

VJEŽBA 2: RAD S DODATNIM PYTHON BIBLIOTEKAMA.

I. Cilj vježbe: *Upoznati se s načinom korištenja dodatnih Python biblioteka/modula: re, Numpy, Matplotlib.*

II. Opis vježbe:

Za Python je napisano velik broj biblioteka koje olakšavaju programiranje. U ovoj vježbi studenti se upoznaju s regularnim izrazima (biblioteka *re*) koji olakšavaju pretraživanje znakovnih nizova, zatim *numpy* bibliotekom koja olakšava numeričke proračune, *matplotlib* bibliotekom za grafički prikaz rezultata.

II.1. Re

Biblioteka *re* sadrži implementaciju regularnih izraza za Python programski jezik. [Regularni izraz](#) je niz znakova koji definiraju uzorak koji se traži unutar određenog znakovnog niza. Regularni izraz sastoji se od meta znakova (posebno značenje znaka) i znakova koji imaju svoje normalno (literalno značenje). U [dodatku VI](#) mogu se pronaći najvažniji meta znakovi i njihovo značenje u regularnom izrazu.

Pomoću metode [find.all](#) moguće je izdvojiti dijelove znakovnog niza koji zadovoljavaju dani regularni izraz. Metoda vraća listu.

Primjer 2.1.

```
import re

stringS = 'bla bla bu zla fla bu ble bik elp ble ble bub Zabluda'
lst = re.findall('bl.', stringS)      # točka predstavlja bilo koji znak
print(lst)
```

Primjer 2.2.

```
import re

stringS = 'ovo su neke ispravne i neispravne @mail adrese ivan@firma.hr marko@firma.hr marko@ jura@li.hr'
lst = re.findall('m\S*@\S+', stringS)      # izdvoji ispravne mail adrese koje počinju sa slovom „m“
print(lst)
```

II.2. Numpy

[Numpy](#) je open source Python biblioteka za numeričke proračune. Sadrži niz brzih prekompajliranih funkcija za različite numeričke rutine. Ovo je važno kada se razmatraju problemi strojnog učenja u kojima se provode različite optimizacije na temelju raspoloživih podataka. Prednosti korištenja Numpy biblioteke u odnosu na osnovni Python su efikasno (brzo) matricno računanje te efikasna implementacija multidimenzionalnih polja. Osnovna struktura Numpy biblioteke je polje (array) koje je slična python listi, ali svaki element polja je istog tipa (float ili integer).

Naredbom `import numpy` učitava se numpy biblioteka. Tada se raspoloživim funkcijama pristupa na način `numpy.funkcija`. Stoga se često zbog kraćeg zapisa koristi naredba `import numpy as np` pa se funkcijama pristupa na način `np.funkcija`.

Numpy polje (array) je matrica/vektor čiji elementi su indeksirani pozitivnim cijelim brojevima. Definira se pozivanjem funkcije `array`. Iz polja je moguće izdvajati dijelove polja pomoću operatora `:`. Nadalje, postoje funkcije za kreiranje polja koje sadrži nule ([zeros](#)), jedinice ([ones](#)), konstantu ([full](#)), jediničnu matricu ([eye](#)) i sl. Neki primjeri dani su u nastavku

Primjer 2.3.

```
import numpy as np

a = np.array([6, 2, 9])      # napravi polje od tri elementa
print(type(a))              # prikaži tip polja
```

```

print(a.shape)          # koliko redaka ima vektor
print(a[0], a[1], a[2]) # prikaži prvi, drugi i treći element
a[1] = 5                 # promijeni vrijednost polja na drugom mjesti
print(a)                 # prikaži cijeli a
print(a[1:2])            # izdvajanje
print(a[1:-1])           # izdvajanje

b = np.array([[3,7,1],[4,5,6]]) # napravi 2 dimenzionalno polje (matricu)
print(b.shape)             # ispiši dimenzije polja
print(b)                   # ispiši cijelo polje b
print(b[0, 2], b[0, 1], b[1, 1]) # ispiši neke elemente polja
print(b[0:2,0:1])         # izdvajanje
print(b[:,0])              # izdvajanje

c = np.zeros((4,2))        # polje koje sadrži 0
d = np.ones((3,2))         # polje koje sadrži 1
e = np.full((1,2),5)       # polje koje sadrži konstantu 5
f = np.eye(2)              # jedinična matrica 2x2

g = np.array([1, 2, 3], float)
duljina = len(g)
print(duljina)
h = g.tolist()
print(h)
c = g.transpose()
print(g)
np.concatenate((a, g,))

```

Unutar Numpy dostupne su matematičke operacije s numpy poljima poput zbrajanja, množenja i ostalih uobičajenih matematičkih operacija, zatim funkcija za sumiranje polja, traženje maksimalne/minimalne vrijednosti unutar polja, sortiranje i sl.

