Eesti keele töötlus Pythonis

Praktikum 1

Käesoleva praktikumi eesmärgiks on üles seada töökeskkond, mida läheb edaspidi tarvis praktikumiülesannete lahendamisel, ning ühtlasi tutvuda kasutatavate töövahenditega (Python, Anaconda, JupyterLab / Notebook). Püstitatud ülesannete lahendamine aitab mõista töövahendite kasulikkust ja tunda end nende kasutamisel mugavalt.

Pythoni virtuaalkeskkonnad ja Anaconda installimine

Kujutlegem järgmist stsenaariumi: leiate Internetist (aga miks mitte ka nt „sahtlipõhjast", oma vanade ja väärtuslike skriptide seast) ühe Pythoni skripti ja proovite seda jooksutada. Ei tööta. Selgub, et skript annab süntaksivigu (kuna nõuab Pythoni versiooni, kus süntaks oli „natukene teistsugune") ning lisaks nõuab selliseid lisateeke, mida teie Pythonisse installitud pole. Ok, probleemi lahendab natuke aega pusimist ja klikkimist: installite endale vajamineva Pythoni versiooni, lisate sellele vajaminevad teegid ja skript töötab. Siis aga leiate järgmise Pythoni skripti, mida saaksite mugavalt jooksutada oma vastinstallitud Pythoni versioonil, ainult on üks probleem – skript nõuab teegi X versiooni 1.1.5, samas kui eelmine skript nõudis teegi X versiooni 1.2.3. Kuna üks teek tohib korraga olla vaid ühes versioonis, peaksite skripti kasutamiseks installima tagasi versiooni 1.1.5 ja edaspidi hakkamagi niiviisi kord üht ja kord teist versiooni ümber installima. Parem lahendus on ehk installida vajaminev Pythoni versioon ka teist korda ja seekord panna peale siis teegi X versioon 1.1.5. Mida rohkem te Pythoni skriptide ja teekide arendamisega tegelete, seda rohkem sellelaadseid situatsioone tõenäoliselt tekib, ning küllaltki kiiresti võib muutuda raskeks „järge pidada", milline installitud Python millist „teekide konfiguratsiooni" parajasti sisaldab.

Probleemile pakuvad lahendust Pythoni virtuaalkeskkonnad, mis võimaldavad iga arenduse või katsetuse jaoks luua kataloogi, kuhu on koondatud kõik programmi jooksutamiseks vajalik: spetsiifiline Pythoni versioon + spetsiifilistes versioonides lisateegid. Virtuaalkeskondi on suhteliselt lihtne vahetada (piisab ühest käsureakäsust või valikust/klikist Notebookis) ning hallata – ühe käsureakäsuga võib endale installida keskonna koos uue Pythoni versiooniga. Virtuaalkeskkondi saab kasutaja hallata ka oma kodukataloogis – st ei teki probleemi, kus kasutajal pole võimalik uusi Pythoni teeke installida, kuna tal pole kirjutamisõigusi lähte-Pythoni kataloogis.

Käesolevas aines soovitame virtuaalkeskkondade haldamiseks vabavaralist Continuumi Anacondat, mis tuleb vaikimisi olulisemate teaduslike teekidega ning kus on palju teeke ette kompileeritud.

Ülesanne 1. Anaconda/Miniconda installimine (0.75 p)

Installige Anaconda või Miniconda ning looge endale selles virtuaalkeskkond. Ülesande lahendusena: käivitage käsureal käsk `conda env list` ning kopeerige[1] käsu väljund praktikumi lahendusena esitatavas märkmikufailis (Praktikum_1.ipynb) Ülesannete osa vastavasse lahtrisse.

Conda **installimine**, kaks varianti:

1) *Anaconda täisversioon* – sisaldab 1000+ vabavaralist lisapaketti ning võtab kettal ruumi ca 3GB. Oma operatsioonisüsteemile vastava Anaconda versiooni saate tõmmata aadressilt https://www.anaconda.com/download/ (valige *„Skip registration”*, kui soovite installida ilma registreerumata) (hoiatus: allalaetavad failid on ~400 MB – 1000 MB!!).

