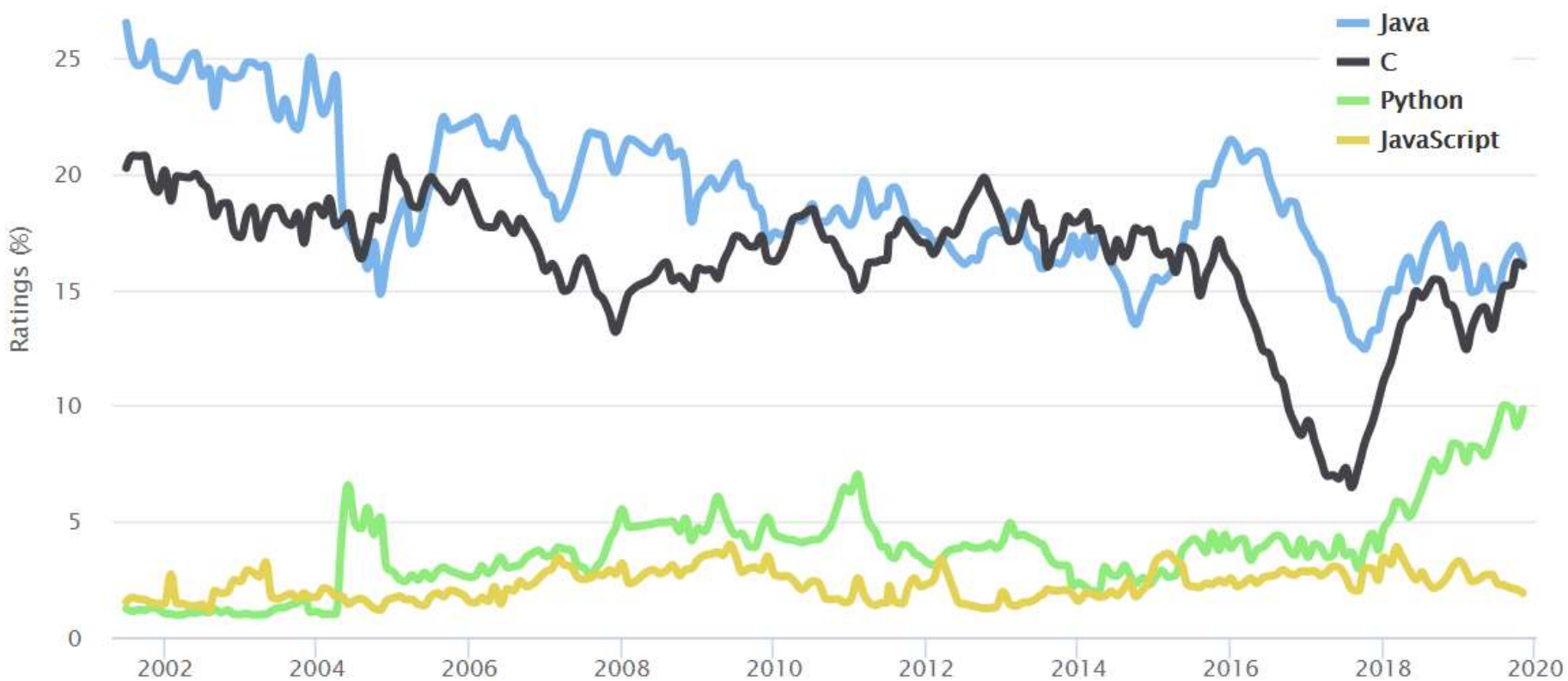


Ezért érdemes Pythonban fejleszteni

Dr.Guta Gábor

Mennyire népszerű?



Forrás: <https://www.tiobe.com/tiobe-index/>

Miért népszerű?

Az ABC nyelv elveinek a továbbgondolása:

- Egyszerű,
- Kompakt,
- Strukturált,
- Kifejező,
- Egységes,
- Magas-szintű,
- Egyértelmű,
- Interaktív

(Adaptálva: Leo Geurts, Lambert Meertens, and Steven Pemberton:

The ABC Programmer's Handbook)



Forrás: <https://freesvg.org/happy-python>

Miért népszerű? (folyt.)

Hatékonyság:

- Megvalósítandó feladat/ráfordított idő
- Átlagos commit méret (KLOC-ban)

Elérhetőség:

- Eszköz támogatás
- Oktató anyag/dokumentáció
- Képzett emberek



Forrás: <https://freesvg.org/happy-python>

Miért lehet utálni?

