

## VJEŽBA 2: RAD S DODATNIM PYTHON BIBLIOTEKAMA.

**I. Cilj 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 unaprijed prevedenih 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 i liste su efikasno (brzo) matrično računanje te efikasna implementacija polja (engl. *array*). Osnovni objekt Numpy biblioteke je je višedimenzionalno polje *ndarray* za pohranu podataka koji su istog tipa (uobičajeno cjelobrojni brojevi ili brojevi s pomičnim zarezom). Koristi se za predstavljanje vektora (1D polje), matrica (2D polja) ili višedimenzionalnih polja. Naredbom `import numpy` učitava se numpy biblioteka. Tada se raspoloživim funkcijama pristupa na način `numpy.ime_funkcije`. Stoga se često zbog kraćeg zapisa koristi *alias*:

```
import numpy as np
```

pa se metodama pristupa na način `np.ime_funkcije`.

Numpy polje definira se pozivanjem funkcije `array`. Ovaj objekt tada sadrži sljedeće atribute:

- `ndim` – broj dimenzija.
- `shape` – dimenzija polja. Za matricu dimenzija  $N \times M$ , `shape` attribute je tuple  $(N, M)$ .
- `size` – ukupni broj elemenata u polju.
- `dtype` – tip elemenata u polju.

Elementi numpy polja indeksirani su pozitivnim cijelim brojevima. Iz polja je moguće izdvajati dijelove polja pomoću operatora `:` na isti način kao i kod Python lista. Nadalje, postoje funkcije za kreiranje polja koje sadrži nule (`zeros`), jedinice (`ones`), konstantu (`full`), jediničnu matricu (`eye`) i sl. Neki primjeri korištenja numpy biblioteke dani su u primjeru 2.1.

#### 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 sa svim elementima jednakim 0
d = np.ones((3,2))              #polje sa svim elementima jednakim 1
e = np.full((1,2),5)            #polje sa svim elementima jednakim 5
```

```
f = np.eye(2)                                #jedinična matrica 2x2

g = np.array([1, 2, 3], np.float32)
duljina = len(g)
print(duljina)
h = g.tolist()
print(h)
c = g.transpose()
print(g)
np.concatenate((a, g,))
```

Unutar Numpy biblioteke 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.argmax())
print(a.max())
print(a.argmax())
print(a.sum())
print(a.mean())

print(np.mean(a))
print(np.max(a))
print(np.sum(a))

a.sort()
print(a)
```

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

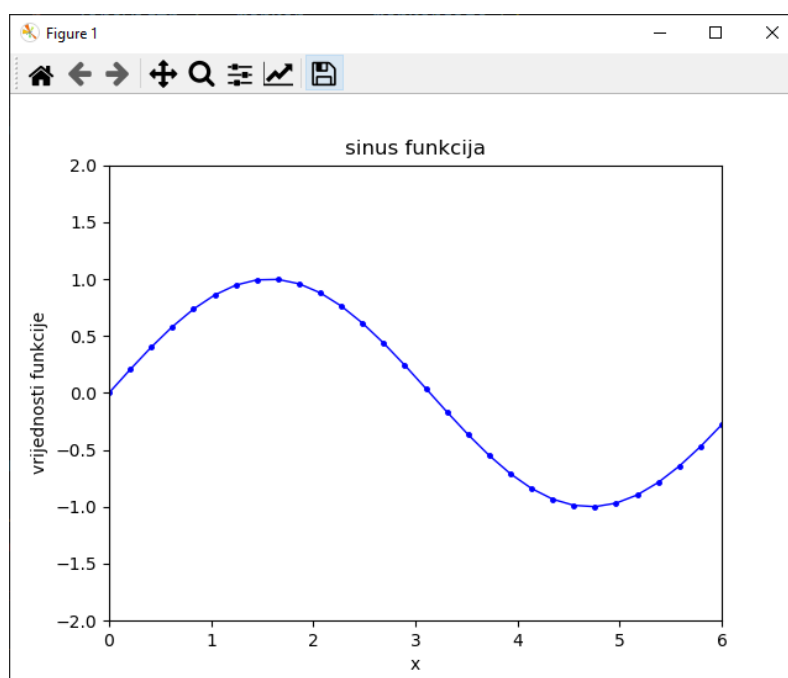
**Matplotlib** biblioteka sadrži različite funkcije za kreiranje statičkih i dinamičkih vizualizacija `matplotlib.pyplot` je sučelje koje omogućava izradu različitih prikaza. Ovo sučelje kreira sliku (engl. *figure*) koje se ujedno ponaša kao i grafičko sučelje. Primjer crtanja sinus i kosinus funkcije u rasponu od 0 do 6 dan je u primjer 2.4. Funkcija `.plt()` prima dva numpy polja (vrijednosti x i y osi) te spaja točke ravnim linijama. Dodatnim funkcijama moguće je označiti osi, dati naslov slici i sl.

