

1 Wat is Skillslab?

Het DLF Skillslab geeft leden de mogelijkheid om zich vaardigheden eigen te maken die nuttig zijn voor de studie, maar niet per se aan bod komen in de opleidingen. Zo kunnen leden in een informele(re) setting zich verdiepen in interessante onderwerpen, want je samen vaardigheden eigen maken is leuker en efficiënter. De precieze vorm zal per keer verschillen, maar het algemene idee is dat iedereen na een korte introductie aan de slag gaat met het onderwerp om zijn of haar eigen doelen te bereiken, maar vergeet vooral niet om bij elkaar te vragen en kijken!

2 Python

Python is een high-level programmeertaal. Dit betekent onder andere dat er heel veel *libraries* beschikbaar zijn om je te helpen om makkelijk dingen te bereiken die andere mensen al wel eens eerder wilden. Daarnaast is het helemaal gratis en open source. Dit betekent dat in tegenstelling tot bijvoorbeeld Matlab of Labview je nooit moeilijk zal hoeven doen met licenties, maar ook dat er misschien binnen de community meerdere stromingen zijn over hoe iets te doen. Voor meer informatie over wat Python is verwijzen we graag naar de desbetreffende pagina op de website van Python zelf

🔗 <https://wiki.python.org/moin/BeginnersGuide/Overview>

Als je wil spelen met de syntax kan dat het beste in de interactieve shell; typ `python3` om te beginnen! Er is ook een hele lijst tutorials voor als je nog nooit met Python gewerkt hebt te vinden:

🔗 <https://wiki.python.org/moin/BeginnersGuide/Programmers>

Enkele waar goede ervaringen mee bekend zijn zijn:

🔗 <http://www.diveintopython.net/toc/index.html>

🔗 http://stephensugden.com/crash_into_python/

2.1 Python 2 versus Python 3

Er zijn twee versies van Python in de omloop: versie 2 (`python`) en versie 3 (`python3`). Dit komt omdat voor versie 3 een aantal keuzes zijn gemaakt die niet compatible zijn met oude scripts (dus voor 2.7 en eerder).

Deze keuzes maken de taal in sommige gevallen een stuk logischer en handiger om mee te werken - werk dus vooral met Python 3 wanneer dat kan! Helaas zijn een aantal libraries nog steeds niet geport. Ook lijkt `numpy` niet te werken te krijgen met Python 3 op de ULCN-machines, omdat magie.

🔗 <https://python3wos.appspot.com/>

2.2 Lokaal packages installeren

Tot ieders grote verbazing kunnen we nog steeds geen software installeren op de UCLN-apparaten. Niet gevreesd, je kunt packages namelijk ook lokaal installeren met `virtualenv` en `pip`! Probeer het volgende:

```
mkdir mijnproject && cd mijnproject
virtualenv --prompt="(mijnproject)" env -p python3
. env/bin/activate
```

Hierna staat de tekst `(mijnproject)` voor je shell-prompt en is Python 3 je defaultversie. Gebruik `deactivate` om de environment weer uit te zetten. Je kunt nu packages installeren. Installeer bijvoorbeeld `sympy`, een library voor symboolmanipulatie:

```
pip install sympy
```

3 Mogelijke projecten

We willen je aanmoedigen om vanavond vooral iets te gaan proberen wat je zelf wil kunnen, maar daartoe moet je wel weten wat je wil kunnen. Hieronder staan wat suggesties van wat er zoal mogelijk is, maar laat vooral je fantasie de vrije loop!

3.1 Rekenen en plotten

Voor (wetenschappelijk) rekenen zijn SciPy en NumPy dé essentiële libraries. Deze verzorgen zowel matrixclasses als allerlei handige functies. Om data en functies te plotten wordt de matplotlib gebruikt. Voor mensen die Python slechts willen gebruiken om mee te rekenen en te plotten, raden we aan om naar iPython notebooks te kijken, hier kan je in een Maple/Sage achtige omgeving direct plotten en resultaten zien. De tutorial van het vak Diffussie is hier een goed startpunt.

3.2 Interfaces met C++ code, het OS

Omdat Python een geïnterpreteerde taal is en geen gecompileerde, kan het voor rekenintensieve projecten nuttig zijn om met externe gecompileerde code te communiceren. Dit kan door losse externe scriptjes aan te roepen en anderszins met het Operating system te communiceren, maar ook direct.

📄 <https://docs.python.org/3/library/subprocess.html>

📄 <https://docs.python.org/3/library/ctypes.html>

3.3 Webservers

Er wordt wel vaker gezegd dat je in Python in twee regels een webserver kan krijgen, en weinig meer om je computer op requests te laten reageren. Dit is in het bijzonder een goede combinatie met een Raspberry Pi en wat extra hardware (vraag maar aan Tobias de J.)

📄 <http://flask.pocoo.org/>

3.4 Scrapen

Misschien wil je wel een webcrawler schrijven. Het is erg makkelijk om HTTP-requests gedaan te krijgen en vervolgens door de HTML-achtige resultaten te bladeren. Kijk wel uit dat je niet per ongeluk iemand DDOS't.

📄 <http://docs.python-requests.org/en/latest/>

📄 <http://www.crummy.com/software/BeautifulSoup/>

3.5 L^AT_EX-output

Wat zou er nou mooier zijn dan je rekenresultaten meteen omgezet te zien worden in L^AT_EX? Met Python verre van moeilijk! Genereer TikZ-grafieken, ingewikkelde matrixvergelijkingen of plaatjes van katten, zodat je scriptie er geweldig uitziet.

📄 <https://github.com/JelteF/PyLaTeX>

3.6 MPI

Supercomputers zijn cool. Dus voor de mensen met een DAS4- of Cartesius-account: jullie kunnen aan de slag met MPI! De Python-bindings hiervoor heten mpi4py en zijn redelijk makkelijk te gebruiken. Reken eens een Mandelbrot uit!

📄 <http://mpi4py.scipy.org/docs/usrman/index.html>

3.7 Een spelletje

Een van de beste manieren om een taal te leren is een spelletje te schrijven. Probeer Snake te implementeren in de terminal, bijvoorbeeld met Blessings.

📄 <https://pypi.python.org/pypi/blessings>