

# Python telepítési segédlet

Sykora Henrik, Tóth Gergely\*  
BME-GPK Műszaki Mechanikai Tanszék

2020. február 18.

## 1. Bevezető

### 1.1. A Python nyelv

A Python egy nyílt forráskódú, általános célú, *interpretált, objektum orientált* programozási nyelv. 2020-ra a legelterjedtebben alkalmazott általános célú programozási nyelvvé nőtte ki magát, széles körben használják informatikai (pl: YouTube, Instagram, Spotify web-backend), tudományos és matematikai célokra (pl: Datamining, AI development, Machine Learning). Számos előnye van a mérnöki alkalmazásokban is, rengeteg ingyenesen elérhető függvénykönyvtár készült hozzá, néhány példa:

- `numpy`: numerikus számítások, lineáris algebra, numerikus integrálás, stb...
- `sympy`: szimbolikus számítások (computer algebra)
- `PIL`: képfeldolgozás (*Python Imaging Library*)
- `matplotlib`: függvény kirajzolás, eredmények megjelenítése
- `pandas`: data mining, adatfeldolgozás
- `sklearn`: machine learning
- ...

További előnye, hogy a Python nyelv törekszik a tömörségre, olvashatóságra és átláthatóságra, így az alapok órák alatt elsajátíthatóak (ellenben pl. a C, C++, Java nyelvekkel). A nyílt forráskódnak köszönhetően ingyenesen használható, minden ismertebb platformon elérhető (ellenben pl. a Matlabbal és Mathematicával), ezért a Python-ban írt kód mindenhol könnyen futtatható.

### 1.2. Fejlesztői eszközök

A Python nyelv fejlesztői által hivatalosan készített, úgynevezett 'core' (lásd még: interpreter, mag, fordító) funkciói erősen korlátozottak. A [Python.org](https://python.org) oldalról letölthető gyári csomag segítségével, ugyan futtathatjuk a programunk kódját egy console applikáción keresztül, de ebben a formában semmilyen pár sornál hosszabb kódot nem praktikus írni. Ami teljessé teszi a Python-nal történő munkánkat, azok a felhasználó közösség által fejlesztett IDE-k és Package managerek.

### 1.3. Package manager-ek

A Python nyelv fő előnye az alternatíváival szemben a moduláris jellegében rejlik. A különböző alkalmazásokhoz, különböző package-k (függvénykönyvtárak) állnak rendelkezésünkre, amiket valamilyen módon rendszerezniük kell. A jelen szöveg írásakor a legelterjedtebb package managerek a következők:

- `PIP`: A Python gyári package kezelő modulja. Sok funkcióval rendelkezik, de kezdők számára sok utánajárást igényelhet a használata (Command ablakból lehet a PIP-el kommunikálni).
- `Virtualenv`: Lehetővé teszi a virtuális környezetek könnyű kezelését, amire szükség lehet, ha ugyan azon packagek több fajta verzióját akarjuk felváltva használni. Elsősorban haladó fejlesztőknek ajánlott.

---

\*Észrevételeiket vagy a talált hibákat kérem, hogy a [sykora@mm.bme.hu](mailto:sykora@mm.bme.hu) e-mail címen jelezzék

- Anaconda: A mérnöki és data-science igényekre kifejlesztett manager az Anaconda. Grafikus felületével egyszerűen kezelhető, és gyárilag telepíti a legfontosabb package-eket. A segédlet további részében az Anaconda használata lesz részletezve.

