Miler M., Odobašić D. (2010): Python - Uvod u programiranje za inženjere
Ekscentar, br. 12, str. 108-110

Mala škola programiranja:

Python

Uvod u programiranje za inženjere

» *Mario Miler* » *Dražen Odobašić*

SAŽETAK. Programiranje je jedna od najčešćih vještina koju inženjeri zaobilaze misleći da je to prekomplicirano ili rezervirano samo za vrsne informatičare. Ne uzimaju pritom u obzir da većina skripti i aplikacija koje mogu ubrzati svakodnevne poslove ne zahtijevaju veliko znanje programiranja, a mogu uvelike olakšati svakodnevne rutine. Jedan od tih programsko/skriptnih jezika je Python. U ovom članku neće biti opisano zašto je Python bolji ili lošiji od drugih programskih jezika, već zašto i kada se može koristiti Python kao programski ili skriptni jezik. Python možda nije najbrži i najpopularniji programski jezik, ali definitivno je među najkorisnijima u inženjerskoj struci. Ovu činjenicu su već primijetili neki tehnički fakulteti u Hrvatskoj te su umjesto ili uz postojeće programske jezike u nastavi uveli i Python. Većina inženjera ne programira svakodnevno, već samo u situacijama kada im je potrebna automatizacija radnog procesa te im je upravo zbog toga potreban programski jezik koji je istodobno jednostavan i dovoljno napredan.

KLJUČNE RIJEČI: Python, programiranje, skripte, programski jezik, skriptni jezik

> 1. Python

Python je hibrid između skriptnih jezika poput Perl-a i Scheme i sistemskih programskih jezika poput C, C++ i Jave. Zbog toga omogućava jednostavnost skriptnih jezika, ali istovremeno i napredne programske alate koje možemo naći u sistemskim programskim jezicima. Python je slobodan programski jezik sa vrlo dobrom popratnom dokumentacijom i literaturom. O njegovoj raširenosti i upotrebljivosti govori i činjenica da na internetu možemo pronaći veliki broj primjera i gotovih aplikacija za razne struke. Python kod se može izvršavati na više načina: interaktivni, gdje se kod izvršava u trenutku kada ga pišemo, skriptni, gdje se kod sprema unutar tekstualne datoteke koja se izvršava u trenutku pokretanje skripte (datoteke) i umetnuti, gdje je moguće Python kod izvršavati unutar programa pisanog u drugim programskim jezicima (recimo C-u). U inženjerskoj praksi najčešće se koristi skriptni način izvođenja.

Python programski jezik ne »dolazi« sa standardnom instalacijom Windows operativnog sustava, već se mora posebno instalirati, dok većina distribucija Linux operativnog sustava imaju Python u standardnom instalaciji. Centralni Python repozitorij nalazi se na stranici http://www.python.org/. U trenutku pisanja ovog članka najnovija verzija Python-a je 3.1.1.

> 2. Osnove programiranja

Postavlja se pitanje: »Što je računalni program?« Računalni program je niz instrukcija koje računalo izvršava. Redoslijed instrukcija definiran je zadaćom ili rezultatom koji se želi postići. Programi su različiti, ali su izgrađeni od nekoliko dijelova:

- ulazne kontrole,
- uređaji poput miša, tipkovnice ili datoteke na tvrdom disku računala,
- kontrole izlaza,
- prikaz rezultata na zaslonu računala, spremanje u datoteku, slanje podataka mrežom,
- matematika,
- koriste se osnovne matematičke operacije poput zbrajanja i množenja kako bi se izvršio zadatak,
- uvjetno izvršavanje
- ovisno o zadovoljavanju uvjeta instrukcije će se izvršiti ili ne,
- ponavljanje,
- isti niz instrukcija izvršava se više puta.

Proces u kojem kompleksne zadatke razbijamo na mnogo manjih, koje je moguće izraziti jednostavnim operacijama, naziva se programiranje. Programiranje se uglavnom svodi na ispravljanje grešaka *(engl. bug)*. Moguće je definirati tri skupa grešaka, koje se uklanjaju u procesu koji se naziva *debugging*:

- Sintaksne:
  > najbrže se otkrivaju, npr. krivo napisana instrukcija, umjesto »open« je zapisano »oenp«.
- Izvršne:
  > relativno se brzo otklanjaju, a otkrivaju se prilikom izvršavanja programa, npr. datoteku nije moguće otvoriti, program pokušava dijeliti s nulom.
- Semantičke:
  > najteže ih je otkriti jer se program uspješno izvršava, ali rezultat nije ono što očekujemo, tj. rezultat izvršavanja programa biti će upravo ono što je definirano instrukcijama.

