

# GreenICTComp



ICT-ohjelmistot ja kiertotalous



CC-4-BY



# ICT ohjelmistot ja kiertotalous

Kimmo Tarkkanen  
Turun ammattikorkeakoulu

CC-BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)



# Sisältö

A

## Kotitehtävän läpikäynti

Oman organisaation ICT-laitteistopolitiikka

B

## Kiertotalous ohjelmistohankinnoissa

- Kriteerejä kestävien ICT-hankintojen tueksi

C

## Kiertotalous ohjelmistokehityksessä

D

## Ennakkotehtävän läpikäynti

- Ohjelmistokehityksen vihreät käytännöt – kysely
- Omat listat

E

## Kurssitehtävän suunnittelu

# Kotitehtävä

- Keskustelua kotitehtävästä
  - Oman organisaation ICT-laitteisto-politiikka

## Laitteen arvioitu elinkaari

- Tietokone 4-5 vuotta
- Matkapuhelin 2-3 vuotta.
- Suosittelemme näitä merkkejä ja Android-puhelimissa **Android Enterprise-malleja:**  
iPhone, Motorola, Samsung, Nokia, Sony



# Kiertotalouden periaatteita

Kiertotalouden periaatteet pyrkivät minimoimaan hukan ja resurssien tuhlaamisen, sekä edistämään resurssien tehokasta käyttöä ja kestäväää kehitystä.

- **Resurssien tehokas käyttö:** Tavoitteena on käyttää luonnonvaroja ja muita resursseja mahdollisimman tehokkaasti niin, että niiden käyttöaika ja arvo optimoidaan.
- **Hukan vähentäminen:** Kiertotalous pyrkii minimoimaan hukan määrää tuotannossa, kulutuksessa ja hävittämisessä. Tämä saavutetaan optimoimalla prosesseja ja suunnittelemalla tuotteita niin, että niitä voidaan hyödyntää paremmin niiden elinkaaren aikana.
- **Kierrätys ja uusiokäyttö:** Kiertotalous korostaa kierrätyksen ja uusiokäytön merkitystä, jotta materiaalit ja resurssit voidaan pitää kierrossa mahdollisimman pitkään.

Miten ohjelmistoilla edistetään kiertotalouden periaatteita?  
Miten kiertotalouden periaatteita voi soveltaa itse ohjelmistojen hankinnassa tai kehittämisessä?

# Kiertotalouden periaatteita

1. Jakaminen
2. Säilyttäminen ja ylläpitäminen
3. Uudelleenkäyttö tai jälleenmyynti
4. Uudistus tai uudelleenvalmistus
5. Kierrätys raaka-aineeksi tai komponenteiksi
6. (Hyödyntäminen toisessa arvoketjussa)

Ellen MacArthur Foundation on visualisoinut kiertotalouden perusperiaatteita perhosdiagrammin avulla. Tutustua perhosdiagrammiin osoitteessa <https://www.ellenmacarthurfoundation.org/circular-economy-diagram>

# ICT laitteet ja kiertotalous

## Esimerkit

- Apple's comprehensive carbon footprint
- Swappie – puhelimien kierrätystä

## Lähteet:

[https://www.apple.com/mideast/environment/pdf/en/Apple\\_Environmental\\_Progress\\_Report\\_2023.pdf](https://www.apple.com/mideast/environment/pdf/en/Apple_Environmental_Progress_Report_2023.pdf)

<https://swappie.com/fi>

# Kiertotalous ja ohjelmistot

# Ohjelmistot kiertotalouden mahdollistajana

Mitä kiertotalouden periaatteita seuraavat ohjelmistot mahdollistavat?

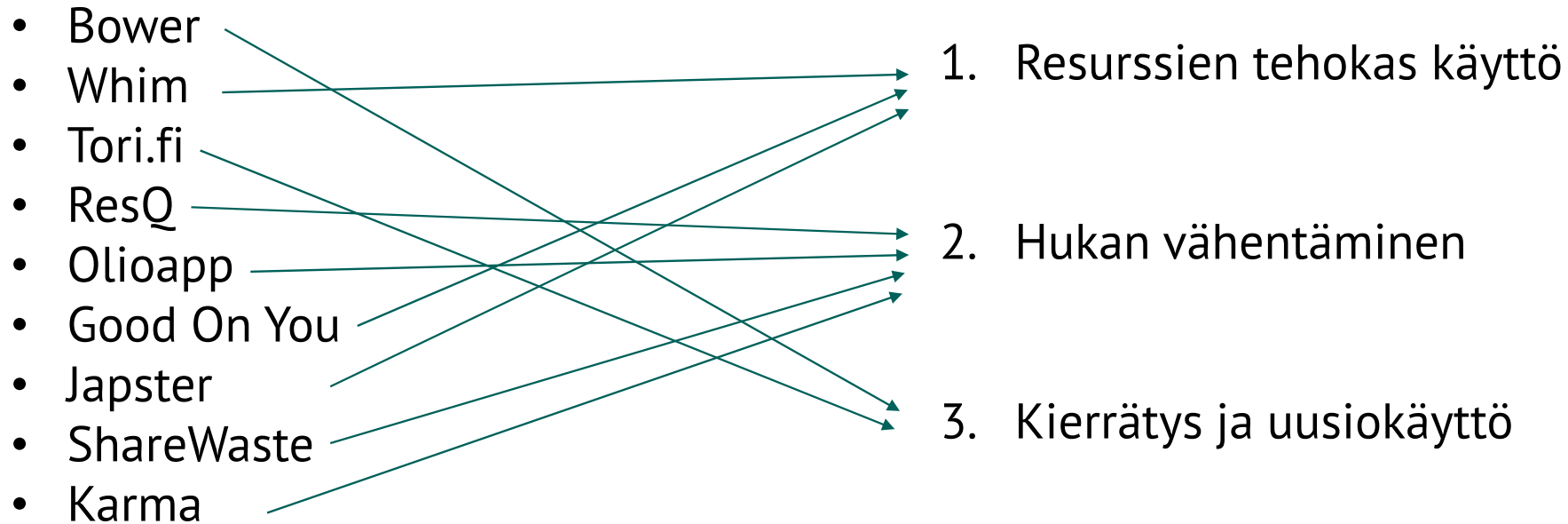
Esimerkit:

- Bower-sovellus
  - Linkki: HS 9.3.2023 “Kierrätä sipsipussi, saa rahaa” <https://www.hs.fi/suomi/art-2000009442250.html>
  - Vastaus: Kohta 5. Sovelluksen tarkoituksena on varmistaa kierrätys raaka-aineeksi.
- Whim-sovellus
  - Linkki: Yle 2.5.2018 ”Mikä ihmeen Whim? – Uusi palvelu yhdistää bussit, taksit, kaupunkipyörät ja vuokra-autot yhden klikkauksen alle, mutta väheneekö yksityisautoilu?” <https://yle.fi/a/3-10171507>
  - Linkki: HS 4.3.2024 ”Jälleen yksi liikkumispalvelu lopetetaan Helsingissä” <https://www.hs.fi/pkseutu/art-2000010269339.html>
  - Vastaus: Kohta 1. Sovelluksen tarkoituksena on jakaminen

# Ohjelmistot kiertotalouden mahdollistajana

- Kiertotaloudessa on kolme keskeistä tapaa lisätä ja ylläpitää arvoa
  - Resurssien tehokas käyttö
  - Hukan vähentäminen
  - Kierrätys ja uusiokäyttö
- Ohjelmistojen avulla voidaan edistää tuotteiden jatkokäyttöä ja kierrättämistä, vähentää hävikkiä, tukea eettistä valmistusta ja vahvistaa kierrätetyn materiaalin uusiokäyttöä

