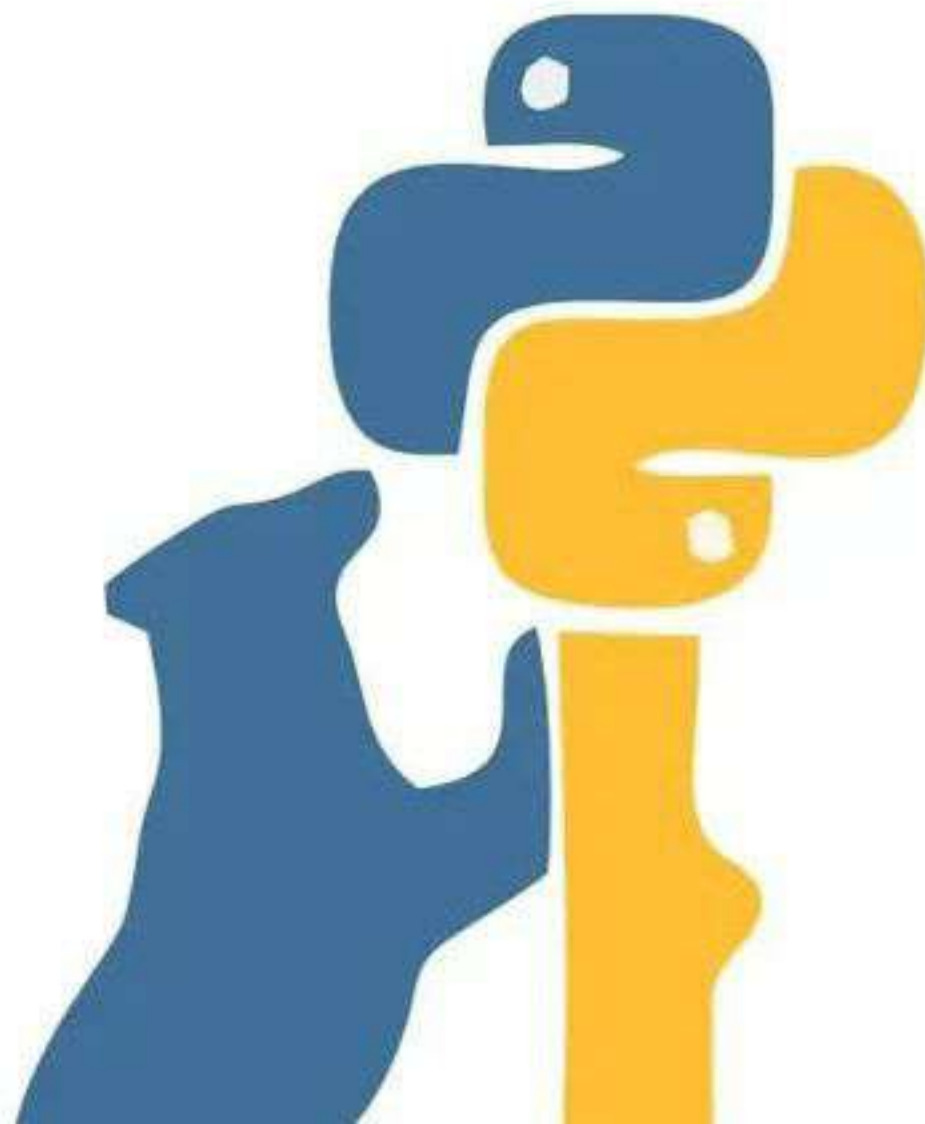# pandas

CHEATSHEET

Reading and Writing Data:

1. **Read data from a CSV file**

```python
df = pd.read_csv('file.csv')
```

2. **Write DataFrame to a CSV file**

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

3. **Read data from an Excel file**

```python
df = pd.read_excel('file.xlsx',
sheet_name='Sheet1')
```

4. **Write DataFrame to an Excel file**

```python
df.to_excel('output.xlsx',
sheet_name='Sheet1', index=False)
```

Exploring Data

```python
# Display the first n rows
df.head(n)
# Display the last n rows
df.tail(n)
# Summary statistics
df.describe()
# Info about DataFrame
df.info()
# Unique values in a column
df['column_name'].unique()
# Number of unique values in a column
df['column_name'].nunique()
# Count occurrences of each unique value
df['column_name'].value_counts()
```

Data Selection and Filtering

```python
# Selecting a single column
df['column_name']
# Selecting multiple columns
df[['column1', 'column2']]
# Filtering rows based on condition
df[df['column_name'] > 10]
# Multiple conditions
df[(df['column1'] > 10) & (df['column2'] ==
'value')]
# Using isin() for filtering
df[df['column_name'].isin(['value1', 'value2'])]
```

Handling Missing Data

```python
# Check for missing values
df.isnull()
# Drop rows with any missing values
df.dropna()
# Fill missing values with a specific value
df.fillna(value)
# Interpolate missing values
df.interpolate()
```

Grouping and Aggregation

```python
# Group by a column and calculate mean
df.groupby('column_name').mean()
# Group by multiple columns and calculate sum
df.groupby(['col1', 'col2']).sum()
# Aggregation with multiple functions
df.groupby('column_name').agg(['mean', 'sum'])
# Pivot table
pd.pivot_table(df, values='value_column',
index='index_column', columns='column_to_pivot')
```

Data Manipulation

```python
# Adding a new column
df['new_column'] = values

# Renaming columns
df.rename(columns={'old_name': 'new_name'},
inplace=True)

# Sorting by a column
df.sort_values(by='column_name', ascending=False)

# Drop a column
df.drop('column_name', axis=1, inplace=True)

# Concatenating DataFrames
new_df = pd.concat([df1, df2], axis=0)
```

Time Series Operations

```python
python

# Convert a column to datetime

df['date_column'] = pd.to_datetime(df['date_column']
# Set datetime column as index

df.set_index('date_column', inplace=True)
# Resampling time series data

df.resample('D').mean()   # Daily resampling
```