



Python és a turbó csiga

Dr.Guta Gábor

∇xonmatics
SCIENCE MEETS SOFTWARE
DEVELOPMENT

Bemutakozás



Dr.Guta Gábor
K+F vezető

- Kémiai-/bio-/egészségügyi-informatikai projekteken dolgozom Python, C és Java nyelven;
- rendszeresen foglalkozom tanácsadással és oktatással;
- a magyar piacon elsők között dolgoztam ki és tartottam piaci szereplőknek Python tréninget.

Bevezető



- *Hiedelem:* A Python egy lassú script nyelv, amiben nincsenek típusok
- *Valóság:* A Python egy fordított nyelv, ami egy C-ben írt VM-en fut; dinamikusan, erősen típusos
- *Kérdés:* **Lassú-e?**
- *Rövid válasz:* **Attól függ...**
- *Hosszú válasz:* **Lássuk mitől függ!**

Hogyan fut egy Python program?

- A Python forráskód lefordítódik egy köztes kódra
- A köztes kódot fogja futtatni a Python VM

Python forrás	Python VM forrás	Python VM bináris
<pre>def add(a, b): return a + b</pre>	<pre>0 LOAD_FAST 0 (a) 2 LOAD_FAST 1 (b) 4 BINARY_ADD 6 RETURN_VALUE</pre>	<pre>0x7c 0x00 0x7c 0x01 0x17 0x00 0x53 0x00</pre>





Mi van a memóriában?

Pythonban minden objektum

PyVarObject
Referencia számláló
Típus (mutató)
Méret
Adatok

Mi az a GIL?



- *Hiedelem:* A Python nem tud több magot kihasználni a GIL miatt.
- *Valóság:* A CPython több szálon való futtatáskor a GIL (Global Interpreter Lock) segítségével biztosítja, hogy csak egy szál módosítsa a VM állapotát
- A GIL-re váró szálak nagyon jól tudnak I/O-ra várakozni
- A C-ben írt csomagok futhatnak GIL-től függetlenül is
- Párhuzamosságot el lehet érni másképp is (processzek)
- Nem csak CPython létezik



Lassú-e a Python?



(gyorsít)

- Beépített típusok C-ben vannak megvalósítva
- Sokat vár I/O-ra egy átlagos program
- Nem direktben fut a szöveges forráskód



(lassít)

- CPython a köztes kódot C-ben írt VM-en futtatja
- Dinamikus típusosság (és minden objektum)
- Egy Python szál lehet aktív egy processzen belül

Gyorsítási lehetőségek



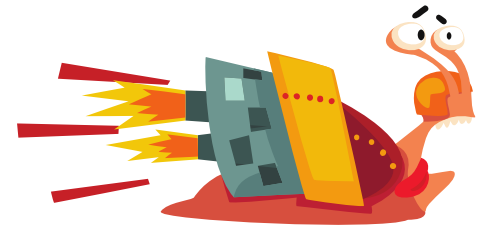
- Cseréljük le a CPython megvalósítást
 - PyPy
 - Nuitka
 - MypyC
- Kritikus részek gyorsabbra cserélése
 - C alapú csomagok
 - saját C csomagok
 - Cython (Python szintaxis C-hez :)
 - JIT-eljük a függvényeket Numba-val

PyPy



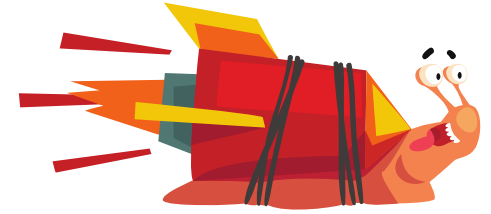
- 4x-es gyorsítást ígér a CPythonhoz viszonyítva
- A JIT-nek bemelegedési időre van szüksége, ezért kis programoknál nem biztos, hogy megéri a használata
- Általában kevesebb memóriát foglal
- A C részeket tartalmazó csomagokat újra kell fordítani a PyPy-jal való használathoz

Nuitka



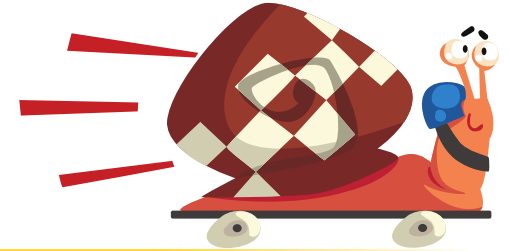
- Egy Pythonban írt Python fordító
- Teljes Python kompatibilitást ígér 3.7-es-ig
- A Python kódot C-re fordítja és a libpython könyvtárral linkeli
- Kompatibilitás: `co_code` és a debugger támogatás hiányzik

MypyC



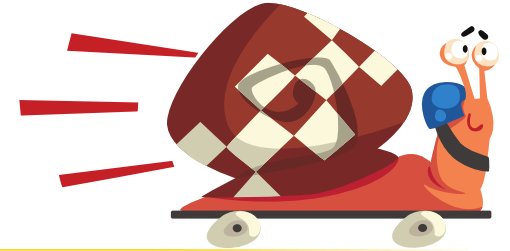
- C kódot állít elő Pythonból
- Kihaszználja a típus annotációkat
- Nincs még végfelhasználóknak való állapotban
- Csak a nyelv egy részhalmazát támogatja;
Hiányoznak:
 - számos speciális metódus támogatása
 - dinamikus objektum szerkesztési funkciók
 - öröklődés részlegesen van támogatva

C alapú modulok



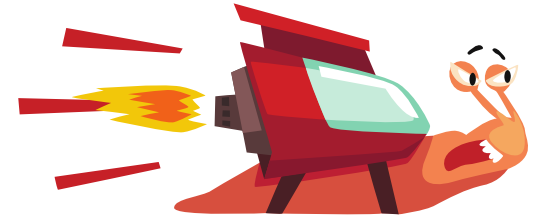
- Számos C-ben vagy C könyvtárakat használó csomag létezik.
- A legismertebb a NumPy
 - Saját ndarray adat típusa van, ami C tömböket használ
 - A műveletek hatékonyan C-ben vannak megvalósítva

C modulok



- A ,legfapadosabb' megoldás
- A CPython saját modulja is ebben készültek
- A leggyorsabb és legalacsonyabb szintű megoldás
- Viszonylag sok munkát és a CPython megvalósításának részleteit kell ismerni hozzá
- NumPy csomaghoz is hozzá lehet férni C-ből

Cython



- A Cython nyelv egy Pythonra nagyon hasonlító statikusan típusos nyelv
- Python kódokhoz lehetővé teszi szeparált típus definíciós fileok írását
- C kóddal azonos sebességet lehet elérni
- Könnyű CPythonnal integrálni
- `cdef` kulcsszóval definiálhatók a típusok

Numba



- Egy JIT fordító Pythonhoz, ami CPU-ra vagy GPU-ra fordít Python kódokat
- A programkódot egyszerűen @jit vagy @njit dekorátorokkal kell ellátni
- Támogatja függvények vektorizálását
- Támogat Nvidia CUDA és AMD ROC GPU-kat
- Ez utóbbi két esetben korlátozva van, hogy milyen műveletek és típusok engedélyezettek

Köszönöm a figyelmet!



gabor.guta@axonmatics.com