# Tehtävä: Miten seuraavat ohjelmistot edistävät kiertotaloutta?



# Vastaus: Miten seuraavat ohjelmistot edistävät kiertotaloutta?

1. Bower: 3. Kierrätys ja uusiokäyttö. Bower mahdollistaa käytettyjen ja ylijäämämateriaalien kierrättämisen rakennusteollisuudessa, mikä edistää materiaalien uusiokäyttöä ja kierrätystä.
2. Whim: 1. Resurssien tehokas käyttö. Whim tarjoaa kokonaisvaltaisen joukkoliikenne- ja liikkumispalvelun, joka kannustaa käyttämään julkisia kulkuneuvoja ja muita yhteiskäyttöisiä liikkumisvaihtoehtoja tehokkaasti.
3. Tori.fi: 3. Kierrätys ja uusiokäyttö. Tori.fi tarjoaa foorumin käytettyjen tavaroiden ostamiselle, myymiselle ja vaihtamiselle, mikä edistää tavaroiden uusiokäyttöä ja kierrätystä.
4. ResQ: 2. Hukan vähentäminen. ResQ auttaa vähentämään ravintolaruokahävikkiä tarjoamalla alennettuja ruoka-annoksia ruoan sulkemisaikojen lähestyessä.
5. Olioapp: 2. Hukan vähentäminen. Olioapp helpottaa ruoan jakamista ja kierrättämistä yhteisöissä, mikä auttaa vähentämään ruokahävikkiä. [https://en.wikipedia.org/wiki/Olio\\_\(app\)](https://en.wikipedia.org/wiki/Olio_(app)): lahjoittamalla, hankkimalla, lainaamalla tai lainaamalla yhteisössasi tavaroita ilmaiseksi. Tavoitteena on vähentää kotitalous- ja ruokahävikkiä. Se tekee tämän yhdistämällä naapurit, joilla on ylimääräistä ruokaa tai taloustavaroita, muihin lähellä oleviin, jotka haluavat noutaa nämä tavarat.
6. Good On You: 1. Resurssien tehokas käyttö. Good On You tarjoaa tietoa vaatemerkkien eettisyydestä ja kestävydestä, mikä auttaa kuluttajia tekemään tietoisia valintoja ja käyttämään resursseja tehokkaasti vaateteollisuudessa.
7. Japster: 1. Resurssien tehokas käyttö. Japster yhteiskäyttöautopalvelu.
8. ShareWaste: 2. Hukan vähentäminen. ShareWaste mahdollistaa ruokajätteen kompostoinnin yhteisöissä, mikä auttaa vähentämään ruokajätteen määrää kaatopaikoilla.
9. Karma: 2. Hukan vähentäminen. Karma mahdollistaa ruoan ylijäämän ostamisen alennettuun hintaan ruokakaupoista, mikä auttaa vähentämään ruokahävikkiä kaupoissa. <https://save.karma.life/>

# Millaisia kiertotaloutta edistäviä asetuksia löydät käyttämistäsi ohjelmistoista?

Tutustu esimerkiksi seuraaviin:

- Selaimen asetukset
- Käyttöjärjestelmän asetukset
- Älypuhelimien asetukset
  
- [Esimerkki Google Chrome tietyn välilehden muistin käyttö]

# Kiertotalous ohjelmistohankinnassa

# Kiertotalouden periaatteita noudattavia ohjelmistohankinnan käytäntöjä

- 1. Pilvipalvelut ja SaaS:** Pilvipalveluiden ja ohjelmistotason palveluiden käyttö voi olla kiertotalouden periaatteiden mukaista, koska se voi vähentää tarvetta fyysisille resursseille, kuten palvelimille ja laitteistolle. Lisäksi palveluita voidaan skaalata ja maksaa vain tarpeen mukaan.
- 2. Kestävät hankintakäytännöt:** Organisaatiot voivat valita ohjelmistoja, jotka on suunniteltu kestävästi ja jotka ovat helposti päivitettäviä ja ylläpidettäviä. Tämä voi sisältää tukipalveluiden, korjausten ja päivitysten saatavuuden.
- 3. Sopimusten joustavuus:** Sopimusten rakentaminen joustavasti voi mahdollistaa ohjelmistojen käytön ja maksamisen vain tarpeen mukaan. Tämä vähentää hukkaa ja tarpeettomia kustannuksia.
- 4. Vastuulliset toimittajat:** Organisaatiot voivat suosia ohjelmistotoimittajia, jotka noudattavat kestävyteen ja vastuullisuuteen liittyviä käytäntöjä sekä tarjoavat pitkäaikaista tukea ja yhteistyötä.

# Kestävien ICT-hankintojen tueksi

- Kriteeripankki
  - Hankintalain mukaisia, syrjimättömiä ja vertailukelpoisia kriteereitä
  - Sopivat nykyiseen markkinatilanteeseen kannusten samalla kehittymään
  - Hankkijan pitää muokata kriteeripankin kriteerit juuri kyseiseen hankintaan sopiviksi
  - [kriteeripankki.fi](https://kriteeripankki.fi)
- ICT-laitteille on julkaistu uudet vastuullisuuskriteerit 2023
  - Kattaa tietokoneet ja tietokonenäytöt
  - Pääpaino laitteiden käyttöiän pidentämisessä. Lisäksi on huomioitu haitallisten aineiden vähentäminen ja energiatehokkuus
  - Sosiaaliselle vastuulle on määritelty vähimmäisvelvoitteet
- Konesali ja pilvipalveluhankinnoille on julkaistu omat vastuullisuuskriteerit
  - Kasvihuonepäästöjä pyritään vähentämään asettamalla konesaleille teknisiä vaatimuksia ja vaikuttamalla energianlähteisiin
  - Palveluiden hiilijalanjälkeä pienentää myös hukkalämmön hyödyntäminen

# MitViDi-hankkeen hankintakriteerit

- MitViDi-hankkeessa kehitettiin arviointikehikko ja kriteeristö julkisten ohjelmistohankintojen ympäristövaikutusten arvioimiseksi.
- Kriteeristöä voi soveltaa sekä julkisiin että yksityisiin ohjelmistohankintoihin
- Myös ohjelmistokehittäjät voivat hyödyntää kriteeristöä ja hankkeen tuloksia

## MitViDi-hankkeen tuotokset

- MitViDi-arviointikehikko on esitelty Ekologisesti kestävien julkisten ohjelmistohankintojen oppaassa (Heinonen (toim.) 2023), joka on saatavilla osoitteessa [www.theseus.fi/handle/10024/806728](http://www.theseus.fi/handle/10024/806728)
- Hankkeen kehitetty työkalu ja muut tuotokset saatavilla osoitteessa [mitvidi.tt.utu.fi](http://mitvidi.tt.utu.fi)

# Kiertotalous ohjelmistokehityksessä

# Kiertotalouden mukainen tuotesuunnittelu

Miten kiertotalouden periaatteita noudattavia suunnittelustrategioita voi noudattaa ohjelmistokehityksessä?

Tuotesuunnittelustrategioita:

1. Pitkäikäisyys
2. Helppo ylläpitää ja korjattavuus
3. Yhteensopivuus
4. Päivitettävyys
5. Modulaarisuus

Lähde: Liiketoimintaa kiertotaloudesta:  
Edelläkävijän opas, Ethica Oy 2018.



