

## L14 Tehtävät

- Hyötyohjelmia auton käyttökulujen laskemiseen, salasanojen luomiseen, datan hakemiseen Internetistä ja datan analysointiin
- Tehtävät tulee ratkaista itse tehdyillä algoritmeilla, ratkaisuissa saa käyttää vain kurssilla aiemmin käytettyjä kirjastoja ja tehtävässä 2 ja 3 erikseen mainittuja kirjastoja.

Pääosa tämän viikon tehtävistä on kertaustehtäviä, joihin liittyvät asiat löytyvät oppaasta. Huomaa, että ohjelmat eivät ole "vaikeusjärjestyksessä" vaan ne ovat erilaisia tehtäviä. Ohjelmointitehtävissä on oltava otsikkotiedot ja ne palautetaan Moodlen kautta CodeGradeen.

## Sisällysluettelo

<b>L14T1: Auton matkakustannuslaskuri (M)</b>	<b>1</b>
<b>L14T2: Salasanageneraattori (P)</b>	<b>2</b>
<b>L14T3: Datan hakeminen Internetistä: SpaceX-laukaisut (T)</b>	<b>4</b>
<b>L14T4: Vuokra-asuntojen vuokrien analyysi (lisätehtävä) (P)</b>	<b>6</b>

## L14T1: Auton matkakustannuslaskuri (M)

*Tehtävän taso: Minimitaso*

Tee Python-ohjelma, joka kysyy käyttäjältä tarvittavat tiedot ja laskee niiden avulla tuloksen. Tee ohjelmaan ensin valikko, josta käyttäjä voi valita haluamansa auton käyttövoiman (benssiini, sähkö tai kaasu). Kysy sen jälkeen jokaisessa valinnassa matkan pituus kilometreinä, auton keskikulutus (per 100 km) sekä käyttövoiman hinta euroina. Käytä oikeita yksiköitä oikeassa valinnassa esimerkkiajo mukaisesti. Laske sen jälkeen tulos käyttäen alla olevaa kaavaa:

$\text{Matka}/100 * \text{Kulutus} * \text{Hinta}$

Tulosta tulos kahden desimaalin tarkkuudella. Huom. pakota tässä tehtävässä tulokseen kaksi desimaalia käyttäen esimerkiksi `format()`-funktia.

Koska valintojen tekstit näyttävät hyvin samoilta, voi ne tehdä yhdessä aliohjelmassa, joka saa parametrina oikean käyttövoiman ja yksikön merkkijonona. Tee ohjelmaasi myös normaalin kurssin käytännön mukaiset pääohjelma ja valikko -aliohjelmat.

## Esimerkkiajo 1

*Syötteet:*

```
1
50
6.5
1.950
2
50
15
0.43
3
```

50  
3.9  
2.05  
0

***Tuloste:***

Tämä ohjelma laskee auton matkakustannuksia eri käyttövoimilla.

Valitse auton käyttövoima:

- 1) Bensiini
- 2) Sähkö
- 3) Maa- tai biokaasu
- 0) Lopeta

Anna valintasi: 1

Anna matkan pituus kilometreinä: 50

Anna keskimääräinen kulutus (litra per 100 km): 6.5

Anna bensiinin hinta euroina per litra: 1.950

50.0 kilometrin matkan hinnaksi muodostuisi noin 6.34 euroa.

Valitse auton käyttövoima:

- 1) Bensiini
- 2) Sähkö
- 3) Maa- tai biokaasu
- 0) Lopeta

Anna valintasi: 2

Anna matkan pituus kilometreinä: 50

Anna keskimääräinen kulutus (kWh per 100 km): 15

Anna sähkön hinta euroina per kWh: 0.43

50.0 kilometrin matkan hinnaksi muodostuisi noin 3.23 euroa.

Valitse auton käyttövoima:

- 1) Bensiini
- 2) Sähkö
- 3) Maa- tai biokaasu
- 0) Lopeta

Anna valintasi: 3

Anna matkan pituus kilometreinä: 50

Anna keskimääräinen kulutus (kg per 100 km): 3.9

Anna kaasun hinta euroina per kg: 2.05

50.0 kilometrin matkan hinnaksi muodostuisi noin 4.00 euroa.

