

Универзитет у Београду
Грађевински факултет
Катедра за геодезију и геоинформатику



ГИС ПРОГРАМИРАЊЕ

Материјали за вежбе

1

Наставник: Жељко Цвијетиновић

Асистент: Јован Ковачевић

Београд, 2017.

САДРЖАЈ

1.	Увод у <i>Python</i>	3
2.	Синтакса језика. Типови података. Основне алгоритамске структуре.	4
2.1.	Покретање програма, коментари, исписивање на стандардном излазу	4
2.2.	Променљиве, типови података, форматиран испис, учитавање са стандардног улаза, аритметичке и логичке операције, наредбе гранања, петље.	4
2.3.	Учитавање модула. <i>Math</i> и <i>random</i> модули.	6
3.	Задаци за вежбу	7

1. Увод у *Python*

Python је платформски независан, објектно оријентисан, интерпретерски и интерактивни програмски језик. Настао је 1990. а његов развој је започео *Guido van Rossum*. Име језика потиче од британске телевизијске комедије *Monty Python's Flying Circus*, чиме је *van Rossum* желео да нагласи свој основни циљ при прављењу новог језика, да програмирање постане забавно као и поменута серија. *Van Rossum* уопште није љубитељ змија, јер се ради о животињама предаторима.

Python спада у групу софтвера отвореног кода (дозвољена је дистрибуција, преглед и измена изворног кода, интеграција са другим бесплатним програмима). Има добру подршку, документацију и пратеће библиотеке. Ради се о програмском језику високог нивоа, односно програмер не мора да води рачуна о детаљима на ниском нивоу попут расподеле и коришћења меморије, цурење меморије ...

Језик је нашао примену у великом броју области. Представља одличан "лепак језик", језик којим се повезују постојећи програми, па им се преко *Python*-а даје заједнички интерфејс. *BitTorrent* клијент је изворно написан у њему. Први *web*-робот за *Google* је писан у *Python*-у, а та компанија и даље користи много његовог кода (*Guido van Rossum* је већ неколико година запослен у *Google*-у). *Youtube* је заснован на *Python*-у, а користи га и *NASA*.

Тренутно актуелна верзија је **3.6**, али је веома заступљена и верзија **2.7**.

Најзначајније карактеристике:

- није строго типизиран језицик, односно нема изразитих разлика међу примитивним типовима: *integer, float, char*;
- инструкције се интерпретирају, а не компајлирају у одговарајући извршни код;
- чланови низова могу бити различитих типова функције могу вратити неskalарне вредности, односно низове индекси петљи могу бити неskalарне вредности;
- имају имплементацију бројних структура и операција на виском нивоу што је својствених природним језицима (нпр. речници).

ЛИТЕРАТУРА:

- <https://www.python.org/>
- <http://www.tutorialspoint.com/python>
- <https://wiki.python.org/moin/>

2. Синтакса језика. Типови података. Основне алгоритамске структуре.

2.1. Покретање програма, коментари, исписивање на стандардном излазу

```
# Ovako se pisu komentari

# Sledeći red definise kodiranje skripta
# -*- coding: utf-8 -*-

# pokretanje programa/skripta iz windows terminala
# $ python D:\GisP_vezbe\Hello.py
# pokretanje iz python shell-a
execfile('D:\GisP_vezbe\Pozdrav.py')

# ispisivanje na standardnom izlazu
print "Zdravo svete!"

# informacije(help) python funkcije
# help('if')
```