# Kiertotalouden periaatteita noudattavia ohjelmistokehityksen käytäntöjä

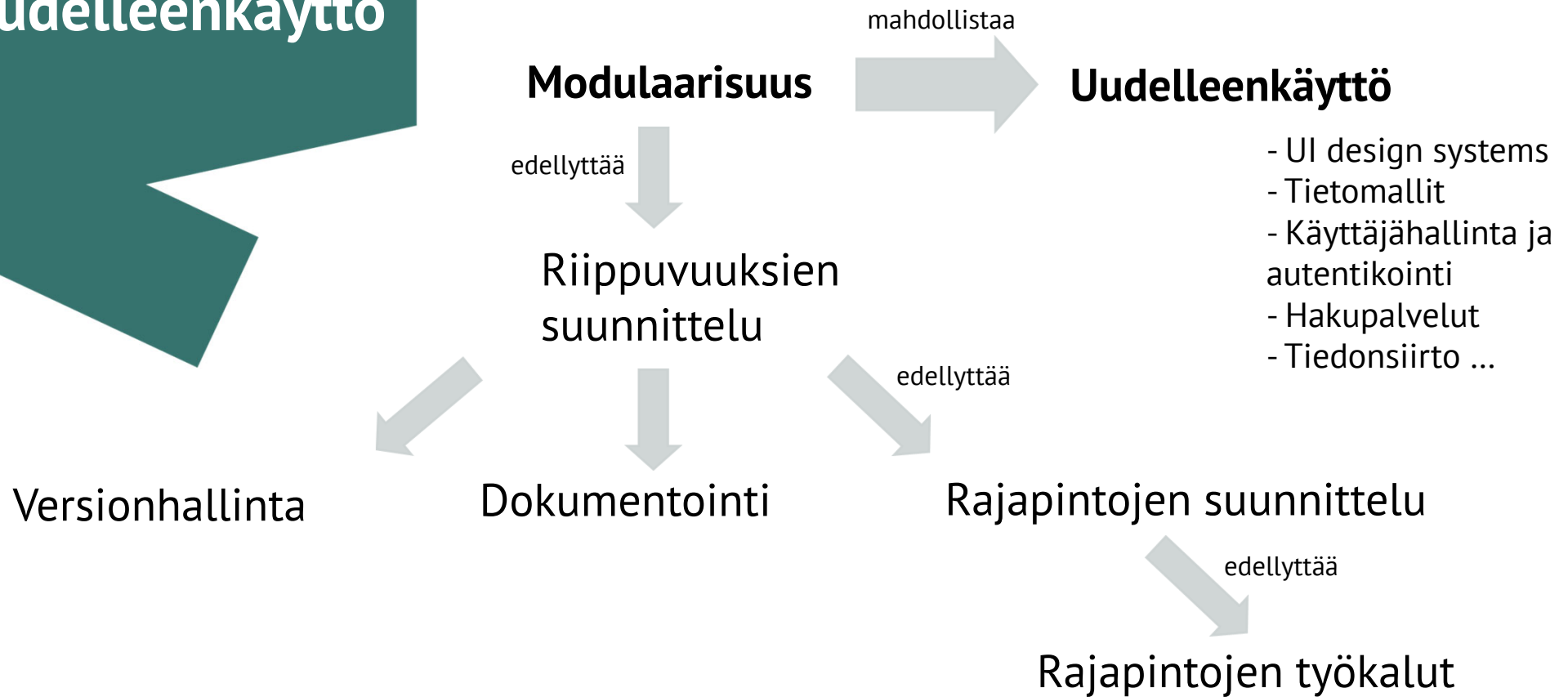
- 1. Ylläpito eli päivitykset ja korjaukset:** Vähentää kokonaan uusien versioiden luomista, kun panostetaan ohjelmistojen jatkuvaan ylläpitoon ja päivityksiin kuten vikojen korjaamiseen, tietoturvapäivityksiin ja toiminnallisiin parannuksiin.
- 2. Modulaarisuus ja uudelleenkäyttö:** Vähentää tarvetta rakentaa kaikki alusta alkaen uudessa projektissa, mikä säästää aikaa, resursseja ja energiaa. Software reuse?
- 3. Avoin lähdekoodi:** Vähentää tarvetta kehittää samanlaista ohjelmistoa uudelleen ja mahdollistaa muutokset toiseen käyttötarkoitukseen. Hyvä/paha?
- 4. Optimointi ja resurssitehokkuus:** Vähentää resursseja käytön aikana eli säästää esimerkiksi laskentatehon tarvetta ja energiaa.

# 1. Päivitykset ja korjaukset

- **Vikoja korjaavat päivitykset:**
  - Tunnista ja korjaa ohjelmistossa olevat virheet ja ongelmat, jotka vaikuttavat sen suorituskykyyn tai käyttökokemukseen.
  - Paranna ohjelmiston luotettavuutta ja toimintaa
- **Tietoturvapäivitykset:**
  - Etsi haavoittuvuuksia
  - Seuraa käyttöä ja liikennettä
  - Tarjoa säännöllisiä tietoturvapäivityksiä ohjelmistoon.
- **Toiminnalliset parannukset:**
  - Kuuntele käyttäjien palautetta ja tarpeita
  - Tee toiminnallisia parannuksia ja uusia ominaisuuksia, jos se parantaa käytettävyyttä, tehokkuutta ja käyttäjäkokemusta.



## 2. Modulaarisuus ja uudelleenkäyttö



# 2. Modulaarisuus ja uudelleenkäyttö

## 2.1. Ohjelmiston riippuvuuksien suunnittelu

- Moduuleille suunnitellaan selkeät tehtävät, vastuut ja niille selkeät rajapinnat
- Dokumentoidaan ja testataan riippuvuuksia
- Käytetään versionhallintaa

### 2.1.1. Ohjelmiston rajapintojen suunnittelu

- Mitä toimintoja moduuli tarjoaa ulospäin? Syöteparametrit, tulokset
- Nimeä toiminnot intuitiivisesti
- Dokumentoi

#### 2.1.1.1. Työkalut

- Swagger (<https://swagger.io/>)
- OpenAPI
- UML-kaavioiden käyttäminen

# Modulaarisuus ja uudelleenkäytettävyys: Tehtäviä opiskelijoille

- Millaisia moduuleja tai kirjastoja olet hyödyntänyt omissa ohjelmistoprojekteissasi? Miten keveitä ja tehokkaita ne ovat? (esim. python pandas)
- Pohdi esimerkkejä itse tekemästäsi ohjelmakoodista, jota on mahdollista uudelleenkäyttää toisessa projektissa (sellaisenaan tai muokattuna).
- Pohdi esimerkkejä, joissa ohjelmiston tai koodin parempi dokumentointi olisi edistänyt sen uudelleenkäyttöä.

# Esimerkki

JSON muotoiseen tiedostoon kirjoittamisen ja lukemisen metodien tekeminen FileService-luokkaan ja sen käyttö useissa projekteissa.

```
# FileService.py

import json

class FileService:
    def __init__(self, filename):
        self.filename = filename

    def read_data(self):
        try:
            with open(self.filename, 'r') as file:
                data = json.load(file)
                return data
        except FileNotFoundError:
            print("Tiedostoa ei löytynyt.")
            return None

    def write_data(self, data):
        with open(self.filename, 'w') as file:
            json.dump(data, file)
            print("Data tallennettu tiedostoon.")
```

```
from FileService import FileService

file_service = FileService("data.json")

# Tallennetaan data tiedostoon
data_to_save = {"nimi": "Esimerkki", "ikä": 30}
file_service.write_data(data_to_save)

# Luetaan data tiedostosta
loaded_data = file_service.read_data()
if loaded_data:
    print("Ladattu data:", loaded_data)
```

# Tehtäviä opiskelijoille

Millaisia kiertotaloutta ja kestävyyttä edistäviä asetuksia tai toimenpiteitä voisit tehdä käyttämälläsi kehitysteknologioilla?

