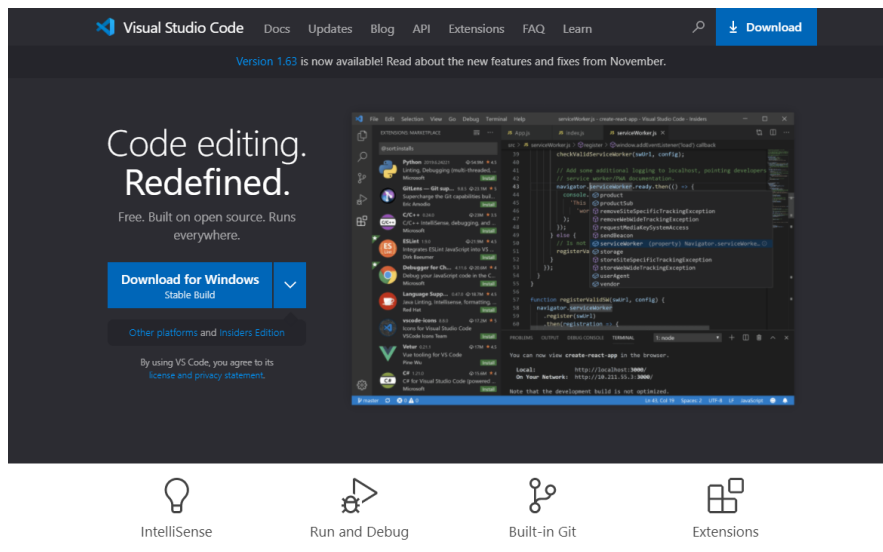
Halutessasi voit käyttää mitä tahansa muuta editoria, kuten esimerkiksi:

- [Atom](#) (ilmainen)
- [Sublime Text](#) (maksullinen, kokeilujaksoa ei rajoitettu)
- [WebStorm](#) (maksullinen, opiskelijana mahdollisuus käyttää ilmaiseksi)