Inglisekeelsed paigaldusjuhised:

Windows: https://docs.continuum.io/anaconda/install/windows.html

Linux: https://docs.continuum.io/anaconda/install/linux

macOS: https://docs.continuum.io/anaconda/install/mac-os

2) *Miniconda* – sisaldab vaid Anaconda tööks esmavajalikke pakette ning võtab kettal ca 300-400 MB. Muud paketid saab sinna hiljem vastavalt vajadusele lisada. Minconda koos paigaldusjuhistega leiate aadressilt http://conda.pydata.org/miniconda.html;

(!) *Oluline:* Anaconda / Miniconda tuleks installida ainult **lokaalsele kasutajale** (mitte admin kasutajale!) ning paigutada kataloogi, kus kasutajal on õigus faile lisada / muuta / kustutada. Sellisel viisil ei nõua installimine administraatori õiguseid ning pärast on lihtne ka pakette lisada, uuendada, eemaldada.

Täiendavat abi leiad Anaconda/Miniconda installimise ingl k lühijuhendist:

https://conda.io/projects/conda/en/latest/user-guide/install/index.html

Käsurealiides. Kuigi Anaconda pakub ka graafilist kasutajaliidest (*Anaconda Navigator*), siis käesolevas praktikumis ning ka edaspidi kasutame siiski **`conda`** käsurealiidest, kuna 1) see on olemas nii Anacondas kui Minicondas, 2) seda on mugav kasutada nt serverisse terminaliga sisse logides ning 3) aktiivse arendustegevusega jätkates puutute tulevikus tõenäoliselt kõige rohkem kokku just sellisel viisil virtuaalkeskkondade kasutamisega.

Windows: pärast installimist peaks tekkima programm „*Anaconda Prompt*”, mille leiate *Start* menüüst (nt *Anaconda %&! => Anaconda Prompt*) ning mida peaks olema võimalik ka leida „*Search Windows*” otsinguakna kaudu. Leidke ja käivitage see programm.

Linux-CentOS: terminaliakna avamiseks: *Applications => System Tools => Terminal*

Linux-Ubuntu: avage Ubuntu ikoonile klikkides otsinguaken ning sisestage *„terminal”;*

MacOS: avage *Spotlight Search* ning sisestage *„terminal”*[2]

1 Juhuks, kui tarvis: juhiseid selle kohta, kuidas kopeerida Windows’i aknast teksti;

2 Detailsemalt kirjeldatakse Anaconda paigaldusejärgset kontrollimist siin: https://docs.continuum.io/anaconda/install/verify-install

Baastarkused. Leheküljelt https://conda.io/docs/user-guide/getting-started.html leiate ca 20 min pikkuse Anaconda baastarkuste kursuse, mille võite iseseisvalt programmi põhjalikumaks tundmaõppimiseks läbi teha. Allpool on toodud valik olulisemaid **`conda`** käske, vajadusel leiate eelviidatud materjalist täiendavaid selgitusi.

Kuvab loendi Pythoni versioonidest, mida on võimalik installida:

```
conda search --full-name python
```

Uue keskkonna loomine:

```
conda create --name keskkonna_nimi [pythoni_versioon] [lisapaketid]
```

Konkreetne näide:

```
conda create --name  minu_keskkond  python=3.10
```

(Loob keskkonna nimega `minu_keskkond`, kuhu on installitud Python 3.10)

Loodud keskkonna aktiveerimine:

```
conda activate keskkonna_nimi
```

Kui keskkond on aktiveeritud, siis ilmub käsuviiba ette sulgudes keskkonna nimi, näiteks:

```
(minu_keskkond) C:\Users\MyUserName>
```

Oluline! Pärast käsureaakna sulgemist suletakse ka keskkond. Seega, kui avate terminaliakna uuesti, tuleb ka keskkond uuesti aktiveerida.

Jooksva keskkonna deaktiveerimine:

```
conda deactivate
```

Märkus: vanemate kui 4.6 `conda` versioonide puhul on aktiveerimise / deaktiveerimise käsud veidi teistsugused:

- aktiveerimine (versioon < 4.6):
  - Windows: `activate keskkonna_nimi`
  - Linux, Mac: `source activate keskkonna_nimi`
- deaktiveerimine (versioon < 4.6):
  - Windows: `deactivate keskkonna_nimi`
  - Linux, Mac: `source deactivate keskkonna_nimi`

Loodud keskkondade loetelu kuvab käsk:

```
conda env list
```

Uute teekide ***installimiseks*** on vaja netiühendust: installimisel tõmmatakse teek koos selle sõltuvustega repositooriumist alla, pakitakse lahti ning paigaldatakse aktiivsesse keskkonda (vaikekeskkond on *base*).