**Primjer 2.4.**

#primjer vizualizacije

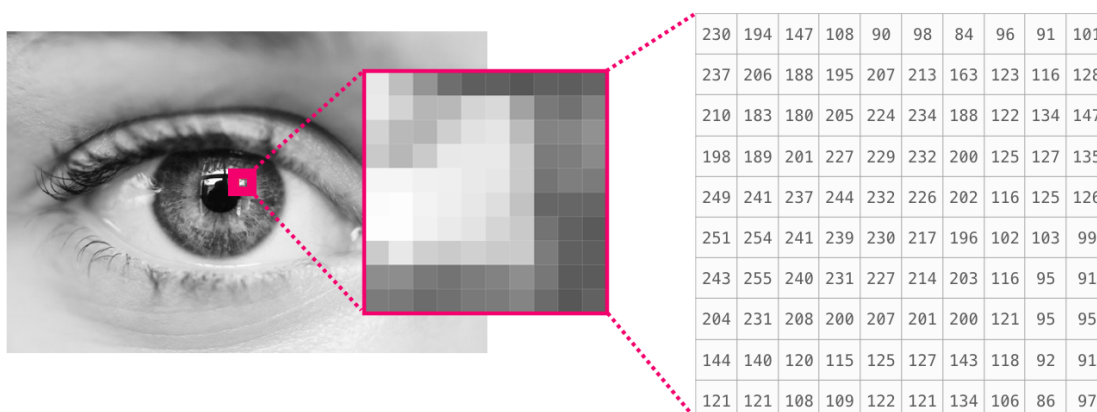
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 6, num=30)
y = np.sin(x)
plt.plot(x, y, 'b', linewidth=1, marker=".", markersize=5)
plt.axis([0,6,-2,2])
plt.xlabel('x')
plt.ylabel('vrijednosti funkcije')
plt.title('sinus funkcija')
plt.show()
```



Sl. 2.1. matplotlib.pyplot slika

Digitalna slika sastoji se od osnovnih elemenata koji se nazivaju pikseli. Digitalna slika se u Pythonu predstavlja numpy poljem pri čemu svaki element numpy polja odgovara jednom pikselu na slici. Vrijednosti elemenata se kreću od 0 (bijela boja) do 255 (crna boja) kao što je prikazano na slici 2.2. Prilikom učitavanja slike neke biblioteke skaliraju vrijednosti piksela na raspon od 0-1 (kao npr. matplotlib).



Sl. 2.2. Primjer slike u sivim tonovima i način predstavljanja slike kao 2D polja.

Na sličan način se u memoriju računala pohranjuju slike u boji. Najčešći način zapisivanja slike u boji je u RGB formatu. Tada je slika u memoriji prikazana s tri kanala (matrice) pri čemu prva matrica predstavlja udio crvene, druga matrica udio zelene, a treća matrica udio plave boje za svaki piksel.

Učitavanje slika s medija za pohranu (npr. hardisk) radi se pomoću `matplotlib.pyplot.imread` funkcije. Potrebno je predati putanju do slike u obliku stringa, a funkcija vraća numpy polje koje predstavlja sliku. Slike se učitavaju u 3D dimenzionalno polje bez obzira radi li se o slici u boji ili slici u sivim tonovima. U slučaju slike u sivim tonovima, sva tri kanala će imati jednake vrijednosti pa je dovoljno koristiti samo jedan.

Primjer 2.5 prikazuje kako se učitava i prikazuje slika 'road.jpg' koja je slika u sivim tonovima. Iz dobivenog 3D polja odabire se samo prvi kanal. Slika je tada predstavljena 2D numpy poljem čiji `shape` je jednak rezoluciji slike. Učitana slika se može prikazati pomoću funkcije `matplotlib.pyplot.imshow` na način da joj se preda numpy polje koje odgovara slici.

#### Primjer 2.5.

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

img = plt.imread("road.jpg")
img = img[:, :, 0].copy()
print(img.shape)
print(img.dtype)

plt.figure()
plt.imshow(img, cmap="gray")
plt.show()
```

