

# Cheat Sheet

	<b><u>Daten importieren/exportieren</u></b>	
1	<code>uploaded = files.upload()</code>	Daten in pandas dataframe importieren
2	<code>data = pd.read_excel(io.BytesIO(uploaded['Dateiname.xlsx']))</code>	
3	<code>df_gesamt.to_csv('df_gesamt.csv', encoding= 'utf-8-sig')</code>	Daten aus pandas dataframe exportieren
4	<code>files.download('df_gesamt.csv')</code>	
	<b><u>Basics</u></b>	
5	<code>data.head(5)</code>	Ersten 5 Zeilen des dataframes anzeigen
6	<code>len(data)</code>	Länge dataframe
7	<code>data['Spaltenname']</code>	Nur die Spalte 'Spaltenname' anzeigen
8	<code>df.columns = ['neuerName',neuerName2]</code>	Spalten neu benennen
9	<code>df = df.drop(index=0)</code>	Zeile löschen (0=erste Zeile)
10	<code>data = data.drop(['Spaltenname1', 'Spaltenname2'], axis =1)</code>	Spalten löschen
11	<code>df_leer = pd.DataFrame(columns=['Spaltenname1', 'Spaltenname2'])</code>	Leeren dataframe erstellen
12	<code>df = pd.concat([series1,series2],axis=1)</code>	Series oder dataframes zusammenfügen nebeneinander
13	<code>df = pd.concat([series1,series2],axis=0)</code>	Series oder dataframes zusammenfügen untereinander
14	<code>df = df.append({'Prüfstand':wert, 'maxTaktzeit':wert}, ignore_index=True)</code>	Neue Zeile hinzufügen
	<b><u>Filtern</u></b>	
15	<code>data.loc[data['Spaltenname']==wert]</code>	Nur Zeilen anzeigen, welche in der Spalte 'Spaltenname' den Wert wert besitzen
16	<code>data.loc[(data['Spaltenname']==wert)&amp;(data['Spaltenname2']==wert)]</code>	Das gleiche für zwei Spalten (& UND-verknüpfte Bedingung)
	<b><u>Gruppieren</u></b>	
17	<code>groupbyObj = data.groupby('Tierarten')</code>	Gruppieren nach Tierarten
18	<code>groupbyObj['Tierarten'].count()</code>	Zählen wie viele Tiere es pro Tierart gibt

# Cheat Sheet

19	<code>groupbyObj['AnzahlBeine'].sum()</code>	Aufsummieren der Beinanzahl pro Tierart (5 Elefanten a 4 Beine = Tierart Elefant 20 Beine)
20	<code>groupbyObj['AnzahlBeine'].apply(lambda s: s[s==6].count())</code>	Zählen wie viele Tiere 6 Beine haben pro Tierart
21	<code>groupbyObj.get_group('Affe')</code>	Zeige nur die Affen an
22	<b><u>Berechnungen mit Spalten/Zeilen eines Dataframes</u></b>  <code>df[neueSpalte] = df['bestSpalte']/df[bestSpalte2]</code>	Neue Spalte berechnet sich aus bestehenden
23	<code>df['Zeitdifferenz'] = df['Zeitstempel'].diff()</code>	Neue Spalte berechnet sich aus der Differenz zweier aufeinanderfolgenden Zeilen
24	<b><u>Durch Dataframe Zeile für Zeile durchiterrieren</u></b>  <code>for i, row in df.iterrows():     wertVonZeile = row['Spaltenname1']     df.at[i, 'Label'] = wertVonZeile</code>	i=Zeilenindex row=Zeile Zellenwert aus Spalte 'Spaltenname1' und der aktuellen Zeile wird ausgelesen und in Spalte 'Label' geschrieben
25	<b><u>Visualisierung</u></b>  <code>plt.plot(days, anzahlFehler)</code>	<code>plt.plot(x Werte, y Werte)</code>
26	<b><u>Kontingenztafel</u></b>  <code>data = [[500, 10], [487, 140]]</code>	
27	<code>import scipy.stats as stats stats.chi2_contingency(data)</code>	
28	<b><u>Machine Learning</u></b>  <code>from sklearn.tree import DecisionTreeClassifier from sklearn import tree dt = DecisionTreeClassifier().fit(dataOhneLabel, Label])</code>	Decision Tree trainieren

# Cheat Sheet

29	<pre>plt.figure(figsize = (24,13)) tree.plot_tree(dt, filled=True) plt.show()</pre>	Visualisierung des Trees												
30	<pre>predictions = dt.predict(dataOhneLabel)</pre>	Die Label werden durch Decision Tree vorhergesagt												
31	<pre>from sklearn.metrics import accuracy_score accuracy_score(predictions,Label)</pre>	Accuracy = Anzahl richtige Vorhersagen / Anzahl Vorhersagen												
32	<pre>from sklearn.metrics import confusion_matrix confusion_matrix(Label ,predictions)</pre>	<p>Konfusionsmatrix output</p> <table><thead><tr><th></th><th colspan="2">predicted</th></tr><tr><th></th><th>1</th><th>2</th></tr></thead><tbody><tr><td>1</td><td>11420</td><td>18</td></tr><tr><td>2</td><td>972</td><td>477</td></tr></tbody></table>		predicted			1	2	1	11420	18	2	972	477
	predicted													
	1	2												
1	11420	18												
2	972	477												