Valitse auton käyttövoima:

- 1) Bensiini
- 2) Sähkö
- 3) Maa- tai biokaasu
- 0) Lopeta

Anna valintasi: 0

Lopetetaan.

Kiitos ohjelman käytöstä.

## L14T2: Salasanageneraattori (P)

*Tehtävän taso: Perustaso*

Toteuta Pythonilla ohjelma, joka luo satunnaisia salasanoja käyttäjän valitsemasta joukosta. Käyttäjä voi valita kirjainten, kirjainten ja numeroiden sekä kirjainten, numeroiden ja erikoismerkkien joukoista. Kun käyttäjä on valinnut joukon, kysy käyttäjältä luotavan salasanan pituus. Joukkojen osat saadaan helposti kiintoarvoina käyttämällä `string`-kirjastoa:

```
KIRJAIMET = string.ascii_letters
```

```
NUMEROT = string.digits
```

```
ERIKOISMERKIT = string.punctuation
```

Kaikki kolme ovat merkkijonoja, jotka sisältävät nimensä mukaiset merkit yhdessä pötkössä ilman välejä tai välimerkkejä. Kirjaimet sisältävät sekä pienet että isot kirjaimet A-Z.

Tee ohjelmaasi normaalin kurssin käytännön mukaiset pääohjelma ja valikko -aliohjelmat. Salasana kannattaa tehdä omassa aliohjelmassaan, joka saa parametrina valitun joukon. CodeGrade-tarkastusta varten, tee ohjelmassasi lisäksi seuraavat asiat:

1. Käytä tässä tehtävässä `random`-kirjastoa. Tämän tehtävän voi toteuttaa käyttämällä merkkijonon indeksia ja `randint()`-funktia, tai ottaa merkit suoraan `choice()`-funktion avulla. Molempien funktioiden käytöstä on esimerkki ohjelmointioppaan 8. luvussa.
2. Käytä `random.seed()`-funktion arvona lukua 8989. Tämä tulee laittaa pääohjelman alkuun, jolloin se suoritetaan kerran.
3. Valinnan 1 joukko on vain KIRJAIMET
4. Valinnan 2 joukko on KIRJAIMET + NUMEROT (huomaa järjestys)
5. Valinnan 3 joukko on KIRJAIMET + NUMEROT + ERIKOISMERKIT (huomaa järjestys)

**HUOM!** Oikeassa elämässä tulisi käyttää `random`-kirjaston sijaan `secrets`-kirjastoa, jolla saamme kryptograafisesti turvallisempia arvoja.

## Esimerkkiajo 1

### Syötteet:

```
1
10
2
8
3
12
0
```

### Tuloste:

```
Tämä ohjelma luo satunnaisia salasanoja.
Luo salasana käyttäen:
1) Kirjaimia
2) Kirjaimia ja numeroita
3) Kirjaimia, numeroita ja erikoismerkkejä
0) Lopeta
Valintasi: 1
Anna luotavan salasanan pituus: 10
Luotiin 10 merkkiä pitkä salasana:
kDoesgeamh

Luo salasana käyttäen:
```

```
1) Kirjaimia
2) Kirjaimia ja numeroita
3) Kirjaimia, numeroita ja erikoismerkkejä
0) Lopeta
Valintasi: 2
Anna luotavan salasanan pituus: 8
Luotiin 8 merkkiä pitkä salasana:
1GDWgndj

Luo salasana käyttäen:
1) Kirjaimia
2) Kirjaimia ja numeroita
3) Kirjaimia, numeroita ja erikoismerkkejä
0) Lopeta
Valintasi: 3
Anna luotavan salasanan pituus: 12
Luotiin 12 merkkiä pitkä salasana:
8lTC0%=;v:1"

Luo salasana käyttäen:
1) Kirjaimia
2) Kirjaimia ja numeroita
3) Kirjaimia, numeroita ja erikoismerkkejä
0) Lopeta
Valintasi: 0
Lopetetaan.

Kiitos ohjelman käytöstä.
```

