

# Ismerkedés a Python® nyelvvel

Fábián Tamás László [giganetom@gmail.com](mailto:giganetom@gmail.com), HA5FTL

A Python® egy letisztult, nagy kifejezőerejű objektumorientált programozási nyelv. Aki még nem találkozott hasonlóan magas szintű nyelvvel, az talán most elbizonytalanodik az „objektumorientált” szó hallatán. Cikksorozatunknak egyik célja éppen az, hogy megmutassa az igazi erejét ennek a modern eszköznek.

A cikkek megírásakor törekedtünk arra, hogy semmilyen előzetes programozási tapasztalat ne legyen szükséges a megértéshez, egy alapvető számítástechnikai készség-repertoárra viszont mindenképp szükség lesz. Aki most ismerkedik a számítógépekkel, annak komoly nehézségei támadhatnak, míg aki most állna neki első programja készítésének, az bátran velünk tarthat.

A cikksorozat folyamán nem csupán elkezdjük az ismerkedést az egyik legmodernebb és legnépszerűbb programozási nyelvvel, de mindjárt olyan hasznos eszközöket is készítünk vele – a megfelelő számítógép és hardver birtokában – amelyek a gyakorlatban is a segítségünkre lehetnek. Fogunk készíteni így QRSS-hez, vagy beacon-höz használható billentyűzöt, hangfrekvenciás spektrum analízátort, szűrőket, I/Q modulátorokat és demodulátorokat – mindezeket egyenként néhány tucat programban.

## Miért pont a Python®?

Elegáns szintaxissal rendelkezik, így könnyű írni és olvasni. Ezen tulajdonsága miatt kiváló első programozási nyelvnek, és alkalmassá teszi arra, hogy gyorsan apró eszközöket készítsünk, vagy új ötleteket próbáljunk ki. A munkát még könnyebbé teszi az, hogy bőséges eszközkészlettel (standard library) rendelkezik: rengeteg függvény és objektum érhető el rögtön a telepítés után. Külön telepíthető modulok formájában hatalmas mennyiségű előre megírt programrészlet áll rendelkezésünkre. Nekünk ezek közül a legfontosabbak talán a tudományos számításokat lehetővé tevő math, numpy és scipy modulok, a grafikonok megjeleni-

tését egyszerűvé tevő matplotlib, a hangeszközöket kezelő sok megoldás, esetünkben a pyaudio. A Python® programok futtatásához szükséges értelmező (interpreter) teljesen nyílt forrású, ingyenes, szabad szoftver, és a legnépszerűbb platformokon működőképes (Linux, Mac OS, Windows).

## Telepítés

A Python® értelmezett (interpreted) nyelv, futtatásához értelmező (interpreter) program szükséges. Ezt a programot telepíteni kellhet, mert nem biztos, hogy operációs rendszerünk részét képezi. Windows és Mac felhasználók töltsék le a <http://python.org/download> címről az előre csomagolt verziót. Linux-ot használók a rendszer megszokott csomagkezelőjével telepítsék, Ubuntu 11.04 „Natty Narwhal” alatt így:

```
sudo apt-get install python
```

Linux rendszerek esetében legtöbbször a disztribúció részét képezi valamelyik Python® verzió. Mi itt a 2.7-est fogjuk használni, ezért tanácsos otthon is ezt installálni. Windows alá telepítsük a 32 bites verziót mindenképp akkor is, ha rendszerünk 64 bites, mert lesznek olyan csomagok, amelyekhez előre csomagolt változat csak így található meg.

