

### 1. Pandas importieren

```
import pandas as pd
```

### 2. Datenstrukturen

#### DataFrame

2D-Tabelle (wie eine Excel-Tabelle)

#### Series

1D-Array (eine Spalte oder Zeile)

### 3. Daten einlesen

#### CSV-Datei einlesen

```
df = pd.read_csv('datei.csv')
```

#### Excel-Datei einlesen

```
df = pd.read_excel('datei.xlsx')
```

#### JSON-Datei einlesen

```
df = pd.read_json('datei.json')
```

#### Aus einer Liste erstellen

```
df = pd.DataFrame([['Alice', 25], ['Bob', 30]], columns=['Name', 'Alter'])
```

### 4. Daten anzeigen

#### Erste 5 Zeilen anzeigen

```
df.head()
```

#### Letzte 5 Zeilen anzeigen

```
df.tail()
```

#### Zufällige Zeilen anzeigen

```
df.sample(3)
```

#### Informationen über den DataFrame (Spalten, Datentypen, etc.)

```
df.info()
```

#### Beschreibende Statistiken (nur numerische Spalten)

```
df.describe()
```

### 5. Daten auswählen

#### Eine Spalte auswählen (als Series)

```
df['Spaltenname']
```

#### Mehrere Spalten auswählen

```
df[['Spalte1', 'Spalte2']]
```

#### Zeilen nach Index auswählen

```
df.iloc[0] # Erste Zeile
df.iloc[0:5] # Zeilen 0 bis 4
```

#### Zeilen nach Bedingung filtern

```
df[df['Alter'] > 30]
```

#### Werte in einer Spalte filtern

```
df[df['Spalte'].isin(['Wert1', 'Wert2'])]
```

### 6. Daten bearbeiten

#### Neue Spalte hinzufügen

```
df['NeueSpalte'] = df['Spalte1'] + df['Spalte2']
```

#### Spalte umbenennen

```
df.rename(columns={'Alter': 'Age'}, inplace=True)
```

#### Spalte löschen

```
df.drop('Spaltenname', axis=1, inplace=True)
```

#### Zeilen löschen

```
df.drop([0, 1], axis=0, inplace=True) # Löscht die ersten beiden Zeilen
```

#### Fehlende Werte behandeln

```
df.dropna() # Zeilen mit fehlenden Werten löschen
df.fillna(0) # Fehlende Werte mit 0 ersetzen
```

### 7. Dateien sortieren

#### Nach einer Spalte sortieren

```
df.sort_values('Spaltenname', ascending=False)
```

#### Nach mehreren Spalten sortieren

```
df.sort_values(['Spalte1', 'Spalte2'], ascending=[True, False])
```

### 8. Gruppieren und aggregieren

#### Gruppieren und Aggregieren

```
df.groupby('Spalte')['AndereSpalte'].mean()
```

#### Mehrere Aggregationen

```
df.groupby('Spalte').agg({'Spalte1': 'sum', 'Spalte2': 'mean'})
```

### 9. Daten speichern

#### Als CSV speichern

```
df.to_csv('neue_datei.csv', index=False)
```

#### Als Excel speichern

```
df.to_excel('neue_datei.xlsx', index=False)
```

#### Als JSON speichern

```
df.to_json('neue_datei.json')
```

### 10. Nützliche Funktionen

#### Duplikate entfernen

```
df.drop_duplicates()
```

#### Spalten in einen bestimmten Datentyp umwandeln

```
df['Spalte'] = df['Spalte'].astype(int)
```

#### Zeilen- und Spaltenanzahl

```
df.shape
```

#### Eindeutige Werte in einer Spalte

```
df['Spalte'].unique()
```

#### Anzahl der eindeutigen Werte

```
df['Spalte'].nunique()
```

### 11. Daten kombinieren

#### Vertikal kombinieren (Zeilen hinzufügen)

```
pd.concat([df1, df2], axis=0)
```

#### Horizontal kombinieren (Spalten hinzufügen)

```
pd.concat([df1, df2], axis=1)
```

#### Merge (SQL-ähnlicher Join)

```
pd.merge(df1, df2, on='GemeinsameSpalte', how='inner')
```

### 12. Zeitreihen

#### Spalte in Datum umwandeln

```
df['Datum'] = pd.to_datetime(df['Datum'])
```

#### Nach Datum filtern

```
df[df['Datum'] > '2023-01-01']
```

#### Zeitreihen-Gruppierung

```
df.set_index('Datum').resample('M').mean() # Monatliche Mittelwerte
```



By **NeuesRos**  
[cheatography.com/neuesros/](https://cheatography.com/neuesros/)

Not published yet.  
Last updated 26th January, 2025.  
Page 2 of 2.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>