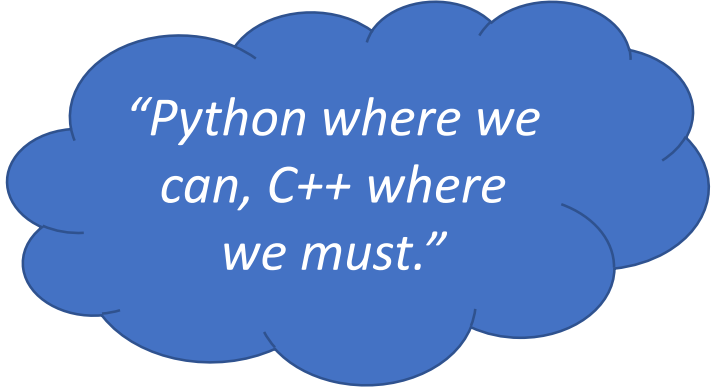
- Formázás (tabulálás)
- Nincs típusosság (de, van)
- Nem olyan mint a megszokott/kedvenc programozási nyelv
- Lassú (általában ez nem gond, de orvosolható)

(Környezetemben hangzottak el)



Kik használják a nagyvilágban?

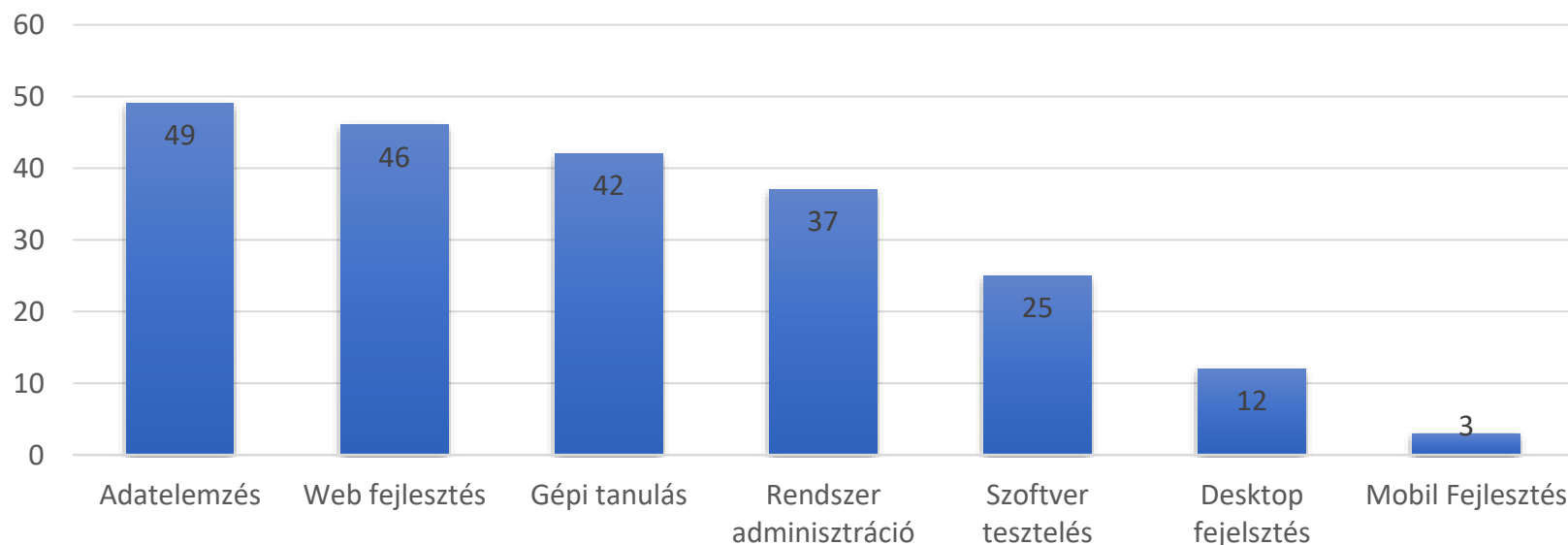
- Web oldalak: Instagram, YouTube, Printest, Prezi...
- Cégek: Dropbox, Google, Bloomberg, Spotify...
- Alkalmazások: Gedit, yum, Home Assistant, Cura...