Kui teegi nimi on teada, siis esmalt tasub proovida selle installimist Anaconda repositooriumi kaudu:

```
conda install teegi_nimi[=versioon]              #  kui teek on Anaconda
                                                 #  repositooriumis
```

Kui teek puudub Anacondast, tasub seda otsida PyPA (*Python Packaging Authority*) repositooriumist. Installimiseks käsk `pip` :

```
pip install teegi_nimi[==versioon]  # kui teeki ei ole Anacondas,
                                    # aga on PyPAs
```

Aktiveeritud keskkonda installitud teekide (ja nende versioonide) loetelu annab käsk:

```
conda list
```

EstNLTK installimine

Alates järgmisest praktikumist kasutame igas praktikumis ka eesti keele automaattöötluse teeki EstNLTK, seega on hea see juba varakult ära installida.

Siin aines kasutame EstNLTK versiooni 1.7.3, mis on saadaval **`pip`** paketina 64-bitisele Linuxile, Windowsile ja macOS-ile. EstNLTK töötab Pythoni versioonidega 3.9 kuni 3.12 aga siin aines on soovitatav kasutada Python 3.10-t.

EstNLTK v1.7.3[3] installimine toimub käsurealt käsuga:

```
pip install estnltk==1.7.3
```

Lisaks tuleb täiendavalt alla laadida NLTK sõnestamise ressurssid, selleks on käsk:

```
python -c "import nltk; nltk.download('punkt_tab');"
```

EstNLTK installi testimiseks võite proovida järgnevat käsurida:

```
python -c "import estnltk; print(estnltk.Text('Jõudu tööle!').tag_layer().lemma)"
```

... mille oodatav väljund on:

```
[['jõud'], ['töö'], ['!']]
```

Kui tekib probleeme, küsige julgelt abi.

Jupyteri installimine

Kui olete EstNLTK-d sisaldavas virtuaalkeskkonnas, installige JupyterLab / Notebook käsuga:

```
conda install -c conda-forge jupyterlab
```

3 Vanemad EstNLTK versioonid, näiteks 1.4.1, ei ühildu käesolevate materjalidega. Seega palun jälgige, et installite õige versiooni.

Märkus: Kuigi Anaconda baaskeskkonnas on JupyterLab juba olemas, siis uut keskkonda luues peate selle ikkagi uuesti installima. Vastasel juhul hakkab käsk `jupyter lab` käivitama hoopis baaskeskkonnas oleva JupyterLab-i ning hakkate pakettide importimisel saama veateateid, kuna baaskeskkonnas puuduvad vajalikud paketid (nt *estnltk*).

JupyterLab on graafiline rakendus interaktiivseks programmeerimiseks. Täpsemalt on tegemist veebilehitsejas jooksva programmeerimiskeskkonnaga, kus on võimalik luua ja jooksutada Pythoni koodi sisaldavaid „märkmikke" (*notebook*), hallata programeerimisprojektile vajalikke (andme)faile ning kasutada Pythoni terminali ja op-süsteemi käsurida.