Instrukcije računalnog programa za-

Mario Miler, dipl. ing. geod., Katedra za geoinformatiku, Geodetski fakultet, Sveučilište u Zagrebu, Kačićeva 26, 10000 Zagreb, e-mail: mmiler@geof.hr
Dražen Odobašić, dipl. ing. geod., Katedra za geoinformatiku, Geodetski fakultet, Sveučilište u Zagrebu, Kačićeva 26, 10000 Zagreb, e-mail: dodobas@geof.hr

pisane su u tzv. računalnom kodu koji razumiju računala, ali ne i ljudi. U tu svrhu razvijeni su programski jezici koji omogućavaju razumijevanje i ljudima i računalima. Programski jezik je formalni jezik, odnosno dizajniran za specifičnu primjenu definiranjem striktnih pravila jezika. Osnovne elemente jezika čine simboli i struktura. Simboli su znakovi, riječi i brojevi, a struktura određuje u kojem se odnosu nalaze simboli. Za primjer možemo uzeti matematički izraz: `2+2=5`, koji je formalno ispravan jer svi iskorišteni simboli mogu imati takvu strukturu, iako sam izraz nije točan.

Računalni program koristi i upravlja vrijednostima, a svakoj vrijednosti može se odrediti tip. Uobičajeno se razlikuju jednostavni i složeni tipovi podataka. Tip vrijednosti određuje njezin opseg i operacije koje je moguće izvršiti. Primjeri vrijednosti: `3`, `'Python je super!'`, i `3.14`, imaju sljedeće jednostavne tipove: `int`, `str` i `float`, odnosno cijeli broj, niz znakova i realni broj. Složeni tipovi podataka u Python-u se mogu opisati kao strukture u koje ubrajamo *n-torke*, *liste* i *rječnike*.

Operatori omogućavaju promjenu vrijednosti, to su posebni simboli koji reprezentiraju operacije poput zbrajanja i množenja te imaju određen redoslijed izvršavanja.

Varijabla je naziv koju povezujemo s nekom vrijednošću. Nazivi varijabli definirani su preciznim pravilima. Koriste se kako bi lakše upravljali vrijednostima, odnosno kako bi nekoj vrijednosti dodijelili lako pamtljiv naziv.

Prije spomenute instrukcije programa, odnosno naredbe programa *(engl. statement)*, izgrađene su od izjava *(engl. expression)*. Izjava je kombinacija vrijednosti, varijabli i operatora. Grupirane naredbe nazivaju se blokovima koje izgrađuju funkcije, a funkcije pak objekte. Skup objekata izgrađuje module, a više modula stvara biblioteku.

Distribucija Python-a dolazi s »uključenim baterijama« što znači da uz Python dolazi velik broj modula koje je moguće odmah iskoristiti. Popis modula uključenih u standardnu distribuciju moguće je pogledati na stranici http://docs.python.org/modindex.html.

> 3. Primjeri

Najbolji način za prikaz rada i sintakse nekog programskog jezika je putem primjera. Uzmimo primjer da imamo tekstualnu datoteku s GPS koordinatama i iz nekog nama nepoznatog razloga visine su elipsoidne i potrebno je oduzeti vrijednost undulacije svakoj visini iz te datoteke.

Naravno, ovo se bez problema može napraviti u nekom tabličnom kalkulatoru poput MS Excel-a ili OpenOffice Calc-a. Ukoliko bi primjerice imali 3 milijuna ovakvih točaka, tablični kalkulator nam ne bi bio od velike pomoći.

