

Das ultimative Pandas Cheatsheet für Excel-Umsteiger

Du weißt, wie man Daten in Excel handhabt? Perfekt! Hier sind die exakten Pandas-Befehle, um die gleichen Aufgaben in Python zu erledigen – nur schneller und automatisierbar.

1. Daten einlesen (Die Datei öffnen)

Der erste Schritt ist immer, deine Daten in Python zu laden. Pandas macht das mit einem einzigen Befehl.

Excel-Datei einlesen:

```
import pandas as pd
df = pd.read_excel('deine_datei.xlsx')
```

CSV-Datei einlesen: (Achte auf das Trennzeichen, oft ist es ein Semikolon ;)

```
import pandas as pd
df = pd.read_csv('deine_datei.csv', delimiter=';')
```

2. Erster Überblick (Die Tabelle ansehen)

Verschaffe dir schnell einen Überblick über deine Daten, genau wie du die ersten Zeilen in Excel ansehen würdest.

Die ersten 5 Zeilen anzeigen:

```
df.head()
```

Technische Zusammenfassung (Datentypen, leere Zellen):

```
df.info()
```

Statistische Zusammenfassung (Mittelwert, Min/Max etc.):

```
df.describe()
```

3. Spalten auswählen & umbenennen

Greife auf einzelne Spalten zu oder gib ihnen sprechendere Namen.

Eine einzelne Spalte auswählen:

```
umsatz_spalte = df['Umsatz']
```

Mehrere Spalten auswählen:

```
auswahl = df[['Kunde', 'Umsatz']]
```

Eine Spalte umbenennen:

```
df = df.rename(columns={'Alter Wert': 'Neuer Wert'})
```

4. Zeilen filtern (Der "AutoFilter" von Excel)

Das ist eine der mächtigsten Funktionen. Wähle nur die Zeilen aus, die bestimmte Kriterien erfüllen.

Alle Umsätze größer als 1000:

```
grosse_umsaetze = df[df['Umsatz'] > 1000]
```

Nur Kunden aus Deutschland:

```
deutsche_kunden = df[df['Land'] == 'DE']
```

5. Daten sortieren

Sortiere deine gesamte Tabelle basierend auf den Werten in einer oder mehreren Spalten.

Nach Umsatz absteigend sortieren:

```
df_sortiert = df.sort_values(by='Umsatz', ascending=False)
```

Die 5 wichtigsten "Pivot-Tabellen"-Befehle

6. Gruppieren & Zählen (Wie ZÄHLENWENN)

Zähle, wie oft jeder Eintrag in einer Kategorie vorkommt.

Anzahl der Bestellungen pro Land:

```
bestellungen_pro_land = df.groupby('Land').size()
```

7. Gruppieren & Summieren (Wie SUMMEWENN)

Die klassische Pivot-Analyse: Bilde Summen für verschiedene Kategorien.

Gesamtumsatz pro Land:

```
umsatz_pro_land = df.groupby('Land')['Umsatz'].sum()
```

8. Neue Spalten berechnen

Erstelle neue Spalten basierend auf den Werten in bestehenden Spalten.

Mehrwertsteuer (19%) berechnen:

```
df['MwSt'] = df['Nettoumsatz'] * 0.19
```

Bruttoumsatz berechnen:

```
df['Bruttoumsatz'] = df['Nettoumsatz'] + df['MwSt']
```

9. Duplikate finden & entfernen

Halte deine Daten sauber, indem du doppelte Einträge identifizierst und entfernst.

Alle doppelten Zeilen finden:

```
duplikate = df[df.duplicated()]
```

Doppelte Zeilen entfernen und eine saubere Tabelle behalten:

```
df_sauber = df.drop_duplicates()
```

10. Ergebnis exportieren (Speichern unter...)

Speichere deine fertige Analyse als neue, saubere Excel-Datei.

Als Excel-Datei speichern (ohne den Index):

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