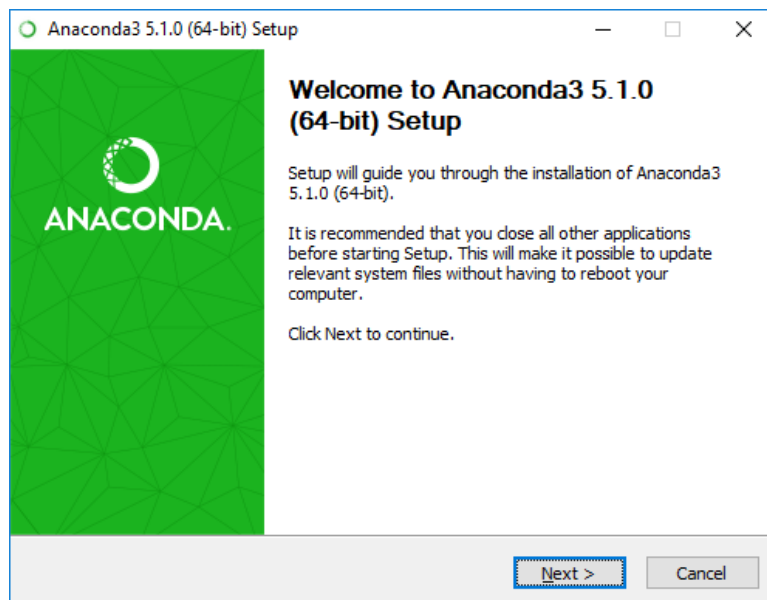
## 1.4. IDE-k

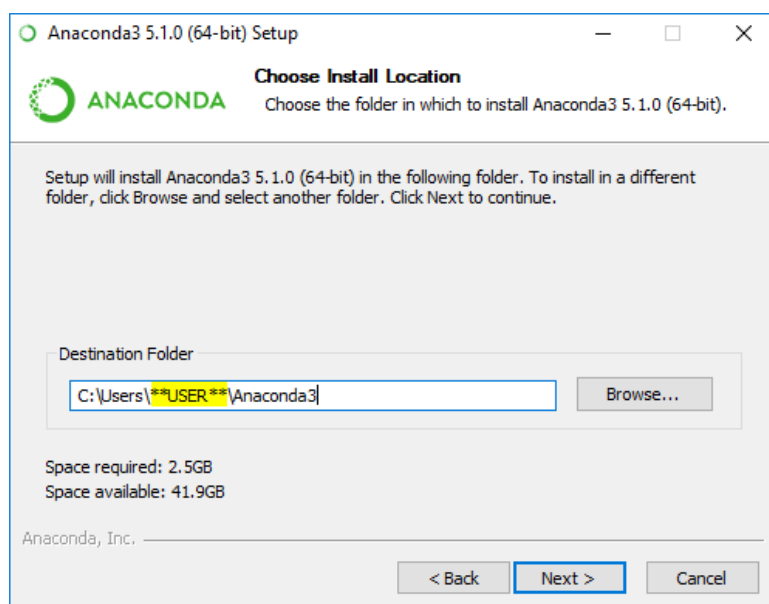
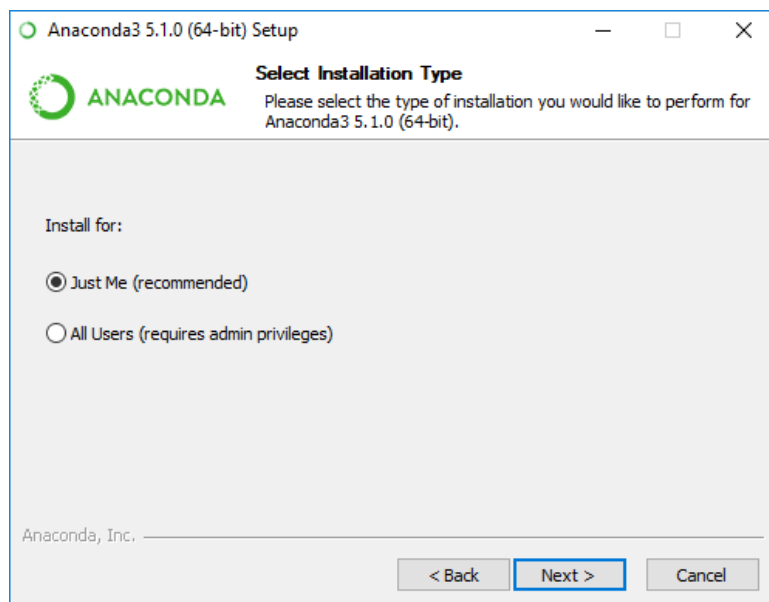
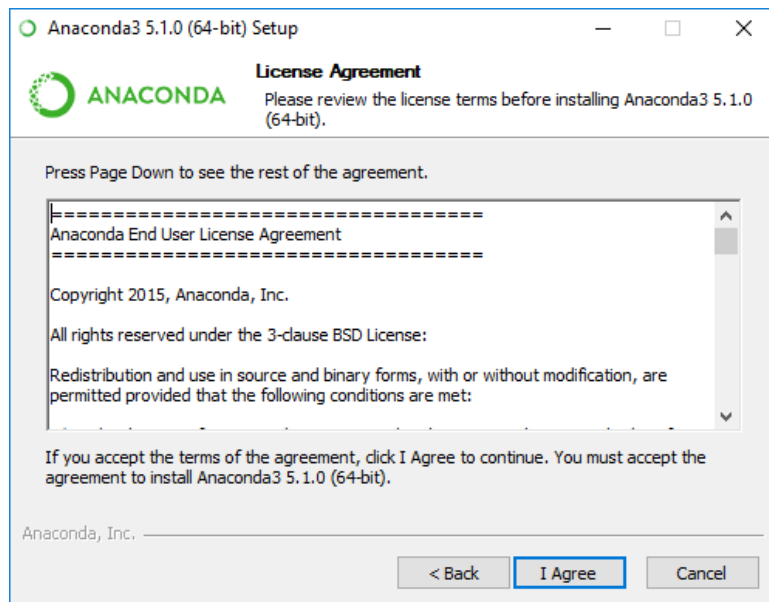
Az IDE (Integrated Development Environment) programok egy felületet teremtenek a felhasználó számára, amivel kommunikálhat a programkódot értelmező core-al. Számos segítő funkciót nyújtanak a kód debuggolásában, átláthatóbbá tételében, optimalizálásában. Egyes IDE-k lehetővé tesznek interaktív notebook és interaktív ábra készítést is. A jelen szöveg írásakor a legelterjedtebb IDE-k a következők:

