

VJEŽBA 2: RAD S DODATNIM PYTHON BIBLIOTEKAMA.

I. Cili vježbe: *Upoznati se s načinom korištenja dodatnih Python biblioteka/modula: Numpy i Matplotlib.*

II. Opis vježbe:

Za Python je napisano velik broj biblioteka koje olakšavaju programiranje. U ovoj vježbi studenti se upoznaju s *numpy* bibliotekom koja olakšava numeričke proračune i *matplotlib* bibliotekom za grafički prikaz rezultata.

II.1. 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) matrično 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 2.1.

```
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 mjestu
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 2.2.

```
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.argmin())
print(a.max())
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 2.3.

```
import numpy as np

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

II.2. Matplotlib

Matplotlib biblioteka sadrži različite funkcije za grafički prikaz. Najčešće korištena funkcija je `pyplot` koja omogućava dvodimenzionalni prikaz podataka.

Primjer 2.4.

```
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')
plt.show()
```

Primjer 2.5.

```
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)
plt.show()
```

Biblioteka omogućava i učitavanje *png* slika. Primjer 2.6. učitava *grayscale* sliku 'tiger.png' i prikazuje ju. Učitana slika predstavljena je numpy poljem *img*.

Primjer 2.6.

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

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

III. Priprema za vježbu:

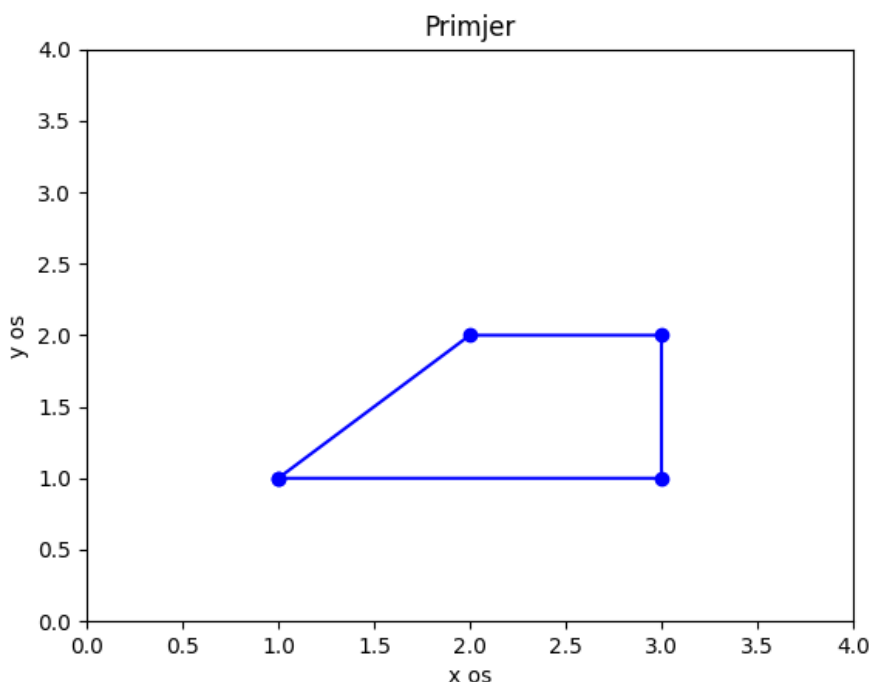
1. Proučite primjere 2.1. do 2.5. Možete li opisati što je rezultat izvođenja svake linije koda u navedenim primjerima?
2. Proučite kako se zapisuje slika u sivim tonovima u memoriju računala.

IV. Rad na vježbi:

1. Isprobajte Python primjere iz II. Opis vježbe u Visual Studio Code IDE. Razmislite o svakoj liniji programskog koda i što je njen rezultat. Pokrenite primjere u *Debug* modu i pogledajte u *Explorer*-u kako izgleda svaka od varijabli u danim primjerima.
2. Klonirajte vaš repozitorij `PSU_LV` na računalo pomoću `git clone` naredbe. Kreirajte direktorij `LV2` unutar direktorija `PSU_LV`. U ovaj direktorij kopirajte sve datoteke vezane za ovu vježbu, a koje se nalaze na loomen stranici predmeta pod `LV2`.
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 `PSU_LV/LV2/`. Svaki zadatak rješavajte u zasebnoj *git* grani koju spojite s glavnim granom kada riješite pojedini zadatak. Pohranite skripte u lokalni *git* repozitorij kao i u `PSU_LV` repozitorij na vašem gitlab računu. Svaki puta kada načinite promjene koje se spremaju u *git* sustav napišite i odgovarajuću poruku prilikom izvršavanja `commit` naredbe.
4. Nadopunite postojeću tekstualnu datoteku `PSU_LV/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

Pomoću funkcija `numpy.array` i `matplotlib.pyplot` pokušajte nacrtati sljedeću sliku:



Igrajte se sa slikom, promijenite boju oblika, debljinu linije i sl.

Zadatak 2

Simulirajte 100 bacanja igraće kocke (kocka s brojevima 1 do 6) pomoću for petlje i funkcije `numpy.random.randint`. Na zasloni ispišite koliko puta se pojavio pojedini broj. Pomoću histograma prikažite rezultat ovih bacanja.

Zadatak 3

U direktoriju `PSU_LV/LV2/` nalazi se datoteka `mtcars.csv` koja sadrži različita mjerenja provedena na 32 automobila (modeli 1973-74). Opis pojedinih varijabli nalazi se u datoteci `mtcars_info.txt`.

- Učitajte datoteku `mtcars.csv` pomoću:

```
data = np.loadtxt(open("mtcars.csv", "rb"), usecols=(1,2,3,4,5,6), delimiter=",", skiprows=1)
```
- Prikažite ovisnost potrošnje automobila (`mpg`) o konjskim snagama (`hp`) pomoću naredbe `matplotlib.pyplot.scatter`.
- Na istom grafu prikažite i informaciju o težini pojedinog vozila (npr. veličina točkice neka bude u skladu sa težinom `wt`).
- Izračunajte minimalne, maksimalne i srednje vrijednosti potrošnje (`mpg`) automobila.
- Ponovite zadatak pod d), ali samo za automobile sa 6 cilindara (`cyl`).

Zadatak 4

Na temelju primjera 2.8. učitajte sliku 'tiger.png'. Manipulacijom odgovarajuće numpy matrice pokušajte:

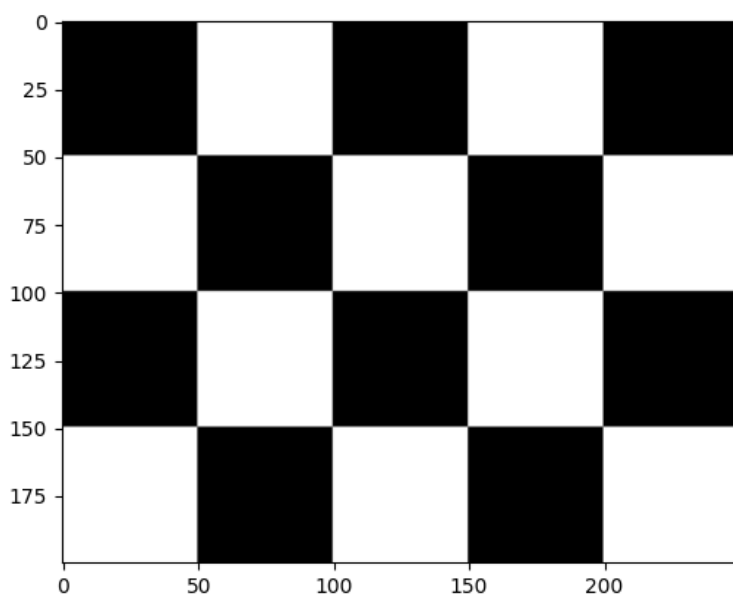
- posvijetliti sliku (povećati *brightness*),
- zarotirati sliku za 90 stupnjeva u smjeru kazaljke na satu,
- zrcaliti sliku,
- smanjiti rezoluciju slike x puta (npr. 10 puta),

- e) prikazati samo drugu četvrtinu slike po širini, a prikazati sliku cijelu po visini; ostali dijelovi slike trebaju biti crni.

Zadatak 5

Napišite funkciju koja kao povratnu vrijednost daje sliku (polje) sa crno bijelim kvadratima jednake dimenzije koji se naizmjenično pojavljuju (vidi primjer slike ispod). Funkcija kao argumente prima veličinu kvadrata u pikselima, broj kvadrata po visini i broj kvadrata po širini slike. Za realizaciju ove funkcije koristite `numpy` funkcije `zeros` i `ones` kako biste kreirali crna i bijela polja. Kako bise ih složili u odgovarajući oblik koristite `numpy` funkcije `hstack` i `vstack`. Za prikaz *grayscale* slike koristite naredbu:

```
plt.imshow(img, cmap='gray', vmin=0, vmax=255)
```



V. Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom PSU_LV.