Sljedećim kodom rješavamo ovaj problem u 15-ak linija koda.

```
file1 = "c:\\tocke.txt"
file2 = "c:\\obrada _ tocaka.txt"
undulacija = 45.4
```

U prve tri linije koda definirane su tri varijable, `file1`, `file2` i `undulacija`. Prve dvije varijable su tekstualnoga tipa *(engl. string)* dok je treća realni broj *(engl. float)*. Varijabla `file1` sadrži lokaciju tekstualne datoteke na tvrdom disku koju želimo obraditi, dok varijabla `file2` sadrži lokaciju nove datoteke u koju spremamo rezultat obrade. Varijabla `undulacija` sadrži vrijednost undulacije. Ovdje se već može primijetiti Python-ova jednostavnost, nije potrebno definirati tip varijable, već Python određuje tip varijable po onome što se u njoj nalazi. Ovo je karakteristika dinamičnih programskih jezika.

```
fr = open (file1, "r")
fw = open (file2, "w")
```

U ovom dijelu stvaramo dvije varijable `fr` i `fw` koje sadrže vezu na prije navedene datoteke. Ova veza se stvara pomoću `open()` metode koja zahtijeva dva parametra, gdje se ta datoteka nalazi (`file1` i `file2`) te način pristupa datoteci, `r` označava samo čitanje datoteke *(engl. read)*, `w` označava upisivanje u datoteku *(engl. write)*.

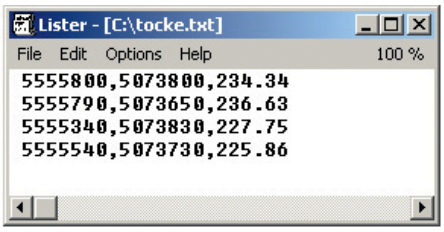
```
for red in fr.readlines():
```

Na ovoj liniji definiran je početak `for` petlje. Ova petlja se koristi kako bismo prošli preko svakog reda u tekstualnoj datoteci *tocke.txt*. Da bismo mogli pregledavati svaki red u našoj datoteci, moramo pozvati metodu `readlines()` veze na datoteku (varijabla `fr`). Varijabla *red* definira vrijednost svakog pojedinog reda tokom pregledavanja tekstualne datoteke, red po red.

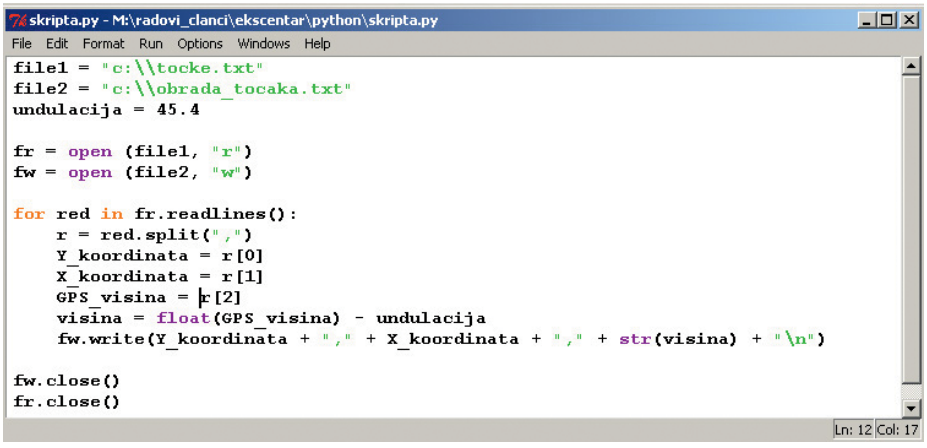
```
r = red.split(",")
```

U Python-u se ne koriste zagrade za odvajanje blokova koda, niti se redovi odvajaju znakovima poput točka-zarez (;), kao kod nekih programskih jezika. Python razlikuje logičke dijelove pomoću uvučenih *(engl. tab)* dijelova koda. Struktura se definira razmacima ili tabulatorima. Ovaj dio koda djelomično je uvučen što znači da se taj dio koda izvršava unutar gore navedene petlje.