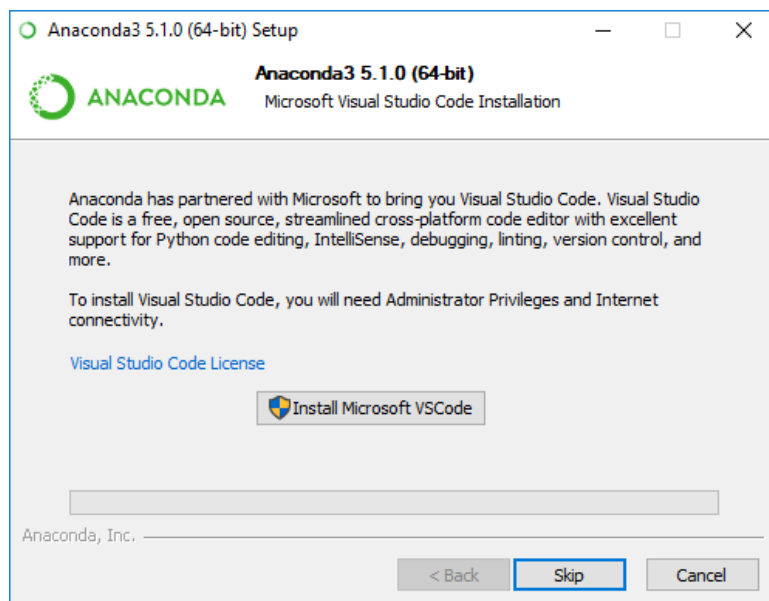
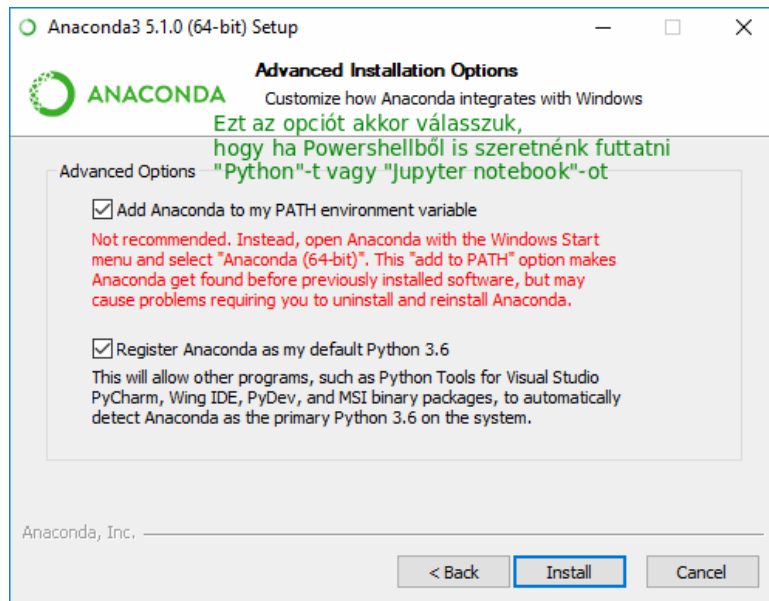
- IDLE: A Python gyári kód szerkesztő IDE-je. Minimalista mind a desing-t mind a funkcióit tekintve.
- Spyder: Tudományos és programfejlesztői munkára egyaránt alkalmas, ingyenes kódszerkesztő, debuggolást segítő funkciókkal.
- VS Code: A Microsoft Visual Studio mintáit követő, ingyenesen elérhető kódszerkesztő, kiterjedt fejlesztői eszközökkel. Nagyobb projekteknél értékelhető igazán a VS Code nyújtotta előnyök.
- Jupyter: A Python-t interaktív képességekkel felruházó IDE. A Jupyter Notebook-ban lehetőség van a programunkat több cellába rendezni, és a feladathoz kötődő formázott szöveget közvetlenül a kódunk cellái közé illeszteni, illetve interaktív ábrákat, diagrammokat készíteni. Ezeknek a funkcióknak köszönhetően elterjedten használják tudományos és műszaki feladatokban is.
- Jupyterlab: A Jupyter fejlesztőinek egy feltörekvő kezdeményezése, ami még több grafikus és interaktív lehetőséget biztosít majd a Python programozóknak. (Nem titkoltan a Matlab és a Wolfram Mathematica programokkal való verseny inspirálta)