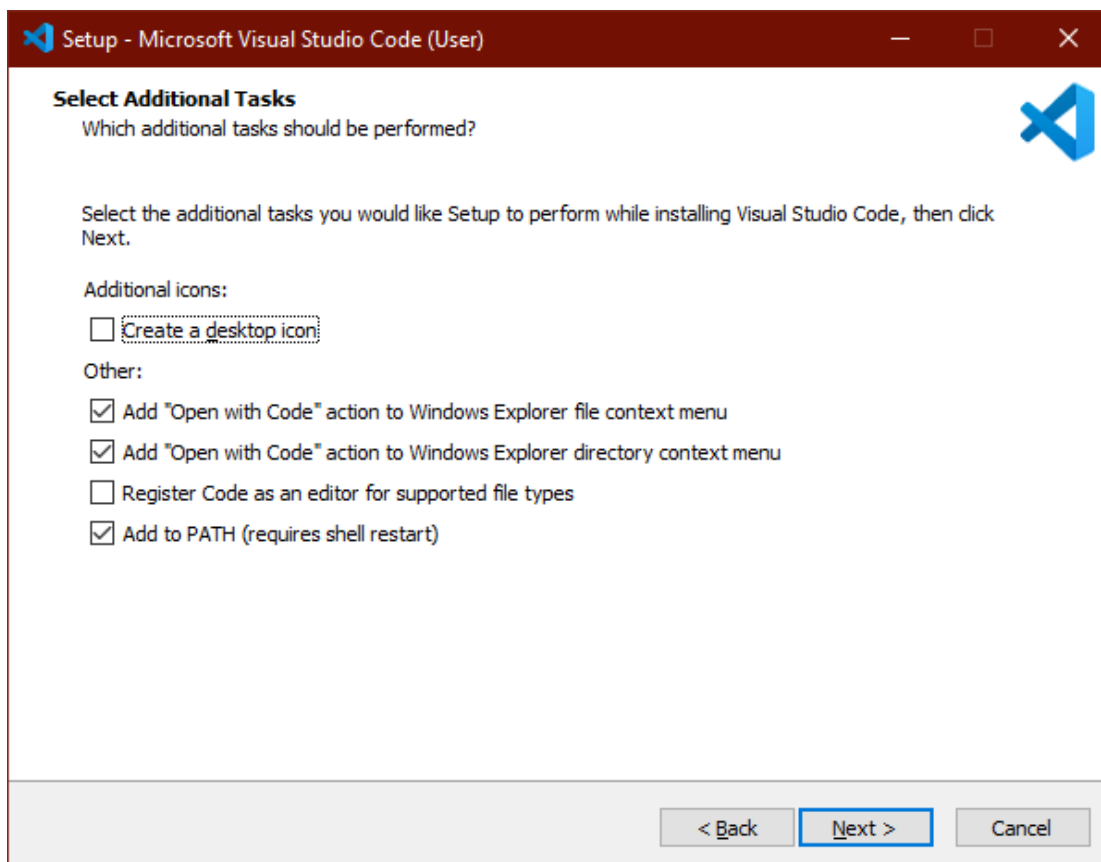
Visual Studio Coden asentaminen

1. Mene selaimella osoitteeseen <https://code.visualstudio.com>.
2. Klikkaa **Download for Windows**-nappia. Sivusto tunnistaa automaattisesti käyttämäsi käyttöjärjestelmän ja tarjoaa sitä vastaavaa asennuspakettia. Jos näin ei kuitenkaan tapahdu, niin klikkaa silloin sivun yläreunasta Downloads-linkkiä ja valitse sieltä käyttöjärjestelmääsi vastaava asennuspaketti.
3. Käynnistä lataamasi asennuspaketti tuplaklikkaamalla sitä. Seuraavat ohjeet noudattavat Windows-paketin asennusvaiheita, jos asennat ympäristöä toiselle käyttöjärjestelmälle, niin asennuksen kulku on silloin erilainen.
 - Hyväksy *License Agreement* -näkylässä lisenssi valitsemalla **I accept the agreement** ja klikkaa **Next**.
 - Valitse *Select Additional Tasks*-näkylässä ne valinnat, jotka haluat ottaa käyttöön. Alla on suositeltavat valinnat:

1. KEHITYKSEN TYÖVÄLINEET



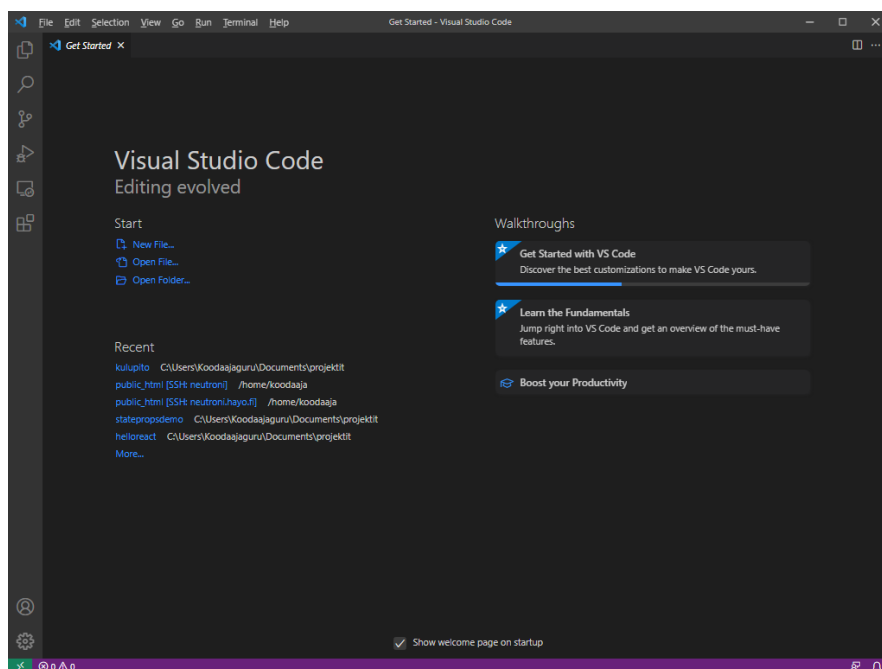
Kuva 1.4: Visual Studio Code-sivusto



Kuva 1.5: Select Additional Task -näkömän valinnat

Voit valita myös kohdan **Register Code as an editor for Supported file types**, jos et käytä aktiivisesti jotain toista tekstieditoria koodien ja muiden tekstitiedostojen muokkaamiseen.

- Käynnistä *Ready to Install* -näkyvässä asennus klikkaamalla **Install**.
- Asennusohjelma asentaa kaikki tarvittavat tiedostot sekä määrittelee polkuasetuksen. Lopulta asennusohjelma päättyy *Completing the Visual Studio Code Setup Wizard* -näkyvään, jossa voit klikata **Finish** päättääksesi asennusohjelman. Asennusohjelma käynnistää oletuksena viimeisenä vaiheena juuri asennetun Visual Studio Coden.



Kuva 1.6: Visual Studio Code

Visual Studio Coden testaaminen

Visual Studio Code perustuu toiminta-ajatuksen, että projektia työstettäessä ei muokata projektin yksittäistä tiedostoa vaan muokkauksen alla on projektikansio ja sen alla olevat tiedostot. Tällöin Visual Studio Code tietää minkä projektin versiohallintaa sen tulisi seurata.

