# data\_processing.py Modul Működése

## Összefoglaló

A data\_processing.py modul a Dash alkalmazás adatkezelési részéért felel.   
Fő célja a nyers adatok betöltése, tisztítása, hiányzó értékek kezelése, és az adatok előkészítése vizualizációra.   
A funkciók biztosítják, hogy a bemeneti CSV-fájl adatai konzisztens és megfelelő formában kerüljenek további feldolgozásra.  
  
Fő lépések:  
1. Adatok beolvasása egy CSV-fájlból.  
2. Oszlopnevek tisztítása és átnevezése.  
3. Az évszámokat azonosítja és új oszlopba helyezi.  
4. Csak a releváns sorokat tartja meg (bruttó és nettó jövedelem típusok).  
5. A numerikus oszlopok értékeit tisztítja és átalakítja.  
6. Hiányzó éveket automatikusan pótolja, hogy a vizualizáció folytonos legyen.  
7. Az eredmény egy Pandas DataFrame, amely tisztított és előkészített adatokat tartalmaz.  
  
Ez a modul biztosítja az alkalmazás stabil működéséhez szükséges tiszta adatokat.

## Részletes Magyarázat

A data\_processing.py modul kódjának részletes magyarázata:  
  
1. \*\*Adatok betöltése\*\*:  
```python  
data = pd.read\_csv(file\_path, encoding='utf-8', sep=';', skiprows=1)  
```  
Betölti az adatokat egy CSV-fájlból a megadott útvonalról Pandas DataFrame-ként. A pontosvessző (`;`) határozza meg az oszlopelválasztót.  
  
2. \*\*Oszlopnevek tisztítása és átnevezése\*\*:  
```python  
data.columns = [col.strip() for col in data.columns]  
data.rename(columns={'Megnevezés': 'Jövedelem\_típus', 'Ország összesen': 'Ország összesen'}, inplace=True)  
```  
Az oszlopnevekből eltávolítja a felesleges szóközöket, majd átnevezi őket beszédesebb nevekre, például "Jövedelem\_típus".  
  
3. \*\*Évszámok felismerése és kezelése\*\*:  
```python  
year\_rows = data.iloc[:, 0].str.isdigit()  
data['Év'] = None  
data.loc[year\_rows, 'Év'] = data.loc[year\_rows, data.columns[0]]  
data['Év'] = data['Év'].ffill()  
```  
Az évszámokat az első oszlopban azonosítja, és egy új "Év" oszlopba helyezi. A hiányzó évszámokat az előző év értékével tölti ki.  
  
4. \*\*Releváns sorok szűrése\*\*:  
```python  
data = data[~year\_rows]  
data = data[data['Jövedelem\_típus'].isin(['Bruttó jövedelem', 'Nettó jövedelem'])]  
```  
Eltávolítja az évszámokat tartalmazó sorokat, és csak a bruttó és nettó jövedelemtípusokat tartja meg.  
  
5. \*\*Numerikus adatok konvertálása\*\*:  
```python  
numeric\_columns = [col for col in data.columns if col not in ['Jövedelem\_típus', 'Év']]  
for col in numeric\_columns:  
 data[col] = data[col].replace({' ': '', '…': '0'}, regex=True).str.replace(',', '').astype(float)  
```  
Az oszlopokban lévő numerikus értékeket megtisztítja (eltávolítja a vesszőket és helyettesíti a speciális karaktereket), majd lebegőpontos számokká alakítja.  
  
6. \*\*Hiányzó évek pótlása\*\*:  
```python  
all\_years = pd.Series(range(int(data['Év'].min()), int(data['Év'].max()) + 1))  
missing\_years = all\_years[~all\_years.isin(data['Év'])]  
```  
Az évszámok teljes tartományát azonosítja, és megkeresi a hiányzó éveket.  
  
Hiányzó sorokat generál minden hiányzó évhez:  
```python  
empty\_rows = []  
for year in missing\_years:  
 for income\_type in ['Bruttó jövedelem', 'Nettó jövedelem']:  
 empty\_rows.append({'Jövedelem\_típus': income\_type, 'Év': year})  
```  
Majd ezeket hozzáfűzi az adatokhoz:  
```python  
empty\_df = pd.DataFrame(empty\_rows)  
data = pd.concat([data, empty\_df], ignore\_index=True)  
data = data.sort\_values(by='Év').reset\_index(drop=True)  
```  
  
7. \*\*Adatok visszaadása\*\*:  
```python  
return data  
```  
A feldolgozott adatokat Pandas DataFrame formájában adja vissza, amelyek teljesek, tisztítottak és rendezettek.  
  
Ez a kód kulcsfontosságú az alkalmazás adatvizualizációjához, mert biztosítja az adatok minőségét és konzisztenciáját.