

Harjoitustyön vaiheen 1 palautus

Johdanto

En ole aiemmin paljoa koodaillut tai tuottanut ohjelmia. Pysin luomaan avointa dataa rajapintojen avulla hyödyntävän koostepalvelun, koska tällaiseen tarkoitukseen löytyy paljon valmiita rajapintoja. Netistä myös löytyy Googlaamalla vinkkejä rajapintojen käyttöön Django ja Visual Studiolla.

Käyttämäni teknologia ja ympäristö

Kehitysympäristö: Django

Ohjelmointikieli: Python 3.8.1

Editori: MS Visual Studio Code

Tietokanta: SQLite

Käyttöjärjestelmä: Windows 10

Valitsin Django, koska sitä suositeltiin aloittelijoille. Ohjelmointikielenä käytin Pythonia, koska olen sitä käyttänyt aiemmin toisella kurssilla ja sitä käytetään myös tämän kurssin koodiklinikoilla. Jälkimmäisestä syystä valitsin myös Visual Studion editoriksi. SQLite on oletuksena Django:ssa tietokannaksi, enkä näe syytä vaihtaa sitä.

Pythonin ja Django:n asennus sekä kehitysympäristön perustaminen

Aluksi latsin Pythonin koodiklinikan ohjeiden mukaisesti. Tämä onnistui, koska tietokoneeltani löytyi asennuksen jälkeen Pythonin uusin versio ja Powershell tunnisti kielen ja version kuvan 1 mukaisesti.

```
PS C:\WINDOWS\system32> python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Kuva 1: Pythonin tunnistaminen Powershellissä

Seuraavaksi loin Powershellillä harjoitustyölle virtuaaliympäristön haluamaani tiedostosijaintiin. Aluksi sallittiin tietokoneelle scriptien suorittaminen antamalla kuvan 2 mukainen komento ja antamalla lupa ("Y") Powershellin varmistaessa olenko varma.

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy 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
PS C:\WINDOWS\system32>
```

Kuva 2: Scriptien salliminen

Pythonin toimivuus testattiin jo alussa, mutta testataan vielä pipin toimivuus *pip freeze* -komennolla.

```
PS C:\Users\Riku Sundqvist> python -V
Python 3.8.1
PS C:\Users\Riku Sundqvist> pip freeze
appdirs==1.4.3
distlib==0.3.0
filelock==3.0.12
six==1.14.0
virtualenv==20.0.1
PS C:\Users\Riku Sundqvist>
```

Kuva 3: Pip toimii

Seuraavaksi asennettiin virtuaaliympäristö komennolla:

pip install virtual env

Seuraavaksi loin harkkatyötä varten kansion haluamaani tiedostosijaintiin. Valitsin koodiklinikan esimerkin mukaisesti kansiolle sijainnin C:\Users\Riku Sundqvist\documents. Powershellissä kansioden välillä navigoitiin komennoilla *cd [kansion nimi]* haluamaansa kansioon ja *cd ..* vei takaisin päin. Halutessaan saatavilla olevat kansiot ja tiedostot sai näkyviin komennolla *cd ls*, joka tulosti listan sen hetkisen sijainnin alavalikosta.

Tässä kohtaa merkittäviä vaikeuksia seurasi siitä, että nimelläni oleva kansio (Riku Sundqvist) käyttäjissä sisältää välilyönnin, jota ei käsitelty koodiklinikalla. Komento *cd Riku Sundqvist* tuotti siis virheviestin, koska *Riku* nimistä kansiota ei ole olemassa. Tätä pähkäilin hyvin kauan, kunnes Googlestä lopulta selvisi, että minun täytyy käyttää heittomerkkejä välilyönnin sisältävää kansiota

käsitellessä. Oikea komento oli siis `cd "Riku Sundqvist"`. Seuraavaksi loin "OhsihaHarkka" -nimisen kansion komennolla `mkdir OhsihaHarkka`.

```
PS C:\> cd Users
PS C:\Users> cd "Riku Sundqvist"
PS C:\Users\Riku Sundqvist> cd documents
PS C:\Users\Riku Sundqvist\documents> ls

Directory: C:\Users\Riku Sundqvist\documents

Mode                LastWriteTime         Length Name
----                -
d-----         10.2.2020    20.15              OhsihaHarkka
```

Kuva 4: Harkkatyökansioni sijainti ja sinne navigointi. Erityishuomio heittomerkkien käytölle välilyönnin sisältävää kansion nimeä käsiteltäessä.