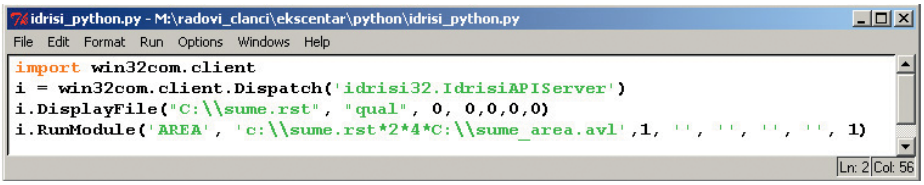
Varijabla *red* sadrži jednu liniju iz tekstualne datoteke koja ima tekstualni tip podatka (npr. "5555800,5073800,234.34"). S obzirom da sa tekstualnim tipom podatka ne možemo raditi računske operacije te da su sve koordinate zajedno, prvo ih razdvojimo. To radimo metodom `split()` koja zahtijeva jedan parametar koji definira razdjelnik, a u našem slučaju on je zarez (,). Kao rezultat ove metode dobivamo listu koju spremamo u varijablu `r` (npr. ["5555800" "5073800" "234.34"]). Elementi ove liste su brojevi koji su tekstualnom tipu podatka. Nulti element ove liste je "5555800", prvi "5073800" i drugi "234.34". Ovdje se može primijetiti da tip podatka lista sve elemente počinje brojati od nule, a ne od jedan kako je nama prirodno.



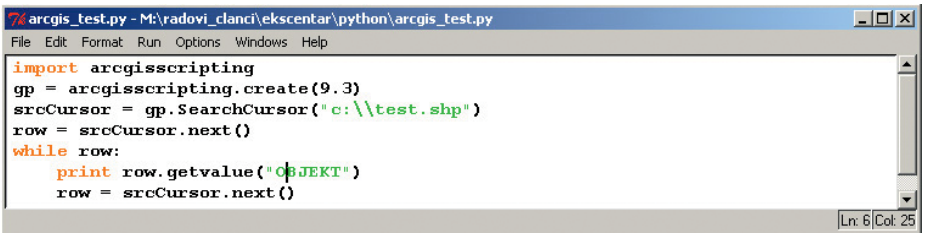
Slika 1. Primjer tekstualne datoteke



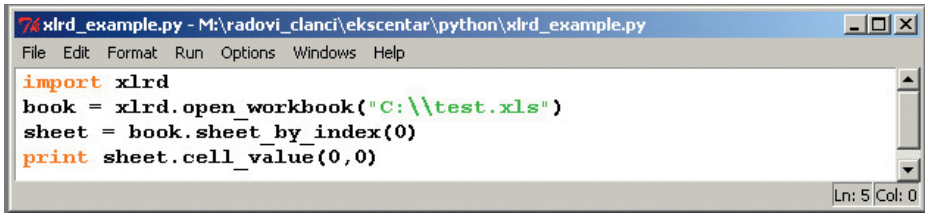
Slika 2. Rješenje problema u Python kodu



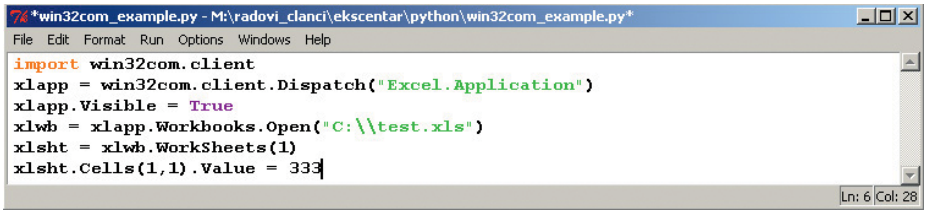
Slika 3. Primjer koda koji koristi IDRISI aplikaciju



Slika 4. Primjer koda koji koristi MS Excel aplikaciju



Slika 5. Primjer koda koji koristi xlrd modul za upravljanje Excel datotekama



Slika 6. Primjer koda koji koristi ESRI ArcGIS apilkaciju

```
Y_koordinata = r[0]
X_koordinata = r[1]
GPS_visina = r[2]
```

Sljedeće tri linije koda spremaju Y i X koordinatu te GPS visinu u zasebne varijable koje smo nazvali `Y_koordinata`, `X_koordinata` i `GPS_visina`. Ovdje možemo primijetiti sintaksu pristupa pojedinim elementima liste `r`. Ukoliko želimo odabrati prvi element liste r, pišemo `r[0]`, za drugi element `r[1]` itd. Liste u Python-u su »zero-based« tj. prvi element ima indeks 0.

```
visina = float(GPS_visina) - undulacija
```

