Pandas DataFrame Cheat Sheet

1. Creation & Basic Info

```
pd.DataFrame(data)    # Create DataFrame
df.head(n)            # First n rows
df.tail(n)            # Last n rows
df.shape              # (rows, columns)
df.size               # Total elements
df.ndim               # Dimensions
df.columns            # Column labels
df.index              # Index (row labels)
df.dtypes             # Data types of columns
df.values             # Numpy array of data
df.info()             # Summary of DataFrame
df.describe()         # Summary stats (numeric)
df.memory_usage()     # Memory usage
```

2. Data Selection & Filtering

```
df['col']                  # Single column (Series)
df[['col1', 'col2']]       # Multiple columns
df.loc[<row_label>]        # Label-based row access
df.iloc[<row_index>]       # Position-based row access
df.at[row, col]            # Fast access to single value (label-based)
df.iat[row_index, col_index]  # Fast access (integer-based)
df[df['col'] > 5]          # Filtering rows
df.query('col > 5')        # Query with string syntax
```

3. Data Manipulation (Add, Drop, Rename)

```
df['new_col'] = ...            # Add column
df.insert(loc, 'col', value)   # Insert at specific location
df.drop('col', axis=1)         # Drop column
df.drop(index=...)             # Drop rows
df.rename(columns={'old': 'new'})  # Rename columns
df.set_index('col')            # Set index
df.reset_index()               # Reset to default integer index
df.astype({'col': type})       # Change column type
```

4. Sorting & Ranking

```
df.sort_values('col')                  # Sort by column
df.sort_values(by=['col1', 'col2'])    # Sort by multiple columns
df.sort_index()                        # Sort by index
df.rank()                              # Rank data
```

5. Handling Missing Data

```
df.isnull()          # Detect NaNs
df.notnull()         # Opposite of isnull()
df.dropna()          # Drop rows with NaNs
df.fillna(value)     # Fill NaNs with value
df.interpolate()     # Interpolate missing values
df.ffill()           # Forward fill
```

```
df.bfill()          # Backward fill
```

6. Aggregation & Grouping

```
df.sum(), df.mean(), df.std(), df.min(), df.max(), df.count(), df.median(), df.mode()
df.agg(['sum', 'mean'])       # Multiple aggregations
df.groupby('col')             # Grouping
df.groupby(['col1', 'col2'])  # Multi-index grouping
df.pivot(index, columns, values)  # Pivot table
df.pivot_table(values, index, columns, aggfunc)  # Flexible pivot
```

7. Combining DataFrames

```
pd.concat([df1, df2])                 # Concatenate along axis
pd.merge(df1, df2, on='col')          # Merge on common column
df1.join(df2, how='left')             # Join on index
df.append(df2)                        # Append rows
```

8. Apply Functions

```
df.apply(func)            # Apply function to rows/columns
df.applymap(func)         # Apply function element-wise
df.map(func)              # For Series
df.transform(func)        # Transform without collapsing groups
```

9. String Methods (Text Columns)

```
df['col'].str.lower()
df['col'].str.upper()
df['col'].str.contains('a')
df['col'].str.replace('a', 'b')
df['col'].str.len()
df['col'].str.strip()
```

10. DateTime Methods

```
pd.to_datetime(df['date'])    # Convert to datetime
df['date'].dt.year
df['date'].dt.month
df['date'].dt.day
df['date'].dt.weekday
df['date'].dt.strftime('%Y-%m')
```

11. Input/Output

```
pd.read_csv('file.csv')          # Read CSV
df.to_csv('file.csv')            # Write CSV
pd.read_excel('file.xlsx')       # Read Excel
df.to_excel('file.xlsx')         # Write Excel
df.to_json('file.json')          # Export to JSON
pd.read_json('file.json')        # Read JSON
```

12. Miscellaneous

```
df.duplicated()       # Find duplicates
df.drop_duplicates()  # Remove duplicates
df.sample(n=5)        # Random sample
df.clip(lower=0)      # Limit values
```

```
df.corr()            # Correlation matrix
df.cov()             # Covariance
```

Extra

```
dir(df)              # List all attributes and methods
help(pd.DataFrame)   # Detailed documentation
```