“Python where we can, C++ where we must.”

Mire használják a nagyvilágban?

Fejlesztők százaléka



Forrás: <https://www.jetbrains.com/lp/devecosystem-2019/python/>

Mi mire használjuk?

Röviden: mindenre

Hosszabban:

- *Prototipizálásra*
- *Webes backend fejlesztésre*
- *Kiértékeléshez/adatbányászatra/gépi tanulásra*
- *Rendszer váznak (glue logic)*

```
P_hat[gamma_id] = np.sum(P[gamma_id, :] * p_t)
P_hat_c[gamma_id, :] = (P[gamma_id, :] @ ancestors) * p_t

P_res = np.asmatrix(gid_sid_map[gamma_id, :] * p_s)

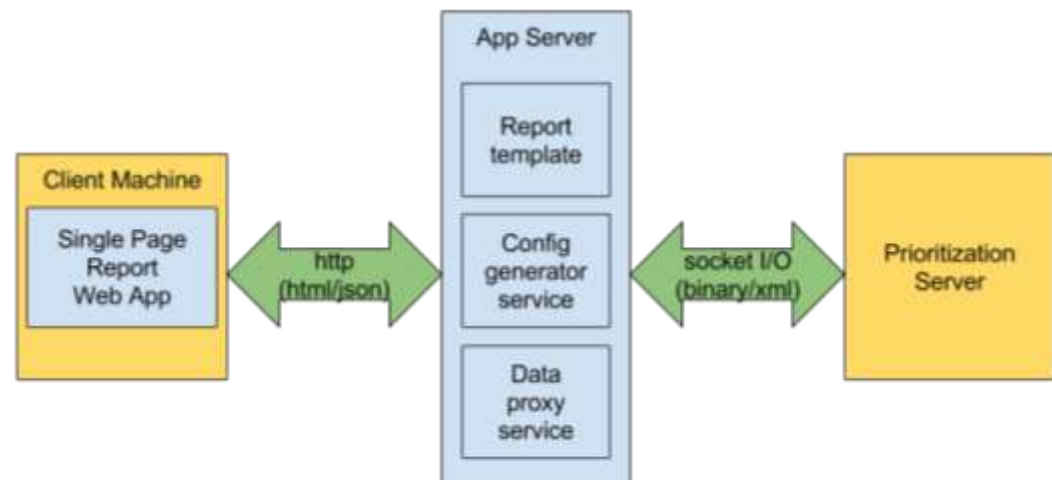
P_res += P[gamma_id, :] * (E_hat - E_hat_c) / w \
        + E * (P_hat[gamma_id] - P_hat_c[gamma_id, :]) / w
P_res += p_d * 2 * P[gamma_id, :] * E
P_res += p_s * ((E @ left) * (P[gamma_id, :] @ right) \
               + (P[gamma_id, :] @ left) * (E @ right))

for gamma_i in gamma_sub.values():
    if gamma_i.children is not None:
        v_key = self.node_to_gid[gamma_i.children[0]]
        w_key = self.node_to_gid[gamma_i.children[1]]
        P_res += gamma_i.norm_score \
                * p_s * ((P[v_key, :] @ right) * (P[w_key, :] @ left)
                       + (P[w_key, :] @ right) * (P[v_key, :] @ left))
        P_res += gamma_i.norm_score \
                * (p_d * (P[v_key, :] * P[w_key, :]) # D event
                  + (P_hat[w_key] - P_hat_c[w_key, :]) / w * P[v_key, :]
                  + (P_hat[v_key] - P_hat_c[v_key, :]) / w * P[w_key, :])
P[gamma_id, :] = P_res
```


UI for Quantitative Semantic Fusion

Tapasztalatok:

- Django alapokon web oldalak és web servicekkel gyorsan lehet haladni
- Sokszor szükséges nem tiszta egy nyelvű megoldás (C++/JavaScript)
- Apró részleteken múlik a használhatóság (hosszan futó taszkok kezelése)
- OS/Web/WSGI szerver konfiguráció is kritikus