Primjer 2.4.

```

import numpy as np

a = np.array([3,1,5], float)
b = np.array([2,4,8], float)
print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a.min())
print(a.argmax())
print(a.maxn())
print(a.argmax())
print(a.sum())
print(a.mean())
print(a.var())
print(a.prod())
a.sort()
print(a)

np.dot(a,b)

```

Unutar Numpy biblioteke nalaze se osnovne „statističke“ funkcije poput srednje vrijednosti, varijance i standardne devijacije. Također je na raspolaganju generator slučajnih brojeva (različite distribucije).

Primjer 2.5.

```

import numpy as np

```

```
np.random.seed(56)          # postavi seed generatora brojeva
rNumbers = np.random.rand(10) # generiraj 10 slučajnih brojeva iz uniformne razdiobe
print(rNumbers)
print(rNumbers.mean())
```

II.3. Matplotlib

Matplotlib biblioteka sadrži različite funkcije za grafički prikaz. Najvažnija je funkcija `pyplot` koja omogućava dvodimenzionalni prikaz podataka te je vrlo slična Matlabovom grafičkom frameworku.

Primjer 2.6.

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, num=30)
plt.plot(x, np.sin(x), '--', linewidth=1)
plt.plot(x, np.cos(x+np.pi/3), 'r-', linewidth=2)
plt.xlabel('x')
plt.ylabel('vrijednosti funkcije')
plt.title('sin i cos funkcija')
```

Primjer 2.7.

```
import numpy as np
import matplotlib.pyplot as plt

mu = 3
sigma = 2
v = np.random.normal(mu, sigma, 10000)
plt.hist(v, bins=50, normed=1)
```

Biblioteka omogućava i učitavanje *png* slika. Primjer 2.8. učitava *grayscale* sliku 'tiger.png' i prikazuje ju. Učitana slika predstavljena je numpy poljem [img](#).

Primjer 2.8.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

img = mpimg.imread('example.png')
imgplot = plt.imshow(img)
```

III. Priprema za vježbu:

1. Proučite kako se koriste regularni izrazi (također provjerite i [VI Dodatak](#)).

IV. Rad na vježbi:

1. Isprobajte Python primjere iz [II. Opis vježbe](#) u Spyder IDE. Razmislite o svakoj liniji programskog koda i što je njen rezultat. Pogledajte u Variable Explorer Spyder IDE sve varijable koje ste kreirali nakon pokretanja svakog primjera.

2. Klonirajte vaš repozitorij `rusu_lv_2019_20` na računalo pomoću `gitbash`. Zatim povucite moguće promjene iz izvornog repozitorija pomoću naredbi:

```
git remote add upstream https://gitlab.com/rgrbic/rusu_lv_2019_20
git fetch upstream
git merge upstream/master
```

3. Riješite dane zadatke, pri čemu Python skripte trebaju imati naziv `zad_x.py` (gdje je `x` broj zadatka) i trebaju biti pohranjene u direktorij `rusu_lv_2019_20/LV2/solutions/`. Svaki zadatak rješavajte u zasebnoj *git* grani koju spojite s glavnom granom kada riješite pojedini zadatak. Pohranite skripte u lokalnu *git* bazu kao i u `rusu_lv_2019_20` repozitorij na vašem korisničkom računu. Svaki puta kada naćinite promjene koje se spremaju u *git* sustav napišite i odgovarajuću poruku prilikom izvršavanja `commit` naredbe.
4. Nadopunite postojeću tekstualnu datoteku `rusu_lv_2019_20/LV2/Readme.md` s kratkim opisom vježbe i kratkim opisom rješenja vježbe te pohranite promjene u lokalnu bazu. Na kraju pohranite promjene u udaljeni repozitorij.