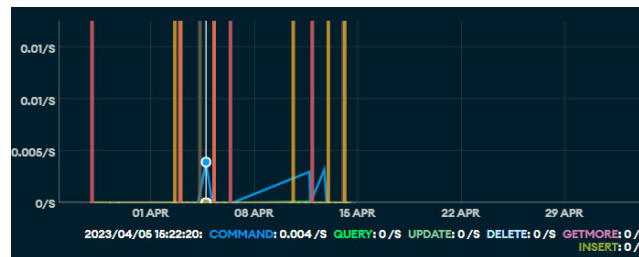
Tutustu esimerkiksi seuraaviin:

- Visual Studio Code
- MongoDB
- ....

# Esimerkki MongoDB tietokanta

**Käytä resurssit mahdollisimman tehokkaasti;** suorituskyky vs. tallennustilan säästö

- Monitoroi tietokannan käyttöä ja useimmin suoritettuja kyselyjä
- Optimoi tietomallia
- Etsi tasapaino kenttien indeksoinnissa
- Pakkaa verkon yli lähetettävä data (zlib, snappy)



Esimerkkejä: Blogiapplikaation tietokannan luku- ja kirjoitusoperaatioiden aiheuttama kuormitus MongoDBssä  
Lähde: <https://www.mongodb.com/docs/manual/data-modeling/schema-design-process/identify-workload/#std-label-data-modeling-identify-workload>

```
Atlas Sandbox-shard-0 [primary] KimmoDB> db.library.find({city: "Turku"}).explain("executionStats")
{
  executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 1,
    totalKeysExamined: 0,
    totalDocsExamined: 26,
    executionStages: {
      stage: 'COLLSCAN',
      filter: { city: { '$eq': 'Turku' } },
      nReturned: 2,
      executionTimeMillisEstimate: 1,
      works: 27,
      advanced: 2,
      needTime: 24,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
    }
  }
}
```

```
Atlas Sandbox-shard-0 [primary] KimmoDB> db.library.aggregate([ { $indexStats: { } } ] )
[
  {
    name: '_id_',
    key: { _id: 1 },
    accesses: { ops: Long("2"), since: ISODate("2024-04-04T06:12:13.345Z") },
    host: 'sandbox-shard-00-02.sxv.mongodb.net:27017'
  }
]
```

# Kestävän kehityksen ohjeistuksia ohjelmistokehitykseen

## Ohjelmistokehityksen vihreitä käytäntöjä ovat esimerkiksi:

- Käytä tummia värejä ja rajoita animaatioita
- Älä hae `SELECT * FROM` koko taulun sisältöä
- Poista MySQL- ja MariaDB-binäärilokit käytöstä
- Korvaa `$i++` syntaksi `++$i`
- Pienennä verkkosivujen informaatioarkkitehtuurin syvyyttä
- Parempi 1 palvelin täydellä käytöllä kuin 4 vajaalla

# The World Wide Web Consortium (W3C) ohjeistus (päivitetty 4.3.2024)

- Web Sustainability Guidelines (WSG) 1.0
  - Sisältää 93 ohjetta kestävämpään (web)ohjelmistotuotantoon
    - Business & Product Strategy
    - UX
    - Web Development
    - Hosting & Infrastructure
  - Ohje, kriteerit, vaikutus- ja kustannusarviot, esimerkkejä ja linkkejä, tukee raportointia
    - **Impact:** Low: Quick sustainability wins. Medium: Noticeable sustainable impact. High: Significant long-term benefit.
    - **Effort:** Low: Minimal implementation. Medium: Some changes needed. High: Heavy refactoring required.
- Draft Community Group Report 04 March 2024
- Viimeisin ja täydellinen versio: <https://w3c.github.io/sustyweb/>
- Suppeampi, käyttäjäystävällinen versio löytyy mm: <https://sustainablewebdesign.org/>

# Kestävän UX suunnittelun ohjeistus

- Kestävän UX suunnittelun peruseriaatteet:
  - Pohdi pitääkö digitoida
    - <https://tarotcardsoftech.artefactgroup.com/>
  - Aseta ympäristölle budjetti
    - <https://www.performancebudget.io/>
  - Suunnittele ympäristöä varten
    - <https://greentheweb.com/mother-nature-as-a-persona-for-more-sustainability-in-user-research-incl-template/>
  - Luo mutkattomia käyttöpolkuja
    - <https://lowwwwcarbon.com/>
- Kansainvälinen verkosto The Sustainable UX Network (SUX) tarjoaa ohjeita, kursseja ja muita sisältöjä kestävästä UX suunnittelusta
  - Valmistelee tarkempaa ohjeistusta (Playbook)
  - Perustettu 2021
  - Tutustu lisää osoitteessa <https://sustainableuxnetwork.com/>

# Tehtäviä opiskelijoille

- Poimi jokin käytäntö WSG-sivuilta (Web Sustainability Guidelines).
- Pohdi pystytkö toteuttamaan käytäntöä nykyisessä projektissasi ja miten sen tekisit?
- Tarvitsetko lisää osaamista tai lisätietoa?
- Toteuta käytäntö projektissasi.

# Opiskelijoiden tiedot, taidot ja ymmärrys vihreän ohjelmistokehityksen käytännöistä

# Tutkimuskyselyn kuvaus

- Lähde: Tarkkanen, K. (2024). Green Software Engineering Practices: Familiarity, Skills and Understandability among Mid-Bachelor ICT students. <http://dx.doi.org/10.54941/ahfe1005040>
- Tutkimuksen tarkoitus oli arvioida vihreiden ohjelmistosuunnittelukäytäntöjen tuntemusta, taitoja ja ymmärrystä ICT-opiskelijoiden keskuudessa sekä ehdottaa koulutusinterventioita vihreiden käytäntöjen ymmärtämisen ja soveltamisen parantamiseksi.
- **Osallistujat:** 40 ICT-opiskelijaa, keskimäärin 125 opintopistettä.
- **Kyselyn sisältö:** Esiteltiin 154 vihreää ohjelmistokäytäntöä, jotka on hankittu kolmesta verkkolähteestä.
  - Bordage, F. (2019). Ecoconception web : les 115 bonnes pratiques, 3e edition. Eyrolles. In English: Web ecodesign: 115 good practices, 3rd edition, published in April 2019. Retrieved from <https://greenit.fr> English version is available at <http://www.ecometer.org/rules/> and in <https://github.com/cnumr/best-practices>
  - Debomy, P. (2020). How to make a Green IT/Low-carbon Website? Retrieved from the website: <https://pasomy.medium.com/how-to-make-a-green-it-low-carbon-website-8062f550a1f7>
  - <https://sustainablewww.org> (valitettavasti sivu on poistunut käytöstä)