### III. Priprema za vježbu:

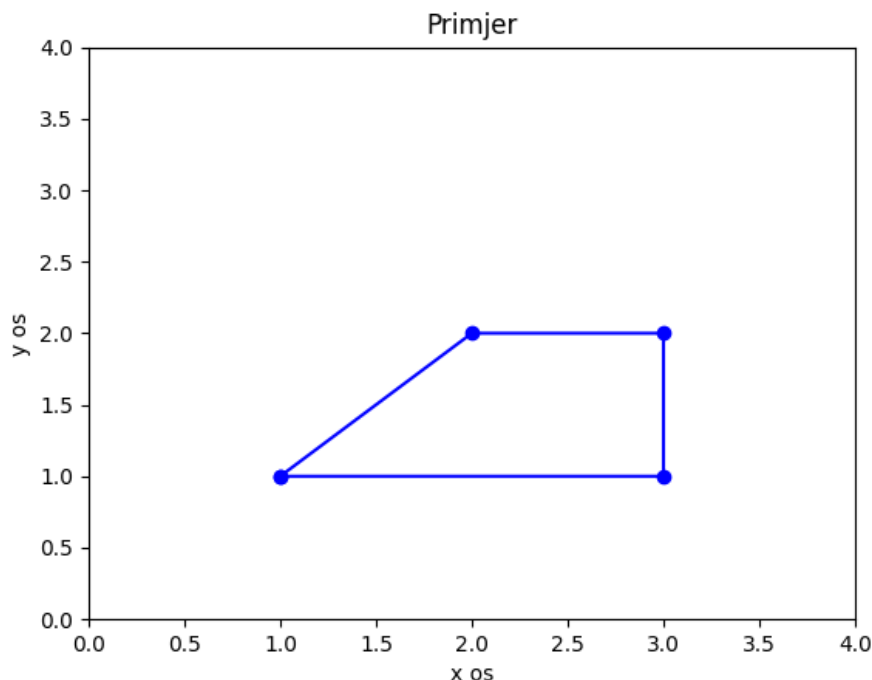
1. Proučite primjere 2.1. do 2.4. Možete li opisati što je rezultat izvođenja svake linije koda u navedenim primjerima?
2. Proučite kako se zapisuje slika u sivim tonovima u memoriju računala. Pogledajte primjer 2.5. koji učitava sliku pomoću biblioteke `matplotlib`.

### 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. Riješite dane zadatke.

### 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

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 3

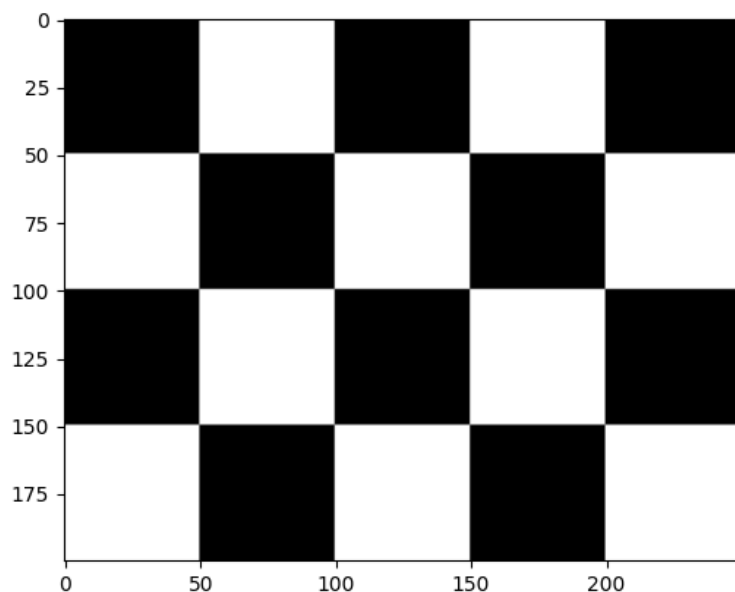
Na temelju primjera 2.5. učitajte sliku 'tiger.png'. Manipulacijom odgovarajuće numpy matrice pokušajte:

- posvijetliti sliku (povećati *brightness*),
- zarotirati sliku za 90 stupnjeva u smjeru kazaljke na satu,
- zrcaliti sliku,
- smanjiti rezoluciju slike x puta (npr. 10 puta),
- prikazati samo drugu četvrtinu slike po širini, a prikazati sliku cijelu po visini; ostali dijelovi slike trebaju biti crni.

**Zadatak 4**

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.