2.2. Променљиве, типови података, форматиран испис, учитавање са стандардног улаза, аритметичке и логичке операције, наредбе гранања, петље.

```
# Promenljive se dinamički tipiziraju
a = 100
b = 11.22
istina = True
niska = "Ovo je tekst"
# Niske su konstantne tj. nisu promenljive.
# To znaci da svaka operacija nad niskom pravi novu nisku u
memoriji.

# Ispisivanje tipa promenljive
type(a)
type(b)

# Odredjivanje tipa promenljive funkcija isinstance()
isinstance(a, float)
isinstance(a, int)

# Konverzija izmedju razlicitih tipova
type(a) #pre konverzije
float(a)
type(a) #posle konverzije

# Ispis na standardni izlaz
print a
print b
print a, b, istina

# Formatiran ispis
print "\n-----Formatiran ispis -----\\n"
print "Ceo broj: {0:d} \\nBroj u pokretnom zarezu {1:f} \\nBulovska
vrednost: {2:b}\\nNiska: {3:s}\\n".format(a,b,istina ,niska)

# Postoje specijalni karakteri prilikom stampanja
# "\\t" - tab, "\\n" - novi red ...
```

```

# Ucitavanje niske sa standardnog ulaza
# Razlika izmedju input i raw_input funkcija
broj = input("Unesite ceo broj: ") # automatsko tipiziranje
string_broj = raw_input("Unesite ceo broj: ") # kao string
broj = int(string_broj) # vrsi se konverzija stringa u ceo broj ,
slicno: float , str

# Osnovne aritmetickie operacije:
# +, -, *, /, %, ** (stepenovanje)
print broj+4

# Osnovne logicke operacije:
# not , and , or
print istina or False

# Blokovi se ne ogranicavaju viticastim zagradama kao sto je u Javi
# vec moraju biti uvuceni tabulatorom.

# Funkcije obezbedjene u okviru python-a
#abs(), int(), max(), min() ...
#kompletna lista obezbeđenih fukcija:
https://docs.python.org/2/library/functions.html

# Naredba grananja
print "\n-----Naredba granjanja -----\\n"
if broj%2 == 0:
    print "Unet je paran broj \\n"
elif broj%3 == 0:
    print "Unet je broj deljiv sa 3\\n"
# Naredbi <<elif >> moze biti vise
else:
    print "Unet je broj koji nije ni paran ni deljiv sa 3\\n"

# Naredba <<switch >> ne postoji

# Petlje
# while petlja
i=1
while i<=10:
    print i
    i=i+1 # i++ ne postoji , moze ili ovako ili i+=1

# for petlja
for x in range(0, 3):
    print "We're on time %d" % (x)

#for ... else pelja
for x in range(0, 3):
    print "We're on time %d" % (x)
else:
    print 'kraj'

# Naredba <<break >> iskace iz bloka
# Naredba <<pass >> preskace blok

```

2.3. Учитавање модула. *Math* и *random* модули.

```
# Matematicke funkcije
# Obezbedjene u okviru modula <<math >>
import math # Ucitavanje modula

# U ovom moduli se nalaze brojne funkcije kao sto su:
math.sqrt(broj)
# math.log(broj , osnova)
# math.sin(ugao_u_radijanima), math.cos(), ...
# math.exp(stepen)
# math.factorial(broj)
# i druge...

print math.factorial (6)
print math.log(125, 5)

# Pseudo slucajni brojevi

# Obezbedjeno u okviru modula <<random >>

import random
# Funkcija random() vraca pseudo slucajan broj tipa float iz opsega
[0.0, 1.0)

print "Pseudo slucajan broj iz opsega [0.0 ,1.0)\n"
print random.random()

# Korisne funkcije:

#randint(a,b) - vraca pseudo slucajan ceo broj n iz opsega [a,b]
# choice(lista) - vraca pseudo slucajan element iz liste
```

3. Задаци за вежбу

1. Написати програм који на стандардни излаз исписује текст “Здраво, геоинформатичари!” (ћириличним писмом).
2. Написати програм који за унета два цела броја на стандардни излаз исписује најпре унете вредности, а затим и њихов збир, разлику, производ, цео део при дељењу првог броја другим бројем и остатак при дељењу првог броја другим бројем. Претпоставити да је унос коректан, тј. да друга унета вредност није 0, као и да се уносе целобројне вредности са стандардног улаза.
3. Написати програм који за унете вредности два правца у облику степени : минути : секунде, исписује угао који та два правца граде. Угао треба да буде у децималном запису, заокружен на 4 децимале.
4. Са стандардног улаза се уносе два четвороцифрена броја. Написати програм који на стандардни излаз исписује суму цифара другог броја, а потом и разлику збира цифара првог броја на парним и непарним позицијама.
5. Написати програм који за унети петоцифрени број исписује његову највећу цифру.
6. Програм учитава пет карактера. Исписати колико пута су се појавиле цифре.
7. Написати програм који испитује да ли се дата тачка $M(X_m, Y_m)$ налази унутар троугла чија су темена тачке $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$ и исписује одговор ДА или НЕ.
8. Написати програм који имплементира игру ”Погоди број”. На почетку игре рачунар замишља један случајан број у интервалу $[0,100]$. Након тога играч уноси своје име и започиње игру. Играч уноси један по један број све док не погоди који број је рачунар замислио. Сваки пут када играч унесе број, у зависности од тога да ли је број који је унет већи или мањи од замишљеног броја исписује се одговарајућа порука. Игра се завршава у тренутку када играч погодио замишљен број, када се на стандардном излазу приказује број покушаја из којег је број погођен.