## L14T3: Datan hakeminen Internetistä: SpaceX-laukaisut (T)

*Tehtävän taso: Tavoitetaso*

Tee Python-ohjelma, joka hakee (epävirallisesta) SpaceX-rajapinnasta rakettien ja miehitettyjen lentojen laukaisudataa. Suodata tiedot niin, että käyttäjä voi katsoa vain miehitettyjen lentojen tietoja.

Tätä tehtävää varten käytämme Launches V5 ja Crew V4 rajapintoja, joiden dokumentaatiot löytyvät täältä kiinnostuneille:

- <https://github.com/r-spacex/SpaceX-API/blob/master/docs/launches/v5/one.md>
- <https://github.com/r-spacex/SpaceX-API/blob/master/docs/crew/v4/one.md>

Ohjelmasi alussa, kysy käyttäjältä vuosi, jolta miehitettyjä lentoja etsitään. Hae sen jälkeen kaikki laukaisut rajapinnan verkko-osoitteesta (<https://api.spacexdata.com/v5/launches>). Valitse datasta ne laukaisut, joiden aikaleimassa ("date\_utc") on käyttäjän antama vuosi JA "success" -kentässä on arvo True. Valitse näistä laukaisuista ne, joiden "crew" kenttä ei ole tyhjä lista (eli laukaisu on miehitetty). Jos vuodella ei ole miehitettyjä laukaisuja, lopeta ohjelman suoritus hallitusti.

Näytä sen jälkeen käyttäjälle valikko, jossa otsikkona on lennon aikaleima ("date\_utc") sekä sen nimi ("name"). Kysy käyttäjältä minkä lennon tiedot hän haluaa katsoa. Näytä käyttäjän valitsemasta lennosta seuraavat tiedot esimerkiajon mukaisesti:

- Nimi ("name")

- Lennon numero ("flight\_number")
- Päivämäärä ("date\_utc")
- Kuvaus ("details") HUOM. Jos kuvaus on tyhjä, tulosta (Puuttuu)
- Laukaisun Wikipedia-artikkelin linkki ("links" -> "wikipedia")
- Miehistön koko ("crew" listan pituus)

Laukaisun tietojen jälkeen, hae miehistön tiedot. "crew" -listassa on miehistön jäsenen ID ja rooli. Käytä ID:tä hyväksesi ja hae sen avulla Crew-rajapinnasta jäsenen tiedot. Rajapinnan osoite on [https://api.spacexdata.com/v4/crew/\[id\]](https://api.spacexdata.com/v4/crew/[id]) jossa [id] on jäsenen ID. Näytä jäsenistä seuraavat tiedot:

- Rooli ja nimi (laukaisun datasta "crew" -> "role" ja jäsenen tiedoista "name")
- Taho, jota jäsen edustaa (jäsenen tiedoista "agency")
- Jäsenen Wikipedia-artikkeli (jäsenen tiedoista "wikipedia")

Tietojen näyttämisen jälkeen tulosta valikko uudelleen, jotta käyttäjä voi valita toisen lennon. Virhe-ilmoitukset, jotka eivät näy esimerkkitulosteissa:

"Tiedonhaussa virhe, lopetetaan."

"Vuodelta ei löytynyt miehitettyjä lentoja."