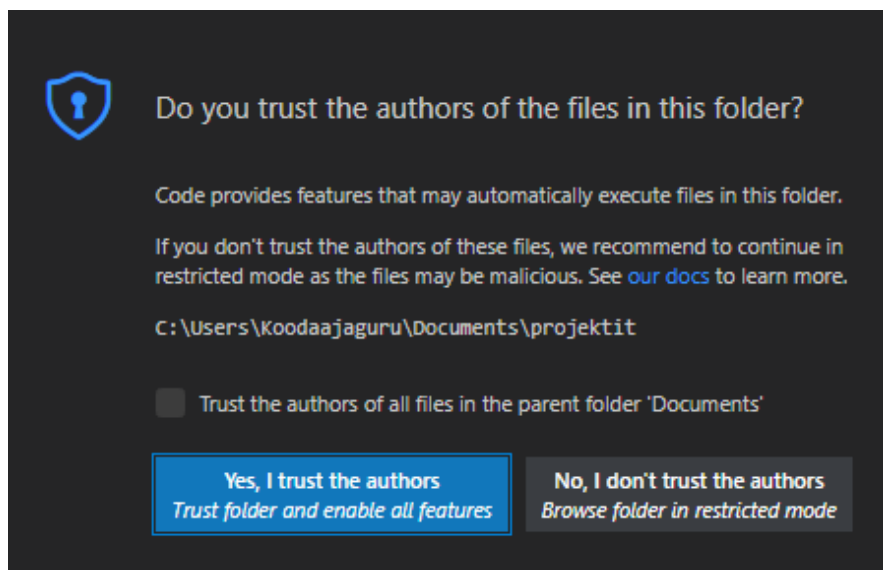
Projektikansion avaaminen käsittelyyn tapahtuu klikkaamalla aloitusnäytöllä olevaa **Open Folder...**-linkkiä tai valitsemalla valikosta **File > Open Folder...**. Voit tällä kertaa valita avattavaksi kansioksi käyttäjän *Tiedostot*-kansion alta löytyvä **projektit**-kansio.

Muista seuraavalla kerralla avata nimenomainen projektikansio, jota haluat käsitellä. Päädyt ennen pitkää ongelmiin, jos sinulla

1. KEHITYKSEN TYÖVÄLINEET

on projektit-kansion alla useampi projekti joita työstät ja muokkaat kaikkia niin, että olet avannut Visual Studio Codessa projektit-kansion, etkä nimenomaista työn alla olevaa projektikansiota.

Kansion avaamisen yhteydessä VSC saattaa varmistaa, että luotatko kansion sisällä olevien tiedostojen sisältöihin. Jos kansio sisältää sinun itsesi tai luotettavan tahon tuottamaa sisältöä, voit valita vaihtoehdon **Yes, I trust the authors**. Valitse toinen vaihtoehto, jos sinulla ei ole täyttä varmuutta koodin sisällöstä tai lähteestä.



Kuva 1.7: Varmistus kansion sisällön luottamiseen

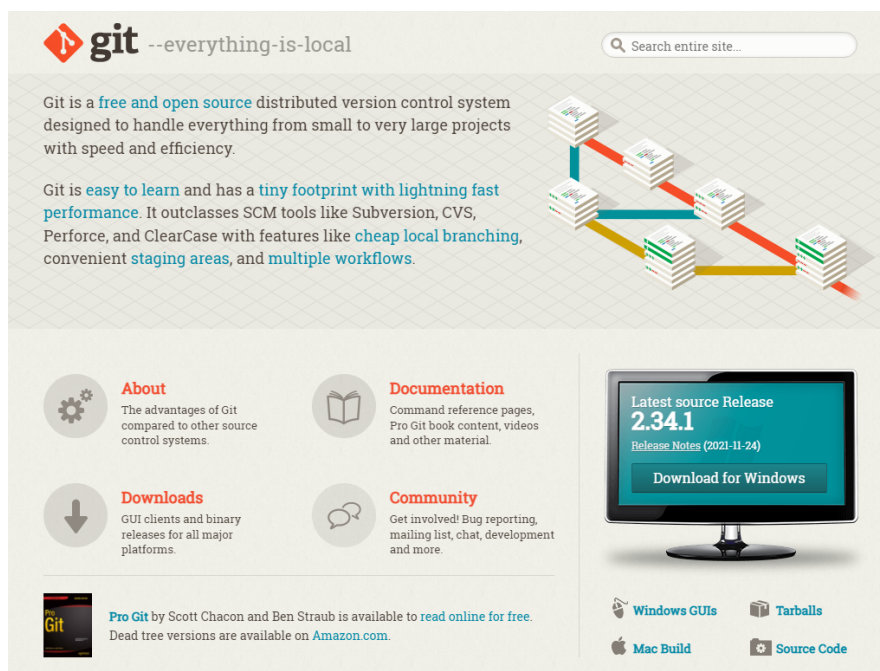
Visual Studio Code -sovelluksen asennus ja peruskäyttö on kerrottu myös [Koodauksen työvälineet - Visual Studio Code](#) -videossa.