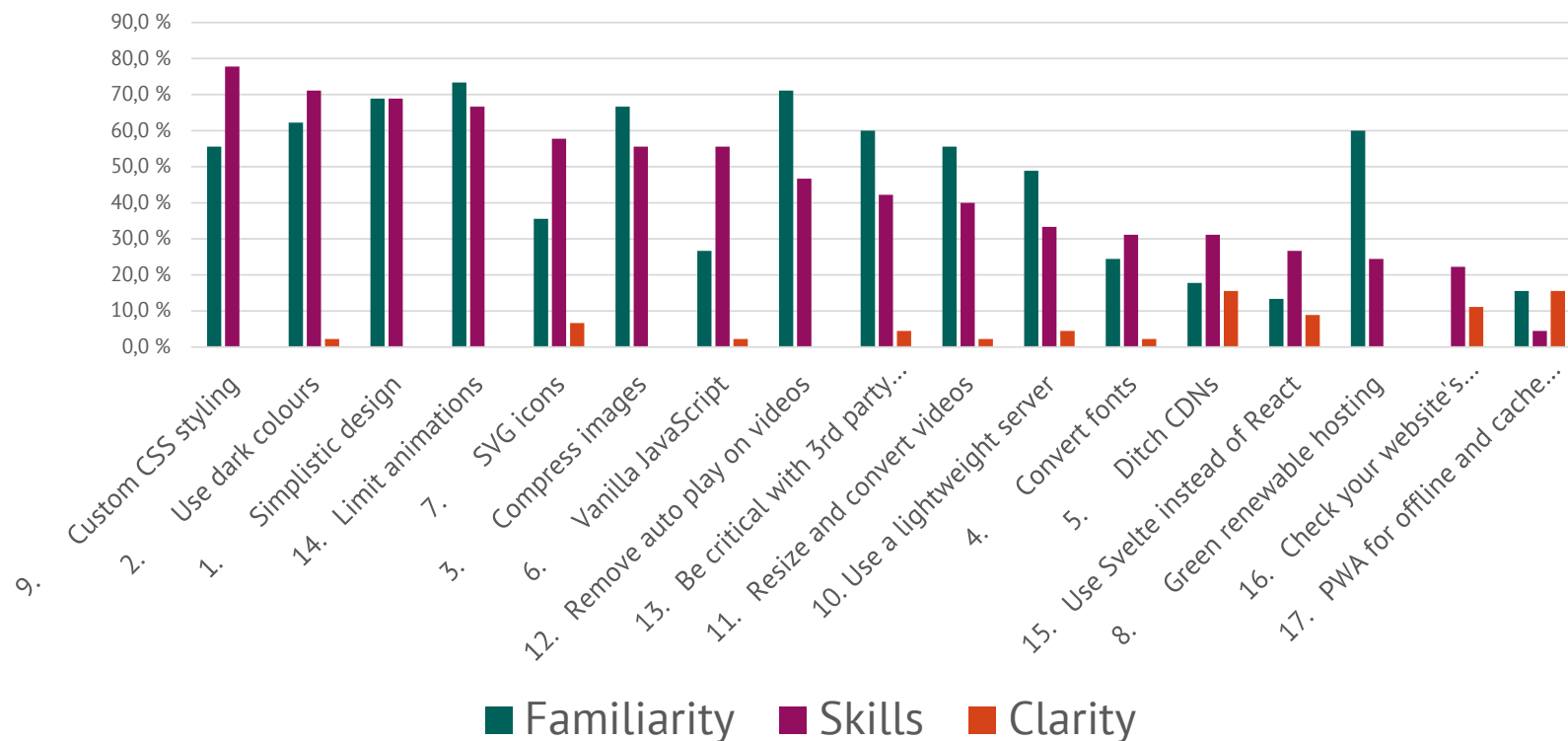
# Menetelmälista 1: Osatuimmat menetelmät

Osatuimmat menetelmät (skills)

Vihreän ohjelmistokehityksen kriteerien lähde: sustainablewww.org

Vastaajia: 45

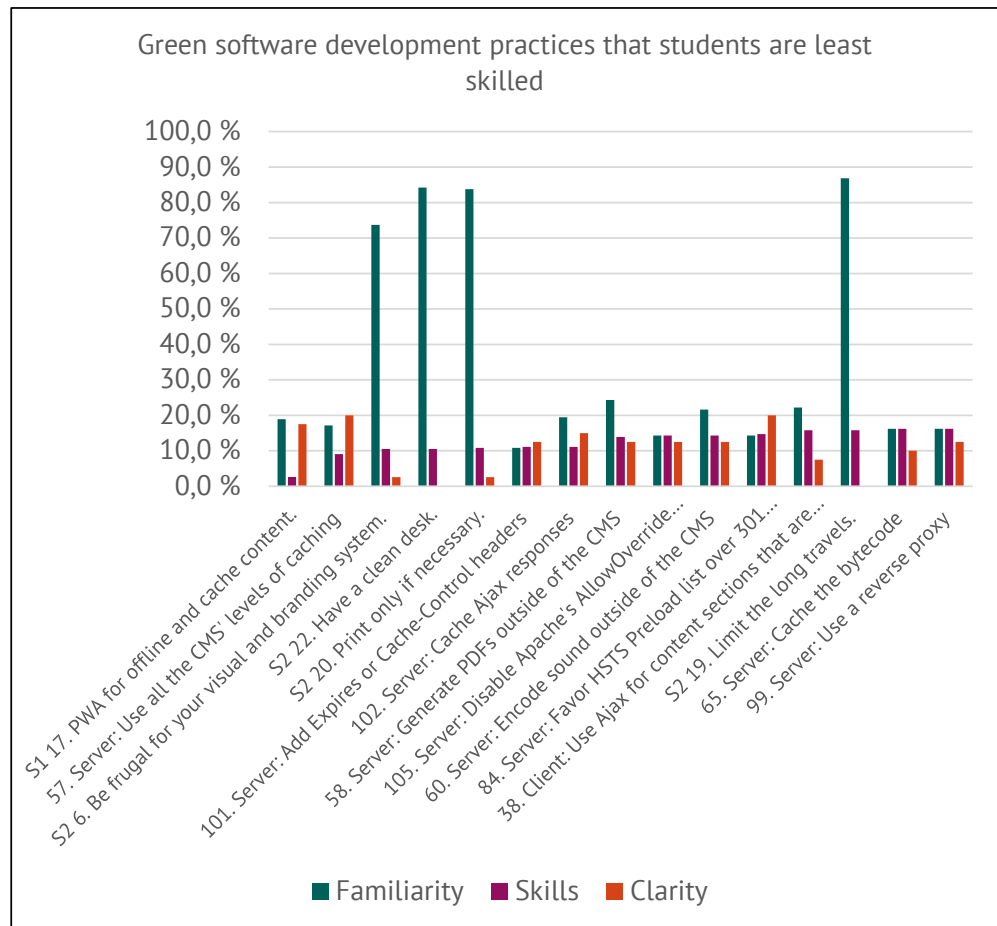
Keskimäärin 125 op suoritettuja tietotekniikan insinöörin tai tietojenkäsittelyn tradenomin opintoja



# Vähiten osatut käytänteet

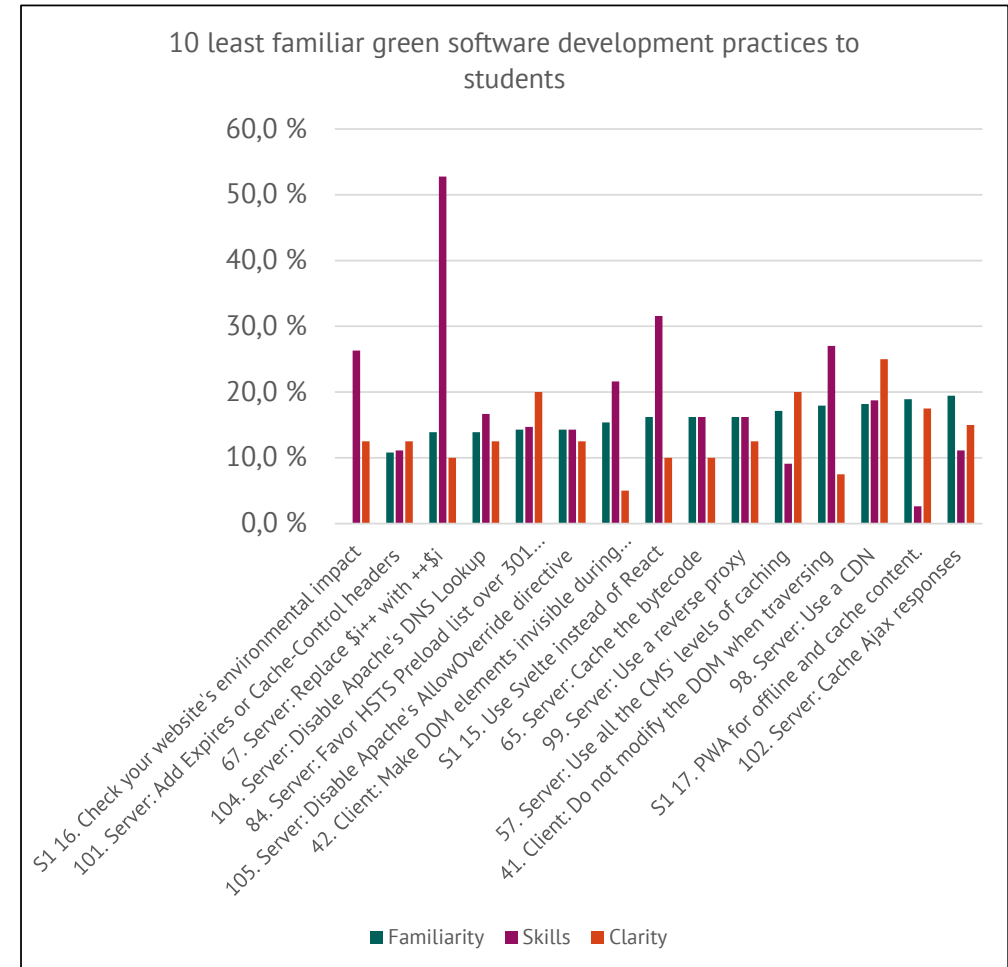
## Yhteenveto:

