

**2025.08.24.**

**Órai Feladat**

**Határidő:**

**80 perc**

# Numpy és Pandas modulok gyakorlófeladatok

## Numpy

### 1. Mátrix létrehozása és műveletek:

Hozz létre egy 3x3-as mátrixot, és számítsd ki:

- A mátrix transzponáltját.
- Az egyes sorok összegét és átlagát.
- A mátrix determinánsát és inverzét (ha létezik).

### 2. Vektor műveletek:

Generálj két 1x5-ös vektort véletlenszámokkal (0-100 között). Számítsd ki:

- A két vektor skaláris szorzatát.
- Az elemenkénti különbséget és szorzatot.

## Feladatok Numpy, Pandas és beépített függvényekhez

### Numpy feladatok

#### 1. Mátrix manipuláció:

Hozz létre egy 4x4-es mátrixot véletlenszámokkal (1-100 között), és számítsd ki:

- A főátló elemeinek összegét.
- A mátrix transzponáltját.

#### 2. Vektorok műveletei:

Generálj két 1x6-os vektort véletlenszámokkal (0-50 között). Számítsd ki:

- A vektorok összegét és különbségét.
- Az elemenkénti szorzatukat.

#### 3. Adathalmaz statisztikája:

Hozz létre egy 100 elemű véletlenszámokból álló tömböt (1-100 között), és számítsd ki:

- Az átlagot, mediánt, szórást, maximumot és minimumot.

#### 4. Maszkolás:

Hozz létre egy tömböt, és állítsd be az összes páratlan számot 0-ra.

#### 5. Két mátrix szorzása:

Hozz létre két 3x3-as mátrixot, és számítsd ki a mátrixszorzatukat.

#### 6. Feltételes módosítás:

Generálj egy 5x5-ös mátrixot, és cseréld ki az összes 50 feletti értéket 100-ra.

#### 7. Sinus- és koszinus görbe:

Hozz létre egy tömböt 0-tól  $2\pi$ -ig 0.1-es lépésközzel, és számítsd ki a szinusz és koszinusz értékeket. Ábrázold őket matplotlib segítségével.

#### 8. Sorok és oszlopok átlagának kiszámítása:

Hozz létre egy 6x6-os mátrixot, és számítsd ki az egyes sorok és oszlopok átlagát.

## Pandas feladatok

### 1. DataFrame létrehozása:

Készíts egy DataFrame-et a következő adatokkal: név, életkor, pontszám. Számítsd ki az átlagpontszámot, és szűrd le azokat, akik pontszáma nagyobb az átlagnál.

### 2. CSV beolvasása és elemzése:

Olvass be egy `csv` fájlt Pandas segítségével, és:

- Számold meg, hány sor van benne.
- Határozd meg az egyes oszlopok maximumát, minimumát és átlagát.

### 3. Csoportosítás és aggregálás:

Egy DataFrame-ben csoportosítsd az adatokat kategóriák szerint, és számítsd ki az egyes csoportok átlagát.

### 4. Hiányzó adatok kezelése:

Hozz létre egy DataFrame-et, amely tartalmaz néhány hiányzó értéket. Töltsd ki ezeket az oszlop átlagával vagy mediánjával.

### 5. Idősoros adatok elemzése:

Generálj egy idősor adatsort a mai dátumtól kezdve 10 napra előre. Adj hozzá véletlenszerű értékeket, és számítsd ki a mozgó átlagot.

### 6. Feltételes szűrés:

Szűrd le azokat a sorokat egy DataFrame-ben, ahol egy adott oszlop értéke meghalad egy adott küszöböt.

### 7. Adatok módosítása:

Egy DataFrame adott oszlopának minden értékét szorozd meg 2-vel.

### 8. Adatok rendezése:

Rendezd a DataFrame-et egy adott oszlop értékei szerint csökkenő sorrendben.

## Beépített függvények feladatok

1. Számítsd ki az abszolút értékeket egy listában lévő számokra az `abs` használatával!
2. Hozz létre egy listát, és számítsd ki az összegét az `sum` használatával!
3. Keresd meg a lista legkisebb és legnagyobb értékét a `min` és `max` függvényekkel!
4. Írj egy függvényt, ami egy számot adott tizedesjegyre kerekít az `round` használatával!
5. Készíts egy programot, ami megadja egy szám négyzetét a `pow` segítségével!
6. Ellenőrizd, hogy egy lista összes eleme igaz-e az `all` használatával!
7. Ellenőrizd, hogy egy lista legalább egy igaz elemet tartalmaz-e az `any` segítségével!
8. Rendezd a listát növekvő sorrendbe a `sorted` használatával!
9. Írj egy programot, ami megadja egy lista hosszát a `len` használatával!
10. Vizsgáld meg az objektumok típusát a `type` függvénnyel!

## List és Dictionary Comprehension feladatok

1. Hozz létre egy listát, amely tartalmazza az 1-20 közötti számok négyzeteit list comprehension segítségével!
2. Hozz létre egy listát, amely tartalmazza az 1-50 közötti páros számokat!
3. Szűrd ki egy listából azokat az értékeket, amelyek nem oszthatók 3-mal!
4. Hozz létre egy dictionary comprehension-t, ami az 1-10 közötti számok négyzetét tartalmazza!
5. Alakítsd át egy lista szöveges elemeit nagybetűssé list comprehension segítségével!
6. Szűrd ki egy dictionary-ből azokat az elemeket, ahol az érték nagyobb, mint 10!

## Linkek

<https://numpy.org/doc/stable/>  
<https://pandas.pydata.org/docs/>  
<https://matplotlib.org/stable/users/index.html>