

Pythonic programiranje



BY  COMTRADE

Šta je Pythonic programiranje?

Ukratko, „pythonic“ opisuje stil kodiranja koji koristi Pythonove jedinstvene karakteristike za pisanje koda koji je čitljiv i lep.

Čak i oni sa ograničenim programerskim znanjem mogu pogledati deo Python koda i razumeti šta on radi.

Zanimljivo je da nije nužno da je početnik taj koji proizvodi kod koji se može smatrati „unpythonic“. To se može dogoditi programerima koji imaju pozadinu na drugom jeziku.

Kod može biti napisan tako da on funkcioniše, ali pythonic stil nije pitanje “da li nešto funkcioniše” već “da li bi tako trebalo”.

Primer – Otpakivanje torki

Torka je nepromenljiv tip strukture podataka u Python-u, čijim elementima možemo pristupiti na isti način kao i kod bilo koje druge liste.

Neka naša funkcija `grad_info(naziv_grada)` vraća informacije o jednom gradu: geografsku dužinu, širinu i ime grada u okviru jedne torke

```
def grad_info(naziv_grada):  
    #kod funkcije  
    return g_duz, g_sir, ime
```

```
bg_info = grad_info("Beograd")
```

Postojala bi dva načina da pristupimo svakoj od informacija:

```
g_duz = bg_info[0]  
g_sir = bg_info[1]  
ime = bg_info[2]
```

ili

```
g_duz, g_sir, ime = bg_info
```

Ne pythonic

Pythonic

Primer – List comprehension

List comprehension je sintaksička konstrukcija za kreiranje liste na osnovu postojećih lista.

Recimo da imamo listu informacija o gradovima koji su u formi torke, kao u prethodnom primeru, a želimo da kreiramo samo listu imena gradova, to možemo uraditi na dva načina:

- klasičnim prolazom kroz listu i dodavanjem imena pomoću `append()` funkcije ili
- pomoću list comprehension opcije

```
lista_naziva = [ g_ime for g_duz, g_sir, ime in lista_gradova]
```

Pythonic stil programiranje

Ako biste u terminalu ukucali komandu ***python*** pa zatim ***import this*** dobili biste listu izjavnih rečenica koje opisuju filozofiju pisanja koda u Python-u odnosno

The Zen of Python>

<https://www.python.org/dev/peps/pep-0020/>

Pored toga takođe imamo i oficijalni vodič za pisanje koda u Pythonu kao spisak preporuka i konvencija u Python zajednici >

<https://www.python.org/dev/peps/pep-0008/>

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> █
```

Hvala na pažnji