Zadatak 1

Napišite Python skriptu koja će ućitati tekstualnu datoteku, pronaći valjane email adrese te izdvojiti samo prvi dio adrese (dio ispred znaka `@`). Koristite odgovarajući regularni izraz. Koristite datoteku `mbox-short.txt`. Ispišite rezultat.

Zadatak 2

Na temelju rješenja prethodnog zadatka izdvojite prvi dio adrese (dio ispred `@` znaka) samo ako:

1. sadrži najmanje jedno slovo `a`
2. sadrži toćno jedno slovo `a`
3. ne sadrži slovo `a`
4. sadrži jedan ili više numerićkih znakova (`0 – 9`)
5. sadrži samo mala slova (`a-z`)

Zadatak 3

Napravite jedan vektor proizvoljne dućine koji sadrži slučajno generirane cjelobrojne vrijednosti `0` ili `1`. Neka `1` oznaćava mušku osobu, a `0` žensku osobu. Napravite drugi vektor koji sadrži visine osoba koje se dobiju uzorkovanjem odgovarajuće distribucije. U slučaju muških osoba neka je to Gaussova distribucija sa srednjom vrijednoću `180` `cm` i standardnom devijacijom `7` `cm`, dok je u slučaju ženskih osoba to Gaussova distribucija sa srednjom vrijednoću `167` `cm` i standardnom devijacijom `7` `cm`. Prikažite podatke te ih obojite plavom (`1`) odnosno crvenom bojom (`0`). Napišite funkciju koja računa srednju vrijednost visine za muškarce odnosno žene (probajte izbjeći `for` petlju pomoću funkcije `np.dot`). Prikažite i dobivene srednje vrijednosti na grafu.

Zadatak 4

Simulirajte `100` bacanja igraće kocke (kocka s brojevima `1` do `6`). Pomoću histograma prikažite rezultat ovih bacanja.

Zadatak 5

U direktoriju `rusu_lv_2019_20/LV2/resources` nalazi se datoteka `mtcars.csv` koja sadrži različita mjerenja provedena na `32` automobila (modeli `1973-74`). Prikažite ovisnost potrošnje automobila (`mpg`) o konjskim snagama (`hp`). Na istom grafu prikažite i informaciju o težini pojedinog vozila. Ispišite minimalne, maksimalne i srednje vrijednosti potrošnje automobila.

Zadatak 6

Na temelju primjera 2.8. ućitajte sliku `'tiger.png'`. Manipulacijom odgovarajuće `numpy` matrice pokušajte posvijetliti sliku (povećati *brightness*).

V. Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom rusu_lv_2019_20 koji sadrži rješenja unutar direktorija rusu_lv_2019_20/LV2/solutions/.

VI. Dodatak

Meta znakovi u regularnim izrazima

\wedge	Početak znakovnog niza
$\$$	Kraj znakovnog niza
\cdot	Bilo koji znak
$\backslash s$	Prazan znak
$\backslash S$	Bilo koji znak osim praznog znaka
$*$	Pojavljivanje znaka 0 ili više puta
$+$	Pojavljivanje znaka najmanje jednom
$[abc]$	Bilo koji znak iz skupa*
$[0-9]$	Bilo koja znamenka iz raspona**
$[^abc]$	Bilo koji znak koji nije u skupu
abc	Konkretni znakovni niz „abc“
$()$	Izdvajanje rezultata iz reg. izraza

* skup znakova $[abc]=\{a,b,c\}$

** raspon znamenki $[0-9] = \{0,1,2,\dots,9\}$, $[a-z] = \{a,b,c,\dots,z\}$, $[a-zA-Z] = \{a,b,c,\dots,z,A,B,C,\dots,Z\}$

Dodatni zadatak: centralni granični teorem

Simulirajte bacanje igraće kocke 30 puta. Izračunajte srednju vrijednost rezultata i standardnu devijaciju. Pokus ponovite 1000 puta. Prikažite razdiobu dobivenih srednjih vrijednosti. Što primjećujete? Kolika je srednja vrijednost i standardna devijacija dobivene razdiobe? Što se događa s ovom razdiobom ako pokus ponovite 10000 puta?