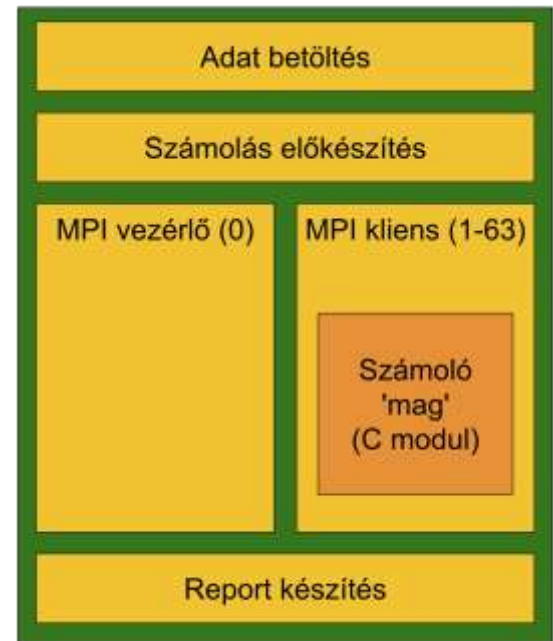


ALENG

(Amalgamated Likelihood Estimation / Next Generation)

Tapasztalatok:

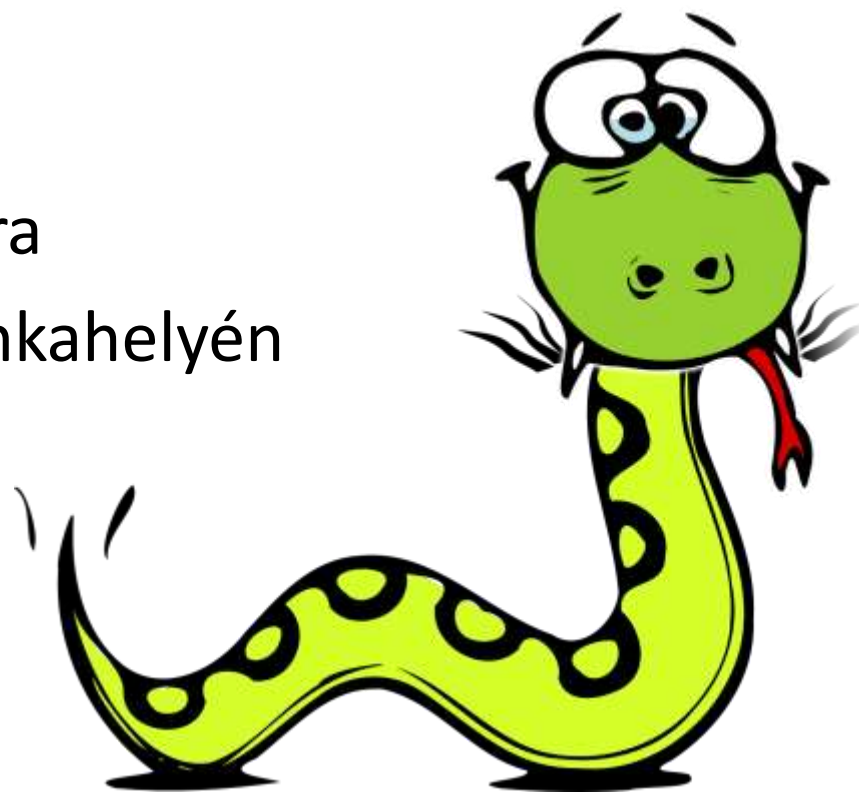
- A rendszer összteljesítménye elsősorban nem biztos, hogy nyelv függő (az új megvalósítás 5x gyorsabb, mint az előző C++ implementáció)
- A technikai implementációs részletek, nagyon tudnak rontani komplex algoritmusok olvashatóságán
- Van, amikor csak a legalacsonyabb szintű API segít (MPI vs Dask vs Spark)



Mire használják a hallgatóink?

250 fős minta

- Tesztelésre
- Rendszer adminisztrációra
- Hobbira (eljött, de a munkahelyén nem tudja használni)
- Adatelemzésre
- Egyéb



Forrás: <https://www.needpix.com/photo/181492/snake-python-serpent-green-reptile-wildlife-animal-predator-vertebrate>

Python jövője

- Beágyazott rendszerekben erősödhet: MicroPython, Circuit Python



Forrás: <https://learn.adafruit.com/welcome-to-circuitpython/overview>

- Nagy teljesítményű (fordított) Python: Numba, MyPyC



- Böngészőbe JavaScript helyett: Web Assembly target



Köszönöm a figyelmet!



gabor.guta@axonmatics.com