- 1) Palvelinpuolen toiminnot, kuten "Käytä kaikkia CMS:n välimuistitasoja"
- 2) Järjestelmä- tai teknologiakohtaiset toiminnot, kuten "Poista Apachen AllowOverride-direktiivi käytöstä",
- 3) Lyhenteet, kuten PWA, CMS ja HSTS (viisi epäselvintä käytäntöä sisälsivät jonkin lyhenteen)



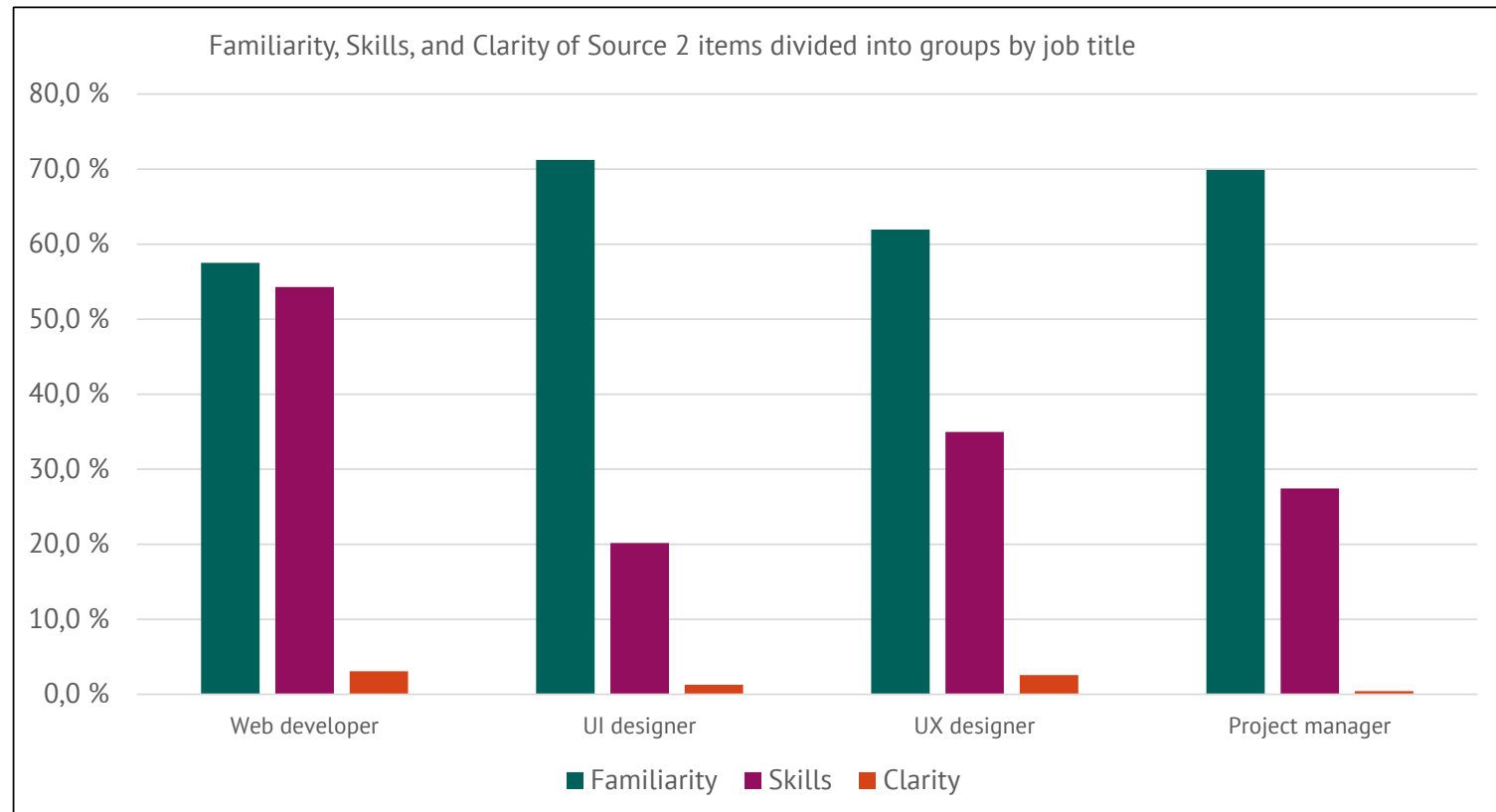
# Vähiten tunnetut käytänteet

- Tuntemattomin käytäntö, (0 %; kukaan ei tuntenut), oli "Tarkista verkkosivustosi ympäristövaikutukset". Kuitenkin yli 25 % osallistujista vastasi, että heillä on taitoja tämän käytännön toteuttamiseen.
- Todennäköisesti opiskelijat eivät ole kuulleet verkkosivustojen vaikutusten arviointityökaluista, kuten Google Lighthousesta tai ecograder.com palveluista, jotka olivat melko uusia työkaluja, mutta silti uskovat, että heillä on taitoja käyttää näitä työkaluja ohjelmistoprojekteissaan.



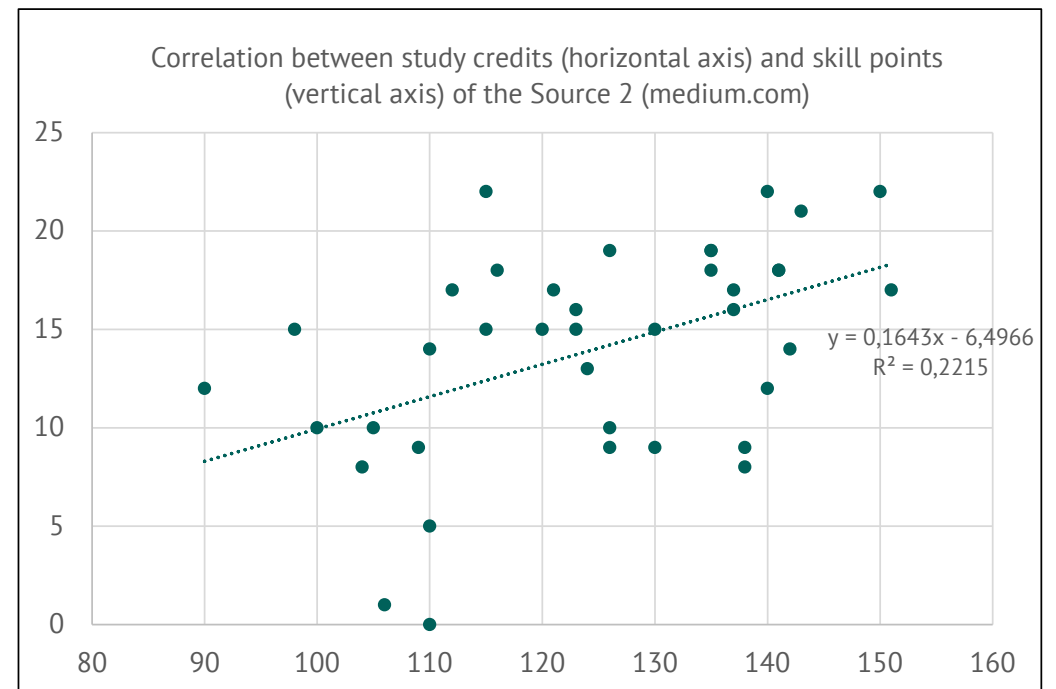
# Menetelmälista 3

- Eri käytännöt on ryhmitelty työnkuvan mukaisesti
- Käyttöliittymäsuunnittelijan vihreät käytännöt ovat tuttuja, mutta osaamista ei ole



