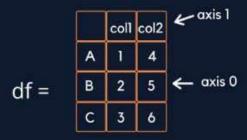
Pandas 🗓 **Cheat Sheet**

Pandas provides data analysis tools for Python. All of the following code examples refer to the dataframe below.



Getting Started

Import pandas:

import pandas as pd

Create a series:

```
s = pd.Series([1, 2, 3],
              index=['A', 'B', 'C'],
              name='col1')
```

Create a dataframe:

```
data = [[1, 4], [2, 5], [3, 6]]
index = ['A', 'B', 'C']
df = pd.DataFrame(data, index=index,
                  columns=['col1', 'col2'])
```

Load a dataframe:

```
df = pd.read_csv('filename.csv', sep=',',
                 names=['col1', 'col2'],
                 index col=0.
                 encoding='utf-8',
                 nrows=3)
```

Selecting rows and columns

```
Select single column:
 df['col1']
Select multiple columns:
 df[['col1', 'col2']]
Show first n rows:
 df.head(2)
Show last n rows:
 df.tail(2)
Select rows by index values:
 df.loc['A'] df.loc[['A', 'B']]
Select rows by position:
 df.loc[1] df.loc[1:]
```

Data wrangling

```
Filter by value:
df[df['col1'] > 1]
Sort by columns:
 df.sort_values(['col2', 'col2']
          ascending=[False, True])
Identify duplicate rows:
 df.duplicated()
Identify unique rows:
df['col1'].unique()
Swap rows and columns:
 df = df.transpose()
 df = df.T
Drop a column:
 df = df.drop('col1', axis=1)
Clone a data frame:
 clone = df.copy()
Connect multiple data frames vertically:
 df2 = df + 5 #new dataframe
 pd.concat([df,df2])
```

```
Merge multiple data frames horizontally:
 df3 = pd.DataFrame([[1, 7],[8,9]],
index=['B', 'D'],
columns=['col1', 'col3'])
#df3: new dataframe
Only merge complete rows (INNER JOIN):
 df.merge(df3)
Left column stays complete (LEFT OUTER JOIN):
 df.merge(df3, how='left')
Right column stays complete (RIGHT OUTER JOIN):
 df.merge(df3, how='right')
Preserve all values (OUTER JOIN):
 df.merge(df3, how='outer')
Merge rows by index:
 Fill NaN values:
 df.fillna(0)
Apply your own function:
 def func(x):
     return 2**x
 df.apply(func)
Arithmetics and statistics
```

```
Add to all values:
 df + 10
Sum over columns:
 df.sum()
Cumulative sum over columns:
 df.cumsum()
Mean over columns:
 df.mean()
Standard deviation over columns:
 df.std()
Count unique values:
 df['col1'].value counts()
Summarize descriptive statistics:
```

df.describe()