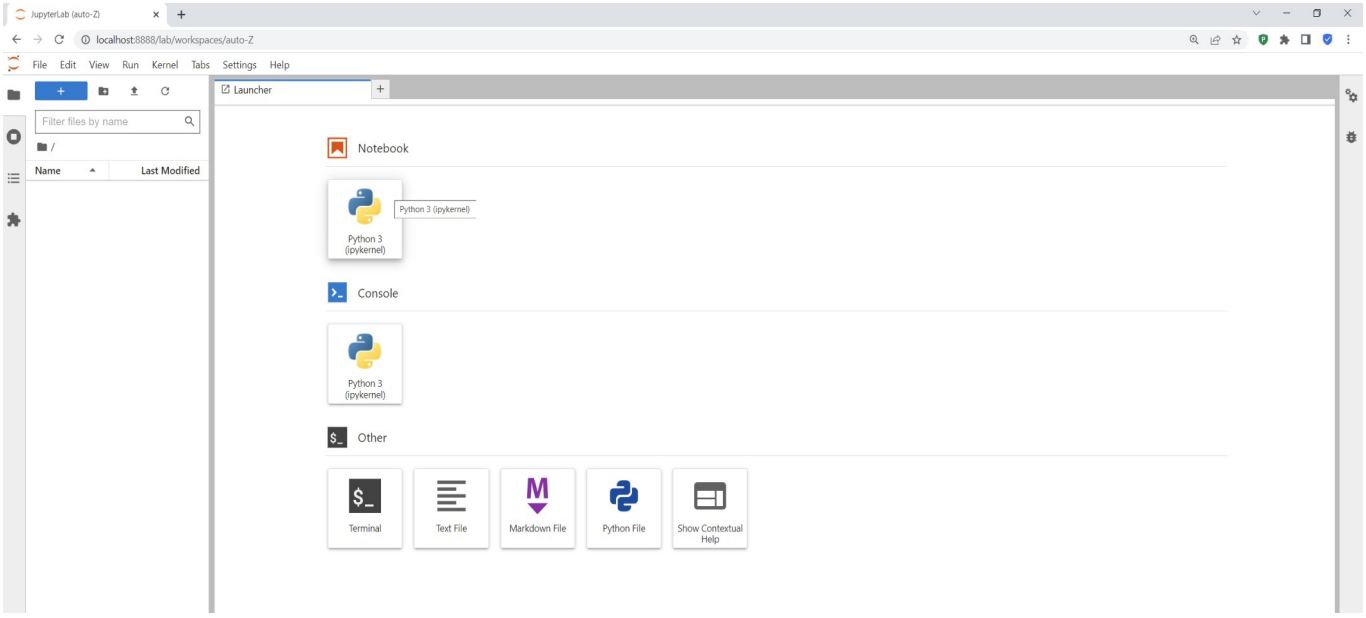
JupterLab-i interaktiivsed märkmikud võimaldavad:

1. testida erinevaid koodiridu ja –plokke ning talletada eelmiste plokkide muutujate väärtuseid (sh imporditud teegid, defineeritud funktsioonid jms);
2. lisada koodile dokumentatsiooni, sh kasutada erinevaid vorminduselemente (nt pealkirjad, tabelid, pildid ja graafikud);
3. eksportida loodud dokumente erinevatesse formaatidesse ning teistega jagada;

Lab-i käivitamiseks tuleks terminaliaknas liikuda[4] soovitud kausta (mis saab rakenduse „juureks", millest ülespoole failisüsteemis enam liikuda ei saa) ning sisestada käsureale:

```
jupyter lab
```

mille tulemusena avaneb veebilehitsejas vaheleht:



Pildil olevas näites on JupyterLab avatud kaustas, kus ei ole mitte ühtegi faili (vasakul olev failide vaade on tühi). *„Launcher"* aknas on võimalik luua uus märkmik (*"Notebook"* -> *„Python 3")*, avada Pythoni käsurida (*„Console"* -> *„Python 3"*) või op-süsteemi käsurida („Other" -> *„Terminal")*; samuti saab luua uusi tekstifaile, Pythoni koodifaile või Markdown-i faile.

4 Vajadusel vt Windows'i terminaliakna kasutamisjuhised (sh liikumise käsud) ning Linuxi/Mac-i terminali kasutamisjuhised.

Et keskkonnaga tutvumine ladusamalt läheks, oleme koostanud näidisfaili, kus saate nii olemasolevaid koodiridu jooksutada kui ise uusi kirjutada. Selleks käivitage `jupyter lab` kaustas, kuhu pakkisite lahti praktikumi materjalid. Avanenud veebirakenduses klikkige failil *Praktikum_1.ipynb*

Alternatiiv: märkmiku saab käivitada ka otse terminaliaknas. See käib nii:

```
jupyter lab Praktikum_1.ipynb
```

Praktikumi juhend jätkub failis Praktikum_1.ipynb!

Ametliku inglisekeelse JupyterLab-i dokumentatsiooni leiab aadressilt:
https://jupyterlab.readthedocs.io/en/stable/index.html

Töökeskkonna sulgemine: JupyterLab-i sulgemiseks valige *File* menüüst *Shut Down* või vajutage serveri käsureaaknas Ctrl+C.