## Vinkkejä

1. Tässä tehtävässä käsitellään verkkosivuja, joten tarvitsemme `urllib`-kirjastoa. Ohjelmointioppaan 8. luvussa on esimerkki miten kirjastoa käytetään.
2. Rajapinta palauttaa JSON-muotoista dataa. Käytä siis `json`-kirjastoa datan muuntamiseen käsiteltävään muotoon. Voit laittaa rajapinnan osoitteen selaimeesi, jolloin näet mitä rajapinta palauttaa.
3. Koska verkkosivuja käsiteltäessä voi helposti tapahtua virheitä tiedonsiirrossa, käytä verkkosivun hakemisessa virheenkäsittelyä. Tässä tehtävässä käy myös `OSError`, sillä kirjaston yleensä antama `URLError` on `OSError`-virheen alla. Mikäli tiedonhaku epäonnistuu, lopeta ohjelma `sys.exit(0)`-käskyllä.
4. Ohjelma kannattaa pilkkoa helpommin käsiteltäviin aliohjelmiin, kuten verkkosivun hakemiseen, tietojen erotteluun, miehistön tietojen hakuun, tietojen tulostamiseen, valikon tulostamiseen ja pääohjelmaan.
5. Rajapintojen osoitteet voi laittaa ohjelmaan kiintoarvoina.

## Esimerkkiajo 1

### Syötteet:

```
2022
2
0
```

### Tuloste:

```
Tervetuloa SpaceX miehitettyjen lentojen selaimeen.
Anna etsittävä vuosi: 2022
Tiedot haettu, yhteensä 205 laukaisua.
```

Valitse tarkasteltava laukaisu:

- 1) 2022-04-08T15:17:00.000Z - Ax-1
- 2) 2022-04-27T07:52:00.000Z - Crew-4
- 3) 2022-10-05T16:00:00.000Z - Crew-5
- 0) Lopeta

Valintasi: 2

::LENNON TIEDOT::

Nimi: Crew-4

Lennon numero: 159

Päivämäärä: 2022-04-27T07:52:00.000Z

Kuvaus: (Puuttuu)

Laukaisun Wikipedia-artikkeli: [https://en.wikipedia.org/wiki/SpaceX\\_Crew-4](https://en.wikipedia.org/wiki/SpaceX_Crew-4)

Laukaisun miehistön koko oli 4 henkilöä.

::MIEHISTÖ::

Miehistön jäsen 1

Commander Kjell Lindgren

Taho: NASA

Wikipedia-artikkeli: [https://en.wikipedia.org/wiki/Kjell\\_N.\\_Lindgren](https://en.wikipedia.org/wiki/Kjell_N._Lindgren)

Miehistön jäsen 2

Pilot Robert Hines

Taho: NASA

Wikipedia-artikkeli: [https://en.wikipedia.org/wiki/Robert\\_Hines\\_\(astronaut\)](https://en.wikipedia.org/wiki/Robert_Hines_(astronaut))

Miehistön jäsen 3

Mission Specialist 1 Samantha Cristoforetti

Taho: ESA

Wikipedia-artikkeli: [https://en.wikipedia.org/wiki/Samantha\\_Cristoforetti](https://en.wikipedia.org/wiki/Samantha_Cristoforetti)

Miehistön jäsen 4

Mission Specialist 2 Jessica Watkins

Taho: NASA

Wikipedia-artikkeli: [https://en.wikipedia.org/wiki/Jessica\\_Watkins](https://en.wikipedia.org/wiki/Jessica_Watkins)

Valitse tarkasteltava laukaisu:

- 1) 2022-04-08T15:17:00.000Z - Ax-1
- 2) 2022-04-27T07:52:00.000Z - Crew-4
- 3) 2022-10-05T16:00:00.000Z - Crew-5
- 0) Lopeta

Valintasi: 0

Lopetetaan.

Kiitos ohjelman käytöstä.

## L14T4: Vuokra-asuntojen vuokrien analyysi (lisätehtävä) (P)

*Tehtävän taso: Perustaso*

Tee Python-ohjelma, joka lukee tiedostosta vuokra-asuntojen keskineliövuokrat postinumeroalueittain. Laske jokaiselta tiedostossa olevalta vuodelta neliövuokrien keskihinta koko maassa sekä kaikkien vuokra-asuntojen yhteismäärä kyseisenä vuotena. Pyöristä keskineliövuokrat

kahden desimaalin tarkkuuteen. Laske vuoden keskineliövuokra laskemalla vuokrat yhteen ja jakamalla se dataa sisältävien postinumeroalueiden määrällä. Kirjoita tiedot tiedostoon esimerkkiajon mukaisesti. Tee kurssin tyyliohjeen mukainen normaali valikkopohjainen ohjelma virheenkäsittelyllä, luokalla ja oliolistoilla sekä kirjastolla.