1.5 Git

Seuraava asennettava työkalu on Git-versiohallinta. Git on lähtökohtaisesti komentorivipohjainen työkalu, mutta esimerkiksi Visual Studio Code mahdollistaa Gitin käyttämisen graafisella käyttöliittymällä. Visual Studio Coden puolella Gitin ominaisuudet ovat käytettävissä heti, kun Gitin peruspaketti on asennettu järjestelmään.

Gitin asentaminen

1. Mene selaimella osoitteeseen <https://git-scm.com>.
2. Klikkaa **Download for Windows**-nappia. Sivusto tunnistaa automaattisesti käyttämäsi käyttöjärjestelmän ja tarjoaa sitä vastaavan asennustavan. Jos



Kuva 1.8: Git-versiohallinnan kotisivut

näin ei kuitenkaan tapahdu, niin klikkaa silloin sivulta löytyvää Downloads-linkkiä ja valitse sieltä käyttöjärjestelmääsi vastaava asennustapa.

Valitse avautuvalta sivulta **Click here to download**-linkkiä, jolloin saat ladattua koneellesi asennuspaketin.

3. Käynnistä lataamasi asennuspaketti tuplaklikkaamalla sitä. Seuraavat ohjeet noudattavat Windows-paketin asennusvaiheita, jos asennat ympäristöä toiselle käyttöjärjestelmälle, niin asennuksen kulku on silloin erilainen.

- Hyväksy *Information* -näkyssä oleva lisenssi valitsemalla **Next**.
- Klikkaa *Select Destination Location*-näkyssä **Next**.
- Klikkaa *Select Components*-näkyssä **Next**.
- Klikkaa *Select Start Menu Folder*-näkyssä **Next**.
- Valitse *Choosing the default editor user by Git*-näkyman listasta vaihtoehto **Use Visual Studio Code as Git's default editor** ja jatka klikkaamalla **Next**.
- Klikkaa *Adjusting the name of the initial branch in new repositories*-näkyssä **Next**.
- Klikkaa *Adjusting your PATH environment*-näkyssä **Next**.
- Klikkaa *Choosing the SSH executable*-näkyssä **Next**.
- Klikkaa *Choosing HTTPS transport backend*-näkyssä **Next**.
- Klikkaa *Configuring the line ending conversions*-näkyssä **Next**.
- Klikkaa *Configuring the terminal emulator to use with Git Bash*-näkyssä **Next**.

1. KEHITYKSEN TYÖVÄLINEET

- Klikkaa *Choose the default behaviour of 'git pull'*-näkylässä **Next**.
- Klikkaa *Choose a credential helper*-näkylässä **Next**.
- Klikkaa *Configuring extra options*-näkylässä **Next**.
- Aloita asennus *Configuring experimental options*-näkylässä klikkaamalla **Install**.
- Asennusohjelma asentaa kaikki tarvittavat tiedostot sekä määrittelee asetukset. Lopulta asennusohjelma päättyy *Completing the Git Setup Wizard* -näkylässä, jossa voit klikata **Finish** päättääksesi asennusohjelman.

4. Testaa Gitin asennuksen onnistuminen antamalla komentoriville komento

```
git --version
```

Asennus onnistui, jos komento tulostaa sinulle asennetun Gitin versionumeron:

```
git version 2.34.1.windows.1
```

Gitin asentaminen on kuvattu myös Pro Git -kirjan [Getting Started - Installing Git](#) -luvussa.

Gitin alkuasetukset

Jotta Gitiä voidaan ottaa käyttöön, on sille määriteltävä kaksi asetusta komentorivin kautta. Määritetään ensin käyttäjän nimi suorittamalla komentorivillä seuraava komento. Korvaa nimen paikalle oma nimesi.

```
git config --global user.name "Koodaajaguru"
```

Nimen lisäksi täytyy määritellä myös käyttäjän sähköpostiosoite. Nimi ja sähköpostiosoite tulee näkymään Git-repon hyväksytyissä commit-taltioinneissa. Määrittele sähköpostiosoite komennolla. Korvaa komentoon oma sähköpostiosoitteesi.

```
git config --global user.email koodaajaguru@gmail.com
```

Gitin alkuasetusten määrittelyä on selostettu Pro Git -kirjan [Getting Started - First-Time Git Setup](#) -luvussa.

Git -versiohallinnan asennus ja asetukset on kerrottu myös [Koodauksen työvälineet - Git](#) -videossa.