Windows alatt a Python® telepítő nem frissíti a PATH változót, ezért a telepítési könyvtárat magunknak kell felvenni a keresési útvonalak listájába. Ezt Windows 2000/XP/Vista/7 alatt a következőképpen tehetjük meg: kattintsunk jobb egérgombbal a „Sajátgép” ikonjára, válasszuk a felugró menüből a „Tulaj-

donságok” menüpontot. A megjelenő dialógus dobozban kattintsunk a „Speciális” fülre. A fül alatti panelen kattintsunk a „Környezeti változók” gombra. A „Rendszerváltozók” listában keressük meg a „path” bejegyzést, jelöljük ki és kattintsunk a „Szerkesztés” gombra. Fontos, hogy ne töröljük ki a benne lévő szöveget, hanem írjuk a végére egy pontosvessző után közvetlenül az útvonalat, például így „;C:\Python27”.

Később szükség lesz még a numpy, scipy, matplotlib, pyaudio és pyserial modulokra, ezek Windows-ra és Mac-re letölthetőek a következő címekről:

```
http://www.scipy.org/
Download,
http://sourceforge.net/
projects/matplotlib/files/
matplotlib/matplotlib-1.0.1/,
http://people.csail.mit.edu/
hubert/pyaudio/,
http://pypi.python.org/pypi/
pyserial.
```

A linux disztribúciók többségében ezek is benne foglaltaknak, így kedvenc csomagkezelőnkkel telepíthetőek. Ubuntu Linux 11.04 alatt a következő parancsokat kiadva telepíthetjük ezeket (egy sorba írva):

```
sudo apt-get install
python-numpy python-scipy
python-matplotlib
python-pyaudio python-pyserial
```

Eleinte csak magát a Python® értelmezőt fogjuk használni egy parancssori ablakban, szövegszerkesztőre nem lesz szükség. Később viszont szükségünk lesz erre, így érdemes már most megismerkedni egy megfelelő eszközzel. Windows-

ra a notepad++ nevűt ajánljuk, Mac-re a TextMate-et, Linux alá pedig a geany-t.

Bízunk benne, hogy mindenki megbirkózik a telepítéssel, ezért nem közöljük részletesen a szükséges lépéseket. Aki mégis elakadna, annak a szerző szívesen nyújt segítséget a giganetom@gmail.com e-mail címen.

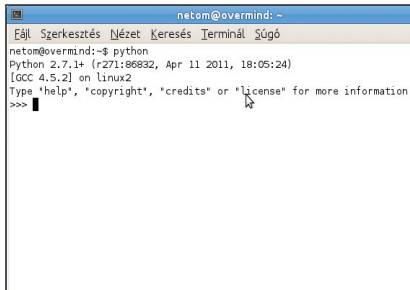
## Első lépések

Mint említettük, a Python® egy értelmezett nyelv. Aki foglalkozott már programozással, de eddig csak assembly-ben, C-ben, vagy esetleg Pascal-ban dolgozott, annak szokatlan lesz, hogy a programok nem lefordított, bináris – így ember számára olvashatatlan – fájlok, hanem egyszerű – és szépen olvasható – szöveges fájlok.

Python®-ban tehát a program maga a forráskód. A fordítás persze itt sem ússzuk meg, csupán az értelmező elrejt előlünk, a forrást ugyanis nem sorról-sorra értelmezi, hanem először lefordítja egy úgynevezett byte kódra, amit a számítási idővel való spórolás miatt bizonyos esetekben el is ment a lemezre és csak akkor fordít valamit újra, ha azt időközben frissítettük. Maga a byte kód azonban a processzor számára továbbra is értelmezhetetlen, a futtatást ezért nem ez, hanem egy virtuális gép – egy program – végzi. A Python® programok minden olyan operációs rendszeren futtathatók, amire készítették Python® fordítót és virtuális gépet.

A Python® hátránya, hogy a byte kód és a hardver között mindig ott van a virtuális gép, így a vas nyers erejét nem tudjuk kihasználni. Ha csupán kevés számítást végzünk, vagy a programunk interaktív, és sokat vár felhasználói beavatkozásra, akkor ez a hátrány nem vehető észre. Komolyabb adatfeldolgozási és számítási feladatok esetében viszont komoly akadály lehet. A cikksorozat további részeiben látni fogjuk, hogyan lehet ezt az egy hátrányt is sikeresen leküzdni.