# Vertailuja

	Source 1 (sustainabl eww.org)	Source 2 (medium.com)	Source 3 (greenit.fr)
Average skills + familiarity points (weighted)	153,77	190,23	137,83
St. deviation	49,21	74,91	70,05
Margin of error	15,74	23,96	22,40



# Johtopäätöksiä

- Yksinkertainen kysely voi toimia lähtökohtana kestävyystaitojen kehittämiseksi tutkimuksen kohteena olleessa ja vastaavissa opinto-ohjelmissa, kurssisisällöissä ja käytännön tehtävissä.
- Järjestelmä- ja teknologiakohtaisia kestävyyskäytäntöjä tulisi mukauttaa ja soveltaa paikallisesti käytössä oleviin teknologioihin
- Hyvin tunnettuja UI-käytäntöjä tulisi ottaa opetuksessa myös harjoittelun kohteeksi ainakin opetussuunnitelman myöhemmissä opinnoissa (>120 op).

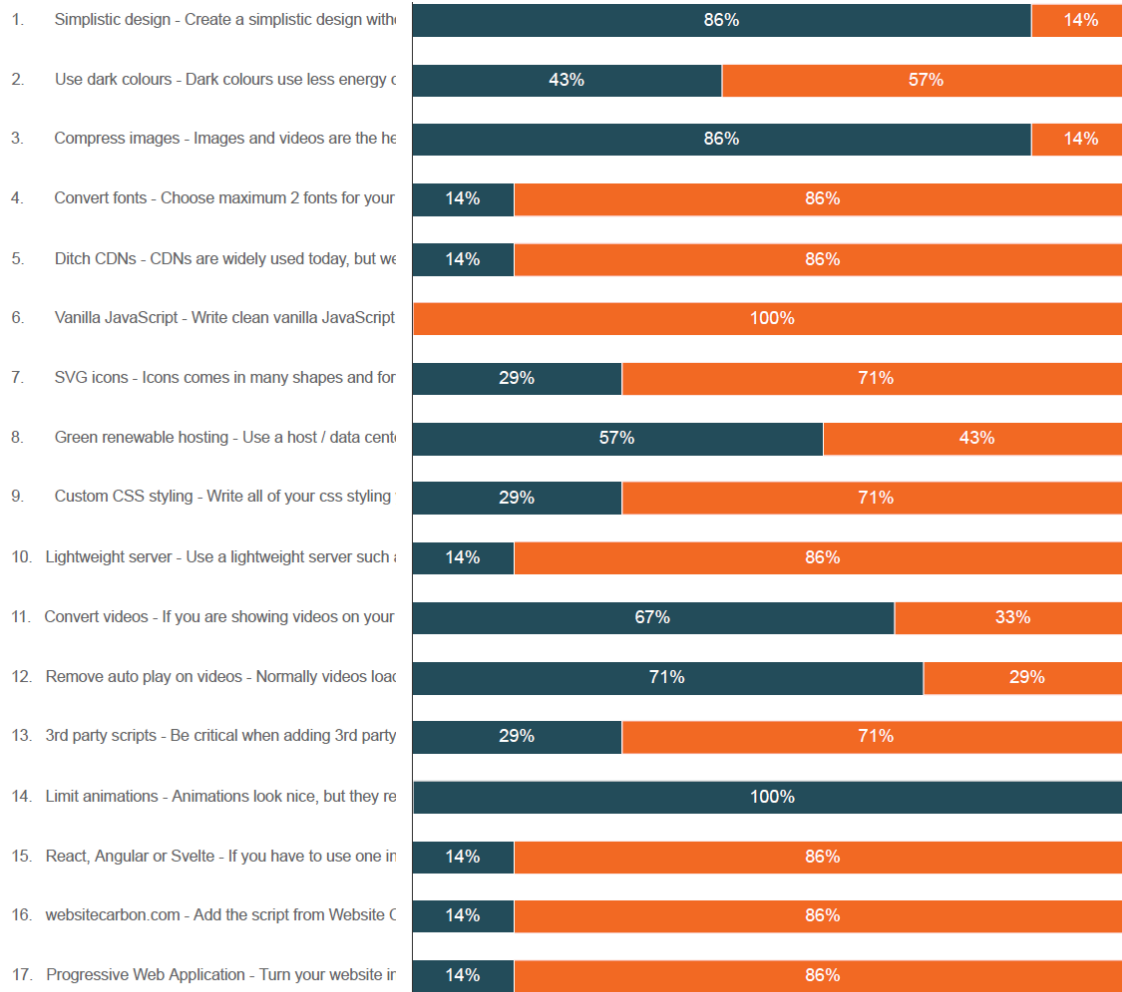
# Ennakkotehtävän tuloksia

# Ennakkotehtävä 1

- Ensimmäisenä ennakkotehtävänä annettiin tehtäväksi täyttää vastaava kysely kuin opiskelijoille edellä

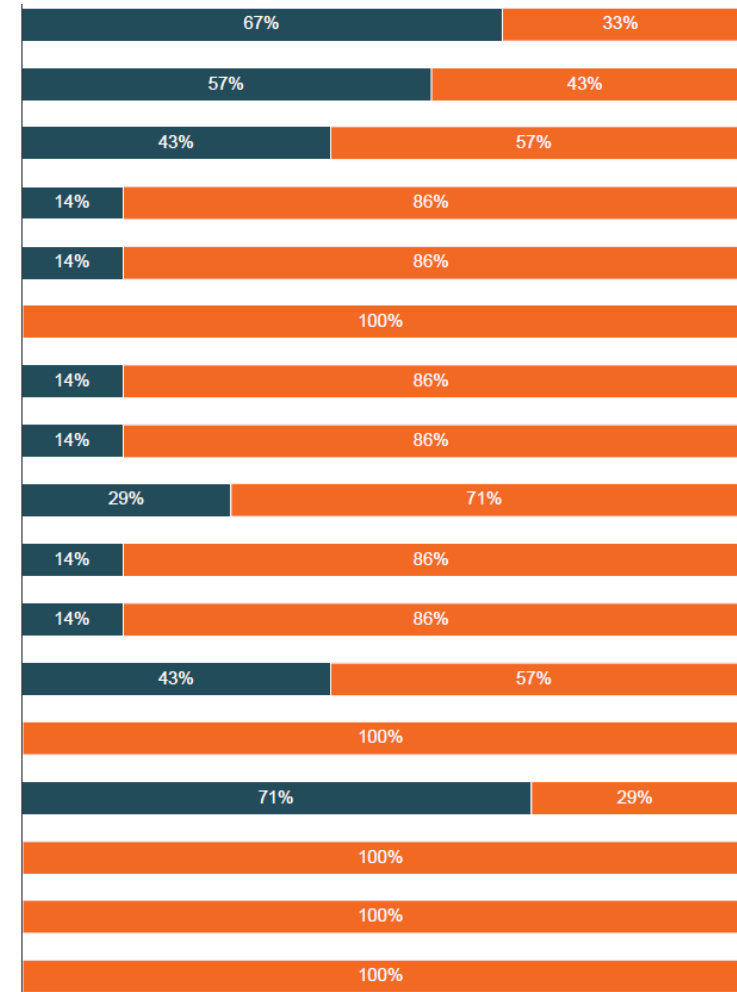
# Ennakkotehtävän tulokset

## Tiedot



● Yes ● No

## Taidot



● Yes ● No

# Omat hyvät käytänteet ohjelmistojen kiertotalouteen?

# Ennakkotehtävä 2

- Toisena ennakkotehtävänä annettiin tehtäväksi etsiä erilaisia vihreän ohjelmistotuotannon käytäntöjä ja linkittää näitä keskustelupalstalle

