

## ▼ Pandas és Folium gyakorlófeladatok

Ez a notebook tartalmazza a Pandas és Folium gyakorlására szolgáló feladatokat. A feladatokat adatfájlokkal és megoldásokkal együtt készítjük el.

### Adatok

Az adatokat a `cities.csv` fájl tartalmazza, amely Magyarország városaira vonatkozó földrajzi és demográfiai adatokat tartalmaz.

## ▼ Adatok létrehozása

Generáljuk le az adatokhoz szükséges `cities.csv` fájlt.

```
import pandas as pd

# Adatok létrehozása
data = {
    "city": [
        "Budapest", "Debrecen", "Szeged", "Pécs", "Győr",
        "Miskolc", "Nyíregyháza", "Kecskemét", "Székesfehérvár",
        "Eger", "Veszprém", "Tatabánya", "Salgótarján", "Kaposvár"
    ],
    "latitude": [
        47.4979, 47.5316, 46.253, 46.0727, 47.6875,
        48.1031, 47.9495, 46.8964, 47.186,
        47.9025, 47.0928, 47.5883, 48.1042, 46.3597
    ],
    "longitude": [
        19.0402, 21.6273, 20.1414, 18.2323, 17.6504,
        20.7784, 21.7247, 19.6897, 18.4221,
        20.3772, 17.9093, 18.3973, 19.8033, 17.7952
    ],
    "population": [
        1752286, 202402, 160766, 145347, 131267,
        151025, 119746, 110687, 96170,
        53696, 58763, 66124, 32133, 60300
    ],
    "area": [
        525, 461, 280, 162, 174,
        236, 274, 321, 170,
        92, 126, 91, 102, 113
    ]
}

# DataFrame létrehozása
df = pd.DataFrame(data)

# CSV mentése
df.to_csv("cities.csv", index=False)

print("Az adatfájl létrejött: 'cities.csv'")
```

## ▼ 1. feladat: Adatok beolvasása és elemzése

```
import pandas as pd

# CSV fájl beolvasása
df = pd.read_csv('cities.csv')

# Adatok megtekintése
print(df.head()) # Az első 5 sor
print(df.info()) # Oszloptípusok és hiányzó értékek
print("Egyedi városok száma:", df['city'].nunique())
```

## ▼ 2. feladat: Adatok szűrése és aggregálása

```
# Városok szűrése, ahol a népesség > 1 millió
filtered_df = df[df['population'] < 1_000_000]
print(filtered_df)

dataf = filtered_df.query('population > 1_00_000 and population < 2_000_000')
print('-----')
print('Dataf: ', dataf.head(5))
print('-----')

# Átlagos népesség számítása
average_population = filtered_df['population'].mean()
print("Átlagos népesség:", average_population)
print('Head')
dataf.head(5)
```

### 3. feladat: Városok térképen Folium segítségével

```
import folium

# Alaptérkép létrehozása
m = folium.Map(location=[47.1625, 19.5033], zoom_start=6)
# Magyarország középpontja

# Jelölők hozzáadása nagy népességű városokhoz
for _, row in filtered_df.iterrows():
    folium.Marker(
        location=[row['latitude'], row['longitude']],
        popup=f"{row['city']} - Népesség: {row['population']}",
    ).add_to(m)

# Térkép mentése
m.save('cities_map.html')
print("Térkép mentve: 'cities_map.html'")
m
```

### 4. feladat: Klaszterezett térkép készítése

```
from folium.plugins import MarkerCluster

# Klaszteres térkép létrehozása
marker_cluster = MarkerCluster().add_to(m)

for _, row in df.iterrows():
    folium.Marker(
        location=[row['latitude'], row['longitude']],
        popup=f"{row['city']} - Népesség: {row['population']}",
    ).add_to(marker_cluster)

# Klaszteres térkép mentése
m.save('clustered_map.html')
print("Klaszteres térkép mentve: 'clustered_map.html'")
m
```

### 5. feladat: Hőtérkép készítése

```
from folium.plugins import HeatMap

# Hőtérkép adatok előkészítése
heat_data = [[row['latitude'], row['longitude'], row['population']] for _, row in df.iterrows()]

# Hőtérkép létrehozása
HeatMap(heat_data).add_to(m)

# Hőtérkép mentése
m.save('heatmap.html')
print("Hőtérkép mentve: 'heatmap.html'")
m
```