A Python® értelmező két üzemmódban indítható. Az egyik akkor használatos, ha egy elkészült programot futtatni akarunk, ekkor a paraméterül kapott fájlt kérdés nélkül



```
netom@overmind:~$ python
Python 2.7.1+ (r271:96632, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on Linux2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

1. ábra

végrehajtja. Számunkra most az úgynevezett interaktív üzemmód az érdekesebb. Ha az értelmezőt úgy indítjuk el, hogy nem adunk meg neki futtatható programot, akkor egy promptot kapunk, ahol rövid programokat gépelhetünk be. Ezeket sajnos nem tudjuk elmenteni, így az interaktív módot kísérletezéshez célszerű csupán használni.

Indítsuk el a Python® értelmezőt interaktív módban! Linux és Mac felhasználók a „Python” parancs kiadásával tehetik ezt meg, Windows alatt keressük meg a telepítési mappát (ami C:\Python27, ha mást nem adtunk meg a telepítéskor). Ebben a mappában rögtön két Python® értelmező program is található: a python.exe és a pythonw.exe. Utóbbi akkor hasznos, ha grafikus felülettel rendelkező Python® alkalmazást szeretnénk futtatni, mivel ez a verzió nem nyit meg szöveges konzol ablakot. Ebből következik, hogy most a python.exe-re lesz szükségünk, hiszen mi pont a konzolt fogjuk használni. Kattintsunk kétszer a python.exe ikonjára, ezzel elindítva a Python® értelmezőt. Az 1. ábra szerinti képhez hasonlóan kell látnunk.

A három „nagyobb, mint” jel mutatja, hogy az értelmező bemenetre vár (lehet gépelni). A továbbiakban mi is így jelöljük, ha valamit az interaktív promptnál kell begépelni. Lássuk is az első példát!

```
>>> print "Helló, Világ!"
Helló, Világ!
>>>
```

A sor, amit begépelünk egy utasítás (statement). Az utasítás a legkisebb egység, amit a Python® értelmező képes végrehajtani. Amint látható, a „print” utasítás arra való,

hogy szöveget jelenítsünk meg vele, a szövegeket pedig idézőjelbe kell tenni. A print paranccsal rögtön meg is ismertünk egy nagyszerű eszközt: egy (egész számok esetében) tetszőleges pontosságú tudományos számológépet. Lássuk a következő példákat:

```
>>> print 1+2
3
>>> print 1*2
2
>>> print 2*2
4
>>> print 2**10
1024
>>>
```

A „print” utasítás után álló „dolgozat” (szövegeket, matematikai műveleteket, és sok más dolgot) kifejezésnek nevezzük. A kifejezés egyszerűen egy olyan dolgot jelöl, aminek van valamilyen értéke. A „print” utasítás ezt az értéket jeleníti meg.

Az alapl műveleteken kívül működik a hatványozás, ezt az legutolsó példa szerint dupla csillaggal jelöljük. A „tetszőleges pontosságú” azt jelenti, hogy – ha egész számokról van szó – bármekkora számokkal boldogul az értelmező:

```
>>> 12293**12234
```

Ennek az eredményét sajnos a cikkre érvényes területi korlátok miatt nem közölhetjük, a számításnak viszont egy másodpercen belül vége.

Kísérletezzünk, milyen kifejezéseket tud a Python® kiszámolni. Ha végeztünk a kísérletezéssel, az értelmezőből kilépni az „exit” függvény meghívásával lehet:

```
>>> exit()
netom@overmind:~$
```

A folytatásban sorra vesszük a nyelv alapelemeit: a használható adattípusokat, függvényeket. Később megismerkedünk az osztályok, objektumok használatával. Ezek után semmi akadály sem lesz annak, hogy a gyakorlatban is használható eszközöket készítsünk, elsőként a soros porton való kommunikációt fogjuk megismerni.

(Folytatjuk)