

HVEE.04.004 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, Jupyter Notebook). Püstitatud ülesannete lahendamine aitab mõista töövahendite kasulikkust ja tunda end nende kasutamisel mugavalt.

NB! Praktikumiülesannete lahendamisel soovitame kasutada Pythoni versiooni 3.5, kuna koodinäited ja kasutatavad teegid ei pruugi vanemate Pythoni versioonidega ühilduda.

Pythoni virtuaalkeskkonnad ja Anaconda installimine

Kujutlegem järgmist stsenaariumi: leiate Internetist (aga miks mitte ka nt „sahtlipõhjust”, 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 pool tundi 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. Parema 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 „järke 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 + lisateegid. Virtuaalkeskondi on suhteliselt lihtne vahetada (piisab ühest käsureakäsi või valikust/klikist Notebookis) ning hallata – ühe käsureakäsi võib endale installida keskkonna koos uue Pythoni versiooniga. Virtuaalkeskondi saab kasutaja hallata ka oma kodukataloogis – st ei teki probleemi, kus kasutajal pole võimalik uusi Pythoni teekke installida, kuna tal pole kirjutamisõigusi lähte-Pythoni kataloogis.

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

Ülesanne 1. Anaconda installimine (1 p)

Installige Anaconda ning looge endale selles virtuaalkeskkond. Ülesande lahendusena: käivitage käsurreal käsk `conda env list` ning kopeerige¹ käsu väljund praktikumi lahendusena esitatava Notebooki (.ipynb) Ülesannete osa vastavasse Markdowni lahtrisse.

Conda **installimine**, kaks varianti:

1) *Anaconda täisversioon* – sisaldab 720 vabavaralist lisapaketti ning võtab kettal ruumi ca 3GB. Oma operatsioonisüsteemile vastava Anaconda versiooni saate tõmmata aadressilt <https://www.continuum.io/downloads> (hoiatus: failid on 400-500 MB!!) või ringlevalt mälupulgalt;

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. Miniconda koos paigaldusjuhistega leiate aadressilt <http://conda.pydata.org/miniconda.html> või ringlevalt mälupulgalt;

(!) *Oluline*: 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, seega saab seda ka teha nt kooliarvutis.

Täiendavalt võib olla abiks Anaconda/Miniconda installimise ingl k kiirjuhend:

<https://conda.io/docs/user-guide/install/index.html>

Käsurrealiides. Kuigi Anaconda pakub ka graafilist kasutajaliidest (*Anaconda Navigator*), siis käesolevas praktikumis ning ka edaspidi kasutame siiski **conda** käsurrealiidest, 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 (*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 *Launchpad* ning sealt edasi *Terminal* või *iTerm*²

1 Juhuks, kui tarvis: juhend 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 30 min pikkuse Anaconda baastarkuste kursuse, mille võite programmi põhjalikumaks tundmaõppimiseks iseseisvalt läbi teha. Allpool on toodud valik olulisemaid **conda** käske, vajadusel leiate viidatud materjalist täiendavaid selgitusi.

Kuvab loendi Pythoni versioonidest, mida on võimalik installida:

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

Uue keskkonna loomine:

```
conda create --name keskkonna_nimi [python=3.5.3] [lisapaketid]
```

Konkreetne näide:

```
conda create --name minu_keskkond python=3.5.3
```

(Loob keskkonna nimega `minu_keskkond`, kuhu on installitud Python 3.5.3)

NB! Käesoleval kursusel on soovitatav kasutada keskkondi Pythoni versioonidega 3.5.X , kuna uuemad versioonid (3.6.X) ei pruugi olla ühilduvad `estnltk`-ga (järgmise praktikumi teema).

Loodud keskkonna aktiveerimine:

Windows: `activate keskkonna_nimi`

Linux, Mac: `source activate keskkonna_nimi`

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

Loodud keskkonna deaktiveerimine:

Windows: `deactivate keskkonna_nimi`

Linux, Mac: `source deactivate keskkonna_nimi`

Loodud keskkondade loetelu:

```
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 *root*).

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

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

Vanemad teegid võivad Anacondast puududa, sellisel juhul tasub neid 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
```

Jupyteri installimine

Kui installisite eelnevalt Anaconda täispaketi, siis Jupyter Notebook peaks olema teil juba olemas. Miniconda puhul tuleks see installida, sisestades (virtuaalkeskkonnas sees olles) käsureale:

```
conda install jupyter
```

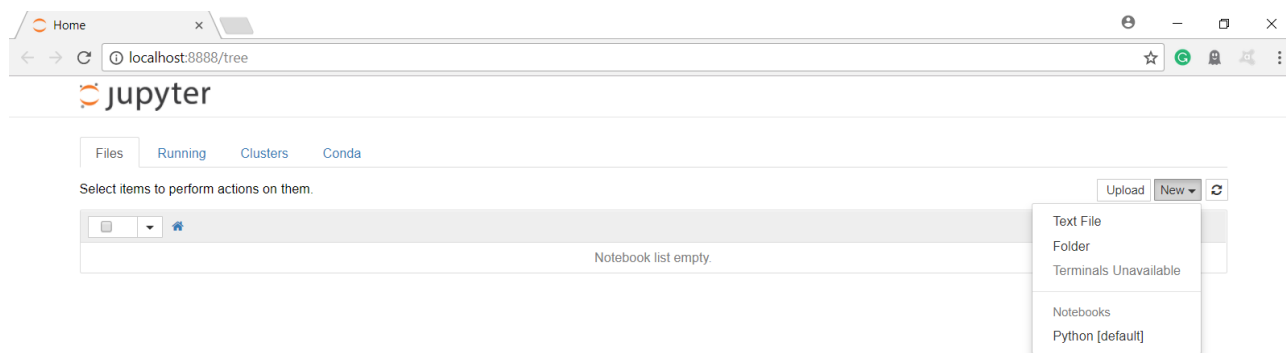
Jupyter Notebook on graafiline rakendus interaktiivseks programmeerimiseks. Täpsemalt on tegemist veebilehitsejas jooksva programmeerimiskeskkonnaga, milles on võimalik:

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;

Notebooki käivitamiseks tuleks terminaliaknas liikuda³ soovitud kausta (mis saab rakenduse „juureks”, millest ülespoole failisüsteemis enam liikuda ei saa) ning sisestada käsureale:

```
jupyter notebook
```

mille tulemusena avaneb veebilehitsejas uus vaheleht:



Pildil olevas näites on Jupyter Notebook avatud uues kaustas, kus pole ühtegi märkmikku. “New” nupu alt on võimalik luua uus tühi fail. Uue „märkmiku” loomiseks tuleks sealt valida *Notebooks* -> *Python*.

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

³ Vajadusel vt meeldetuletuseks: Windows’i terminaliakna [baaskäsud \(sh liikumise käsud\)](#) ning Linux/Mac-i [terminali baaskäsud](#).

Alternatiiv: Notebooki märkmikku saab käivitada ka otse terminaliaknas. See käib nii:

```
jupyter notebook Praktikum_1.ipynb
```

Praktikumi juhend jätkub failis `Praktikum_1.ipynb`!

Ametliku inglisekeelse märkmikku tutvustava juhendi leiate aadressilt:

<http://nbviewer.jupyter.org/github/ipython/ipython/blob/3.x/examples/Notebook/Notebook%20Basics.ipynb>