## 2. Telepítés lépései

Töltsük le a következő helyről, a számunkra megfelelő verzióját az Anaconda csomagnak: [Anaconda 5.1 For Windows Installer - Python 3.6 - 64-Bit Graphical Installer \(537 MB\)](#). Indítsuk el a telepítőt:





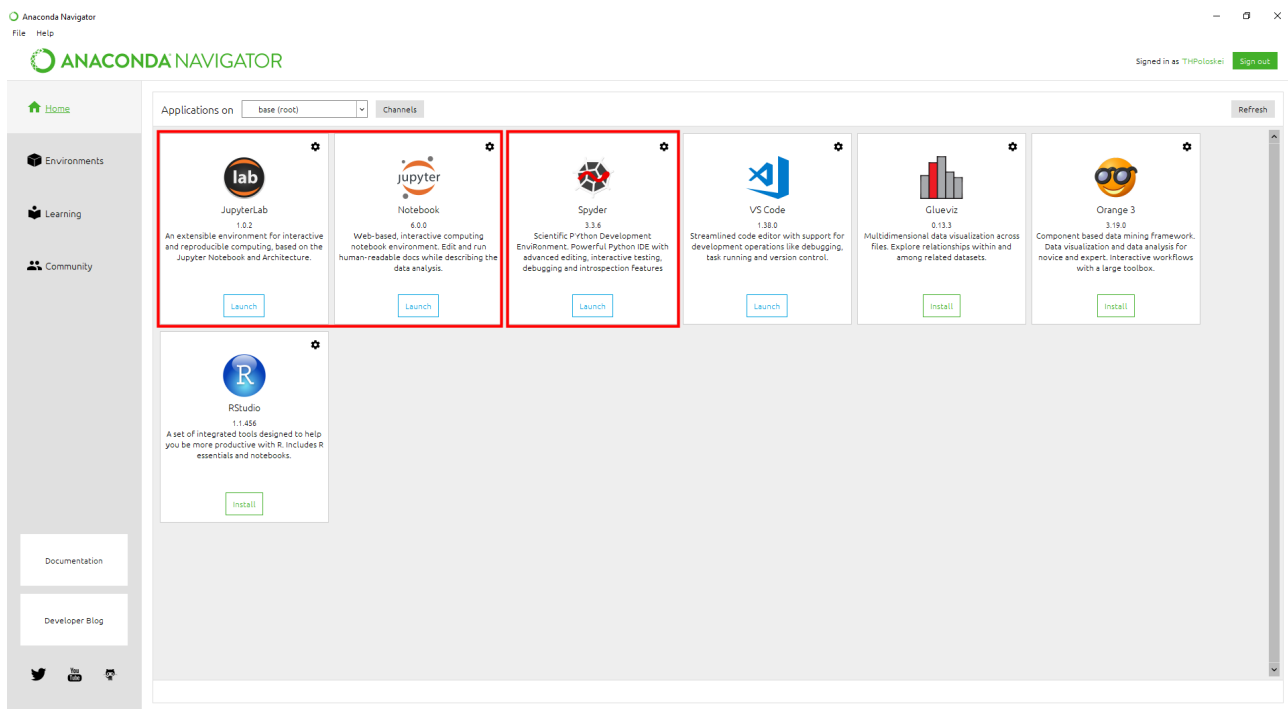


### 3. Python futtatása

Most hogy megvagyunk az Anaconda és a *Python* telepítésével, több módon is elindíthatjuk a *jupyter notebook* szerkesztőt. (A Navigator-ból elérhető több különböző IDE is, amit érdekességképp ki lehet próbálni.)

#### 3.1. Python futtatása Anaconda Navigator-on keresztül

A telepített programok közül futtassuk az Anaconda Navigator-t, majd kattintsunk a Jupyter Notebook alatti Launch ikonra.



#### 3.2. Python futtatása Jupyter notebook-ban parancsikonnal

A telepített programok közül közvetlenül is elérhető a *jupyter notebook*. A parancsikorra kattintva egy console ablak fogad minket, ami a futás során a Python környezetünk működéséről ad tájékoztatást. Ha a console ablak indulásával együtt, nem nyílik meg a *jupyter notebook* egy böngésző ablakban, akkor a console ablakban felkínált URL címet kézzel kell az álltalunk használt böngészőben megnyitni. (Ajánlott a Chomre használata, a bugok elkerülése végett)

Kijelölés Jupyter Notebook (Anaconda3)

```
[I 13:37:52.423 NotebookApp] The port 8888 is already in use, trying another port.
[I 13:37:52.423 NotebookApp] The port 8889 is already in use, trying another port.
[I 13:37:52.548 NotebookApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
[I 13:37:52.548 NotebookApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\jupyter\lab
[I 13:37:52.548 NotebookApp] Serving notebooks from local directory: C:\Users\
[I 13:37:52.548 NotebookApp] The Jupyter Notebook is running at:
[I 13:37:52.548 NotebookApp] http://localhost:8890/?token=a59ffff0f626ba6f73b6d2a70762d78f2a530d4962de76fd
[I 13:37:52.548 NotebookApp] or http://127.0.0.1:8890/?token=a59ffff0f626ba6f73b6d2a70762d78f2a530d4962de76fd
[I 13:37:52.548 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:37:52.611 NotebookApp]
```

To access the notebook, open this file in a browser:

file:///C:/Users/ /AppData/Roaming/jupyter/runtime/nbserver-2788-open.html

Or copy and paste one of these URLs:

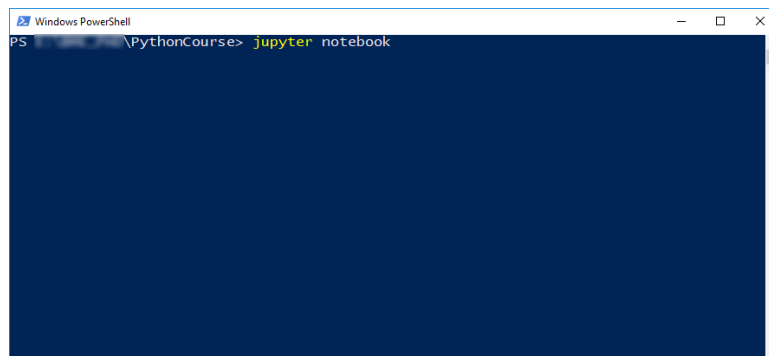
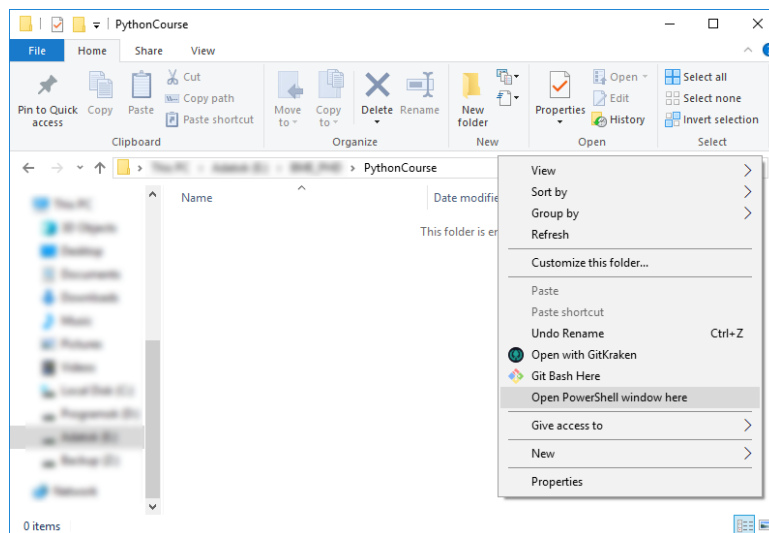
<http://localhost:8890/?token=a59ffff0f626ba6f73b6d2a70762d78f2a530d4962de76fd>

or <http://127.0.0.1:8890/?token=a59ffff0f626ba6f73b6d2a70762d78f2a530d4962de76fd>

A következő indítási módszerekhez szükséges a telepítés során az „Add Anaconda to my PATH environment variable” opció kiválasztása! Ha ez elmaradt, akkor a legegyszerűbben az Anaconda újratelepítésével orvosolható a probléma.

### 3.3. Python futtatása Jupyter notebook-ban terminálon keresztül

A *jupyter notebook* futtatható olyan formában is, hogy azt ne operációs rendszer gyökérmappájában nyissuk meg. Ehhez menjünk egy általunk választott mappába, ahol az ablakba történő *Shift+jobbklikk* segítségével indítunk egy Powershell-t. Ide gépeljük be a `jupyter notebook` parancsot. Ekkor elindul a böngészőben a notebook.



## 4. Parancsok és navigálás a Jupyter felületen

A Jupyter biztosít számunkra egy böngészőt, amivel böngészhetünk, létrehozhatunk, módosíthatunk és törölhetünk fileokat. Új Python Notebook-ot a *New->Python3* gombok megnyomásával hozhatunk létre. A notebookban a cellák *Shift+Enter* vagy *Ctrl+Enter* segítségével futtathatóak, és ha kéken van kijelölve a cella (*Command Mode* - *Esc* billentyű lenyomásával elérhető), akkor a *Shift+L* segítségével bekapcsolható a sorok számozása. Szerkesztő módban (*Editor Mode* - cellát kijelölve *Enter* lenyomása) cellákban akár egyszerre több kurzor is lehelyezhető a *Ctrl-balklikk* segítségével, vagy téglalapszerűen is kijelölhető a szöveg az *Alt* nyomvatartása mellett. A jupyter környezet help-je a "h" megnyomásával előhívható *Command mode*-ban.

A notebook átnevezhető a felső részben lévő "Untitled"-re kattintva. A futó notebook a jupyter menüjéből a *Shutdown* gomb segítségével állítható le, és a *jupyter* futása a *Powershell* bezárásával állítható le (vagy *Ctrl+c* a *Powershell* ablakban).