Viimeinen komento virtuaaliympäristön luomiseksi oli `virtualenv env`, jossa `env` oli ympäristölle antamani nimi. Virtuaaliympäristö käynnistettiin komennolla

`.\Scripts\activate`

Onnistuneen käyttöönoton merkiksi Powershellin komentoriviin ilmestyi `\env` -pääte.

```
PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka> cd env
PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\env> .\Scripts\activate
(env) PS C:\Users\Riku Sundqvist\documents\OhsihaHarkka\env>
```

Kuva 5: Virtuaaliympäristön käynnistys

Seuraavaksi asennettiin Django komennolla `pip install django`. Lisäksi Visual Studioon käyttäjänä asensin PyLintin, joka on Python -koodauksessa apuväline virheiden tarkastamiseen. Asennus tapahtui komennolla `install pylint`.

Seuraavaksi loin Harkkatyö -nimisen Djangoprojektin komennolla `django-admin startproject Harkkatyö`. Ennen tätä menin kansiossa yhden askeleen taaksepäin komennolla `cd ..`.

Seuraavaksi asensin Python lisäosan Visual Studioon. Tämän jälkeen avasin projektin Visual Studioon komennolla

`code`. (Huom! välilyönti ennen pistettä)

Ennen testausta valittiin vielä oikea tulkki:

`ctrl+shift+p -> Python select interpreter`

Testatakseni ympäristön toimivuuden navigoin takaisin projektiini ja ajoin Powershellissä komennon *py manage.py runserver*. Seuraavaksi syötettiin saatu URI (korostettuna kuvassa 6) selaimeen ja saatiin kuvassa 7 näkyvä ilmoitus ympäristön virheettömästä toiminnasta.

```
(env) PS C:\Users\Riku Sundqvist\documents\OhsiaHarkka\env> cd ..
(env) PS C:\Users\Riku Sundqvist\documents\OhsiaHarkka> cd Harkkatyö
(env) PS C:\Users\Riku Sundqvist\documents\OhsiaHarkka\Harkkatyö> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 20, 2020 - 20:54:11
Django version 3.0.3, using settings 'Harkkatyö.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

6: Komennot ympäristön testaamiseksi



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

7: Selaimeen syötetyllä URilla näkyvä viesti toimivasta virtuaaliympäristöstä

Seuraavaksi luotiin django applikaatio komennolla *py manage.py startapp applikaatio* ja tehtiin koodiklinikan mukaiset muutokset tarvittaviin kansioihin.

Tietokannan käyttöönotto

Otettiin SQLite oletustietokannaksi käyttöön komennolla *python manage.py migrate* kuvan 8 mukaisesti.

```
(env) PS C:\Users\Riku Sundqvist\documents\ohsiharkka\harkkatyö> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
```

Kuva 8: Asetetaan oletustietokannaksi SQLite

Seuraavaksi tehtiin tarvittavat muutokset applikaation *models.py* tiedostoon, joka käsittelee tietokantoihin liittyvät seikat:

```

applikaatio > models.py > ...
1  from django.db import models
2
3  class questions(models.Model):
4      question_text = models.CharField(max_length=200)
5      pub_date = models.DateTimeField('date published')
6
7  class Choice(models.Model):
8      question = models.ForeignKey(questions, on_delete=models.CASCADE)
9      choice_text = models.CharField(max_length=200)
10     votes = models.IntegerField(default=0)
11

```

Kuva 9: Tarvittavat muutokset models.py tiedostoon

Muutokset otettiin käyttöön Powershellillä komennolla *py manage.py makemigrations applikaatio*. Viimeisenä vielä testattiin toimivuus komennolla *python manage.py shell*. Ainoa virhe oli kuvan 10 mukainen `NameError`, jota en saanut ratkaistua.

```
NameError: name 'Question' is not defined
```

Kuva 10: virhe tietokantaa testatessa

Helppoa:

- Ohjeistus koodiklinikalla selkeää, mikä helpotti mukana pysymistä ympäristöä luodessa
- Powershellin komennot jäivät nopeasti mieleen
- Visual Studion käyttö helppoa selitteineen ja automaattisine täyttöehdotuksineen komentoja kirjoittaessa.

Vaikeaa:

- Pitää olla huolellinen: jokainen kirjoitusvirhe on kriittinen lopputuloksen kannalta Powershelliä ja Visual Studioa käyttäessä.
- Pitää tarkastaa, että tekee oikeat komennot oikeassa kansiossa.
- Jos putoaa kärryiltä koodiklinikkaa seuratessa joutuu kelaamaan paljon taakse, että pääsee taas kiinni