Varijabla `visina` u ovom trenutku dobiva vrijednost razlike između GPS visine (varijabla `GPS_visina`) i undulacije (varijabla `undulacija`). S obzirom da nam je varijabla `GPS_visina` tekstualnoga tipa, a varijabla undulacija realnoga tipa, nije moguće oduzeti te dvije varijable bez da prvo pretvorimo tekstualni (u našem slučaju `GPS_visina`) u realni tip. U tu svrhu koristimo metodu `float()`.

```
fw.write(Y_koordinata + "," + X_koordinata + "," + str(visina) + "\n")
```

Korištenjem metode `write` varijable `fw` (koja predstavlja našu novu datoteku) upisujemo jednu liniju u novu tekstualnu datoteku (obrada_tocaka.txt). Kao i u prethodnom slučaju kada nismo mogli oduzeti dva različita tipa vrijednosti, tako i ovdje nije moguće spojiti visinu (koja je realni tip) sa `Y_koordinata` i `X_koordinata` (koje su tekstualni tip). Prvo moramo pretvoriti *visinu* u tekstualni tip vrijednosti, za što koristimo metodu `str()`. Oznaka »\n« označava kraj reda i da će sljedeći zapis ići u novi red tj. kao da smo pritisnuli tipku *Enter* za kraj reda.

```
fw.close()
fr.close()
```

Ovaj dio koda nalazi se izvan *for* petlje, a to znači da se naredba izvršava tek kada se završi izvršavanje petlje. Kako smo na početku stvorili veze prema datotekama (pomoću metode `open()`), na isti način ih MORAMO i zatvoriti. U protivnom će te veze ostati otvorene i datoteke će biti zaključane bez obzira da li se kod izvršio ili ne. Zatvaranje datoteka radimo metodom `close()`.

Ukoliko sada pokrenemo naš kod *(engl. Run)*, program bi se trebao izvršiti. U slučaju greške, Python će javiti gdje se i koja greška pojavila.

Ovo je jedan vrlo jednostavan primjer korištenja Python-a u inženjerskoj struci. Python također ima veliku prednost u tome što se može iskoristiti u drugim programima kao MS Excel, IDRISI ili ArcGIS. Sljedeći primjeri pokazuju neke od njih,a može se primijetiti koliko su operacije jednostavnije upotrebom Python-a u samo nekoliko linija koda.

Ovaj kod pokreće IDRISI aplikaciju te otvara rastersku datoteku *sume.rst*. Nakon toga izračunava površinu određenih šumskih područja u toj datoteci. Kao što se može primijetiti, ovo je napravljeno u samo četiri linije koda.

Slično prethodnom primjeru, ovaj kod pokreće MS Excel aplikaciju i otvara datoteku *test.xls* koja se nalazi na tvrdom disku. Nakon toga u ćeliju A1 (1,1) upisuje broj 333.

U ovom primjeru koristimo *xlrd* modul koji omogućava čitanje MS Excel datoteke bez potrebe pokretanja Excel aplikacije. *xlrd* modul nije standardni Python-ov modul već se mora dodatno instalirati. Nalazi se na stranici http://pypi.python.org/pypi/xlrd. Ovim kratkim kodom čitamo podatak koji se nalazi u ćeliji A1 (0,0) iz *test.xls* datoteke.

Ovaj primjer koristi ArcGIS aplikaciju (objekte) pomoću koje čitamo atributne podatke (naziv polja OBJEKT) iz *test.shp* datoteke. Na sličan način koristi se većina ArcGIS modula za automatizaciju radnog procesa.

> Literatura

» Lutz, M.: Programming Python, Third Edition, O'Reilly, Sebastopol.
» Elkner J., Downey A. B., Meyers C.: Learning with Python 2nd Edition, OpenBookProject.
» URL-1: http://www.python.org/ (15.01.2010.).
» URL-2: http://docs.python.org/modindex.html (15.01.2010.).
» URL-3: http://geochalkboard.wordpress.com/2008/02/05/scripting-your-arcgis-geoprocessing-tasks-part-1/ (15.01.2010.).
» URL-4: http://arcscripts.esri.com/ (15.01.2010.).
» URL-5: http://pypi.python.org/pypi/xlrd (15.01.2010.).
» URL-6: http://oreilly.com/catalog/pythonwin32/chapter/ch12.html (15.01.2010.).
» URL-7: http://www.clarklabs.org/support/IDRISI-Applications-Programming-Interface.cfm (15.01.2010.).