

Ez egy komment rész. Ide tudunk esetleg jegyzetelni, ha nem szeretnénk túlkommentálni a kódblokkokat.

Markdown kompatibilis!

A gyors használatot segítő fontosabb billentyűzet kombinációk:

✓ Jupyter Lab környezetben

- **Ctrl + Enter** - jelenlegi cella futtatása
- **Shift + Enter** - jelenlegi cella futtatása, majd továbblépés
- **Esc** - kilépés a szerkesztő módból (így könnyebb lépkedni a cellák között)
- **Enter** - belépés a szerkesztő módba
- **2x D** - cella törlése (szerkesztő módon kívül)
- **A** - cella hozzáadása felülre (szerkesztő módon kívül)
- **B** - cella hozzáadása alulra (szerkesztő módon kívül)
- **Ctrl + /** : kommentelés

Google Colab környezetben

- Hasonló módon, mint Jupyter Lab-ban, csak az egybetűs műveletekhez **Ctrl + M**-et kell nyomni
 - pl. Törlés: **Ctrl + M + D**
 - pl. Új cella a kijelölt fölé: **Ctrl + M + A**

```
!pip install numpy
!pip install matplotlib
!pip install pandas
```

```
#Ezekkel a meghívásokkal tudunk kernelszintű script-utasításokat kiadni (aza
# akkor a ! jelzéssel könnyen letölthető a hiányzó csomag (a cella futtatása
```

✓ Fontos!

A cellákat mindig futtatni kell, hogy a bennük lévő tartalom több cellában is felhasználható legyen.

```
import pandas as pd          ## Adatelemzésre, adatmanipulációra hasz
import numpy as np           ## Numerikus műveletekhez, nagyobb mátri
import matplotlib.pyplot as plt ## Adatvizualizációs könyvtár - adatokat
```

```
#Olvassuk be a shipping_data.csv fájlt!
```

```
shipping_data = pd.read_csv("shipping_data.csv", index_col = 0) #Első oszlop  
shipping_data
```

```
#Számoljuk meg hány null érték van a táblázatban!  
shipping_data.isnull().sum()
```

```
#Mutassuk ki az adatok statisztikáit! (dtypes, describe, info)  
#print(shipping_data.info)  
print(shipping_data.dtypes)  
print(shipping_data.head())  
shipping_data.describe()
```

```
#Probléma: ne legyen tört szám a mennyiség! Nincs 5.5 darab könyv, stb.  
shipping_data["Quantity"] = shipping_data["Quantity"].fillna(1)  
shipping_data["Quantity"] = shipping_data["Quantity"].astype(np.int64)  
shipping_data["OrderDate"] = pd.to_datetime(shipping_data["OrderDate"])
```

```
#Alakítsuk át a Shipped oszlopot logikai értékké!  
#shipping_data["Shipped"] = (shipping_data["Shipped"] == "Yes")  
shipping_data["Shipped"] = shipping_data["Shipped"].apply(lambda value: valu  
shipping_data
```

```
print(shipping_data.dtypes)  
#shipping_data
```

```
#Dobjuk el azokat az értékeket, ahol szerepel valamelyik adattagban!  
shipping_data = shipping_data.dropna()
```

```
shipping_data
```

```
#Számoljuk meg egy Revenue oszlopban, hogy mennyi az összes bevétel az adott  
shipping_data["Revenue"] = shipping_data["Quantity"] * shipping_data["UnitPr  
shipping_data
```

```
# Csináljunk egy books változót, amiben eltároljuk azokat az adatokat, amely  
# (Figyeljünk a helyes feltételekre!)  
books = shipping_data[(shipping_data["Category"] == "Books") & (shipping_dat  
books
```

```
# Kérdezzük le, hogy mik az áprilisi rendelések!  
april_orders = shipping_data[shipping_data["OrderDate"].dt.month == 4]  
april_orders
```

```
#Adjuk meg, hogy mennyi volt a márciusban kiszállított termékek összértéke!  
march_shipped = shipping_data[(shipping_data["OrderDate"].dt.month == 3) & (
```

```
print(march_shipped["Revenue"].sum(), "$")
```

```
#Adjuk meg, hogy mennyi volt a bevételünk napi lebontásban!  
shipped_by_days = shipping_data.groupby(by=shipping_data["OrderDate"])[  
    "Revenue"]  
shipped_by_days #Series lesz!  
#type(shipped_by_days)
```

```
#Rajzoljuk ki a napi bevételeket grafikonnal!  
plt.figure(figsize=(18,7))  
plt.title("Napi bevételek a rendelésekből")  
plt.xlabel("Dátumok")  
plt.ylabel("Bevétel")  
plt.xticks(rotation=45)  
plt.bar(shipped_by_days.index, shipped_by_days.values, color="green")  
plt.plot()
```

Kezdjen el kódolni, vagy [generáljon](#) kódot az AI-jal.