Huomaa, että joissakin data riveissä ei välttämättä ole vuokra- tai asuntotietoja ja sarakkeessa on tällöin piste ("."). Jos asuntojen lukumäärässä tai vuokrassa on pelkkä piste, hylkää rivi äläkä laske sitä mukaan keskivuokraan.

Datan lähde: [Tilastokeskus](#)

### Esimerkki syötetiedostosta 'L14T4D1.csv':

```
"Vuosineljännes";"Postinumero";"Huoneluuku";"Lukumäärä";"Neliövuokra (eur/m2)"
"2015Q1";"00100 Helsinki Keskusta - Etu-Töölö (Helsinki)";"Yksiöt";452;24.26
"2015Q1";"00100 Helsinki Keskusta - Etu-Töölö (Helsinki)";"Kaksiot";243;18.36
"2015Q1";"00100 Helsinki Keskusta - Etu-Töölö (Helsinki)";"Kolmiot+";34;13.09
"2015Q1";"00120 Punavuori (Helsinki)";"Yksiöt";95;21.52
"2015Q1";"00120 Punavuori (Helsinki)";"Kaksiot";52;14.77
"2015Q1";"00120 Punavuori (Helsinki)";"Kolmiot+";.;.
"2015Q1";"00130 Kaartinkaupunki (Helsinki)";"Yksiöt";24;27.01
"2015Q1";"00130 Kaartinkaupunki (Helsinki)";"Kaksiot";.;.
"2015Q1";"00130 Kaartinkaupunki (Helsinki)";"Kolmiot+";.;.
"2015Q1";"00140 Kaivopuisto - Ullanlinna (Helsinki)";"Yksiöt";141;23.26
"2015Q1";"00140 Kaivopuisto - Ullanlinna (Helsinki)";"Kaksiot";56;15.16
"2015Q1";"00140 Kaivopuisto - Ullanlinna (Helsinki)";"Kolmiot+";.;.
"2015Q1";"00150 Eira - Hernesaari (Helsinki)";"Yksiöt";299;22.98
"2015Q1";"00150 Eira - Hernesaari (Helsinki)";"Kaksiot";135;17.84
"2015Q1";"00150 Eira - Hernesaari (Helsinki)";"Kolmiot+";45;22.40
```

### Esimerkkiajo 1

#### *Syötteet:*

```
1
L14T4D1.csv
2
3
L14T4T1.txt
0
```

#### *Tuloste:*

```
Valitse haluamasi toiminto:
1) Lue tiedosto
2) Analysoi tiedot
3) Kirjoita tiedosto
0) Lopeta
Valintasi: 1
Anna luettavan tiedoston nimi: L14T4D1.csv
Tiedosto 'L14T4D1.csv' luettu.

Valitse haluamasi toiminto:
```

```
1) Lue tiedosto
2) Analysoi tiedot
3) Kirjoita tiedosto
0) Lopeta
Valintasi: 2
Vuokrien analyysi suoritettu 1 vuodelle.

Valitse haluamasi toiminto:
1) Lue tiedosto
2) Analysoi tiedot
3) Kirjoita tiedosto
0) Lopeta
Valintasi: 3
Anna kirjoitettavan tiedoston nimi: L14T4T1.txt
Tiedosto 'L14T4T1.txt' kirjoitettu.

Valitse haluamasi toiminto:
1) Lue tiedosto
2) Analysoi tiedot
3) Kirjoita tiedosto
0) Lopeta
Valintasi: 0
Lopetetaan.

Kiitos ohjelman käytöstä.
```

**Esimerkki tulostiedostosta 'L14T4T1.txt':**

```
Vuosi;Asuntojen lukumäärä;Neliövuokra (eur/m2)
2015;1576;20.06
```