

### Installation

```
conda install pandas
conda install seaborn
conda install pandas seaborn
```

### Export

```
CSV: df.to_csv(filename)
Excel: df.to_excel(filename)
SQL: df.to_sql(table_name, connection_object)
JSON: df.to_json(filename)
```

### Spalten bearbeiten

```
Spalte Hinzufügen:
df['fuenf'] = 5
df['neue Spalte'] = 2*df['alte Spalte']

Spalte löschen:
del df['fuenf']

Spalten umbenennen:
df.rename(columns={'alt1':'neu1','alt2':'neu2'}, inplace = True)
df.columns = ['neu1', 'neu2', 'neu3', 'neu4']

Rename: Funktionen entgegennehmen
df.rename(columns= str.upper, inplace = True)
df.rename(columns=lambda x: f'T1_{x}', inplace = True)
```

### Broadcasting

```
s1 = pd.Series(rng.standard_normal(10)) - 5
```

### Ausprägungen

```
Index:
df.idxmin()
df.idxmax()

Max wer zeile :
df.loc[df['Wert1'].idxmax()]

Position:
argmin()
argmax()
```

### Import

```
CSV: pd.read_csv(filename)
Excel: pd.read_excel(filename)
SQL : pd.read_sql(query, connection_object)
JSON: pd.read_json(json_string)
HTML: pd.read_html(url)
Clipboard: pd.read_clipboard()
dict: pd.DataFrame(dict)
```

### Dataframes

```
aus Listen: df = pd.DataFrame(["rot","grün","blau","gelb"], columns=["Farbe"])
aus Dictionaries: d = {"Name": ['Katja', 'Nina', 'Sven', 'Matthias'], "Alter": [32, 32, 36, 31], "Ort": ['Berlin', 'München', 'Frankfurt', 'Köln']}
df = pd.DataFrame(d)

seaborn: import seaborn as sns
```

### Zeilen bearbeiten

```
Zeilen anhand des Index löschen:
df = df.drop(1)
df = df.drop([0,1,2,3])
df.drop('fuenf', axis=1, inplace=True)

Werte anhand von boolescher Bedingung ändern:
df.loc[df["Alter"]>=32,'Ort'] = 'Berlin'
```

### Sortieren

```
Nach index:
df.sort_index()

Nach spaltenwerten
df.sort_values(by='Wert2')
df.sort_values(by=['Wert2','Wert3'], ascending=[False, True])
```

### Series

```
aus Listen: s = pd.Series(["rot","grün","blau","gelb"])
aus Dictionaries: d = {'Berlin':3645000, 'München':1472000, 'Hamburg':1841000, 'Köln':1086000 } staedte = pd.Series(d)

elementen zugriff: wie eine Liste. z.b: zahlen[3]

Bei einem dictionary werden die Schlüssel zum Index: staedte["Hamburg"]

NaN: Not a Number = fehlende Wert
```

### Dataframes Zugriff:

```
Erste zeile: df.head()

Spalten (über Spaltennamen) :
df.columns
df["Alter"] oder df.Alter
df[["Name","Alter"]]

Zeilen und Spalten per iloc
df.iloc[0]
df.iloc[0,0]
df.iloc[0,0:2]

über loc per Index-Label:
df.loc[:, "Alter"]
df.index = df["Name"]
df.loc[["Nina", "Sven"], "Ort"]

booleans:
df.loc[df["Alter"]>=32]
df.loc[df["Name"].str[:1]=="M"]
df.loc[df["Alter"].isnull()]

Werten Zuweisung
df.loc['Sven:', 'Alter'] = 30
```



By omarffkamel

[cheatography.com/omarffkamel/](https://cheatography.com/omarffkamel/)

Published 6th July, 2022.  
Last updated 6th July, 2022.  
Page 1 of 1.

Sponsored by [Readable.com](https://readable.com)  
Measure your website readability!  
<https://readable.com>