# Millaisia listoja löydettiin?

## Esimerkiksi:

- <https://www.cgi.com/fi/fi/blogi/mita-green-coding-on>
- <https://www.lut.fi/fi/artikkelit/maailma-tarvitsee-tehokkaita-ohjelmistoja-mutta-kenen-vastuulle-kestava>
- <https://www.sytyke.org/kestava-kehitys/vihrea-koodi/>
- <https://tivia.fi/uutiset/uutiset-3/vihrea-koodaus-pelastaa-planeetan-1871>
- <https://www.thoughtworks.com/insights/blog/sustainability/illustrated-guide-green-software-engineering>

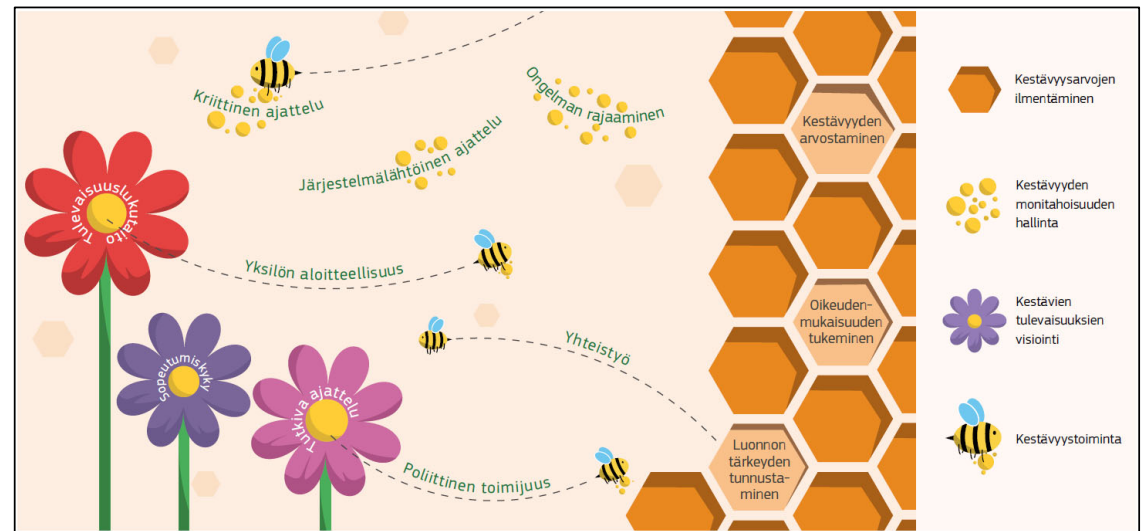
Keskustelua ja esimerkkejä löydetyistä käytännöistä

- Käytäntöjen fokuksen erilaisuus?
- Eroja ja yhtäläisyyksiä käytäntölistoissa??

# Kurssitehtävän suunnittelua

# Oman oppimateriaalin suunnittelun innoittajia

- GreenComp, Kestävää kehitystä koskeva eurooppalainen osaamiskehys: <https://op.europa.eu/fi/publication-detail/-/publication/bc83061d-74ec-11ec-9136-01aa75ed71a1>
  - Eficoden SE pipeline työkalu: <https://pipelinegame.eficode.com/dashboard>
  - Tivoli utopia -peli: <https://kulttuurinvuosikello2.fi/tivoliutopia/>
- OPPIMISPELI



**Kuva:** GreenComp-osaamiskehyn visuaalinen esitys © Daniel N. Buxton, teoksessa Kestävää kehitystä koskeva eurooppalainen osaamiskehys Bianchi, G., Pisiotis, U., Cabrera Giraldez, M. GreenComp – Eurooppalainen kestävä kehityksen osaamiskehys. Bacigalupo, M., Punie, Y. (toimittajat), EUR 30955 FI, Euroopan unionin julkaisutoimisto, Luxemburg, 2022; ISBN 978-92-76-53203-3, doi: 10.2760/595099JRC128040, <https://op.europa.eu/publication-detail/-/publication/bc83061d-74ec-11ec-9136-01aa75ed71a1>, CC-BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

# Esimerkki: Oppimispelin tavoite ja idea

- Opiskelija(ryhmät) rakentavat itselleen kestävän kehityksen mukaisen web-ohjelmistokehityksen mallin/polun toimenpiteineen.
- Opiskelijat tutustuvat monipuolisesti erilaisiin kestäviin web-kehityskäytäntöihin lukemalla toimenpidekortteja.
- Kortin kohdalla opiskelijat pohtivat ja arvottavat kortin esittämän toimenpiteen ympäristövaikutusta (impact) ja resurssikustannusta (effort)
- Pelin tavoitteena on kilpailla muita opiskelijoita tai opiskelijaryhmiä vastaan siitä kenellä on ympäristöystävällisin kehitysmalli



Kuva. Toimenpidekortteja luotuna Miro-työkalulla

# Esimerkki toimenpidekortista

Etupuoli

## 2.17 Take a More Sustainable Approach to Animation

Animation can be both CPU and GPU intensive and have implications for accessibility. While visually appealing and useful in certain situations, care and attention should be taken when considering the use of a high emissions' technology.

Taustapuoli

### Impact & Effort

**Impact** Medium

**Effort** Low

### How to:

<https://w3c.github.io/sustyweb/#take-a-more-sustainable-approach-to-animation>

- Yksi kortti esittää yhtä kehitystoimenpidettä  
Toimenpiteet kerätään täältä:  
<https://w3c.github.io/sustyweb/intro.html#the-web-sustainability-guidelines>
- Kääntöpuolella tieto vaikutuksesta ja kustannuksista pisteiden laskua varten
  - Vaihtoehtoisesti pisteet voidaan laskea erillisen taulukon avulla

# Pelin kulku

- Esille laitetaan pelaajien määrän verran kortteja
- Opiskelijat nostavat uusia kortteja ja/tai vaihtavat vanhoja kortteja uusiin kiertävässä järjestyksessä
- Kortin kohdalla opiskelijat pohtivat ja arvottavat kortin esittämän toimenpiteen ympäristövaikutusta (impact) ja resurssikustannusta (effort) ja tekevät päätöksen nostettavasta kortista
- Lopuksi lasketaan kestävyyspisteet korttien kääntöpuolelta esim.
  - Impact: Low: 1 piste, Medium 2 pistettä, High: 3 pistettä
  - Effort: Low 3 pistettä, Medium 2 pistettä, High 1 piste
- Korkeimmat pisteet saanut voittaa
- Vertaillaan ja keskustellaan erilaisista kehityspoluista
- Ensimmäinen versio (kopioi ja kehitä edelleen!):  
[https://miro.com/app/board/uXjVKWvtexQ=?share\\_link\\_id=680286401518](https://miro.com/app/board/uXjVKWvtexQ=?share_link_id=680286401518)

# Esimerkki: Pelialue Mirossa

