

# Reference guide: Python functions for structuring a dataset

## Pandas structuring reference guide

As you've learned, there are far too many Python functions to memorize all of them. That's why, as every data professional will tell you, you'll be using reference sheets and coding libraries nearly every day in your data analysis work.

The following reference guide will help you identify and correctly apply the most common Pandas functions used for structuring data.

### Functions for combining data

#### **merge()**

- Description: Use 'merge()' to take columns or indexes from other data frames and combine them.
- Example from Notebook:

```
df1.merge(['month','year'],on=list, sort=True)
```

#### **concat()**

- Description: Use 'concat()' to join columns or rows.
- Example from Notebook:

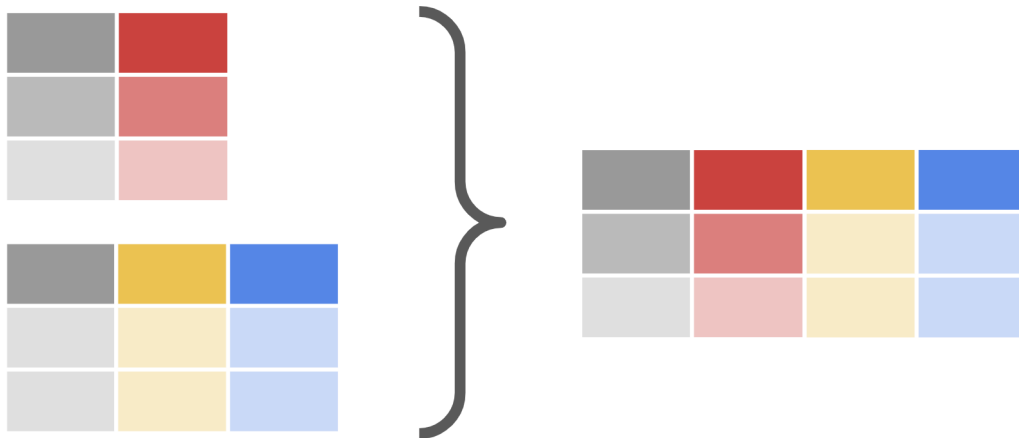
```
df3 = pd.concat([df1.drop(['weekday','week'],axis=1), df2])
```

#### **df.join()**

- Description: Use 'df.join()' to combine columns from other data frames.
- Example from Notebook:

```
df.join(other.set_index('key'), on='key')
```

Visual representation combination functions:



## Functions for extracting data

### **df[[columns]]**

- Description: Use 'df[[columns]]' to extract select columns for a data frame.
- Example(s) from Notebook:

```
df1[['day', 'week', 'weekday', 'month', 'year']]
```

### **df.select\_dtypes**

- Description: Use 'df.select\_dtypes' to filter a specific data type, like objects, strings, or integers.
- Example from Notebook:

```
df.select_dtypes(exclude=['int64'])
```

Visual representation of extraction functions:



## Functions for filtering data

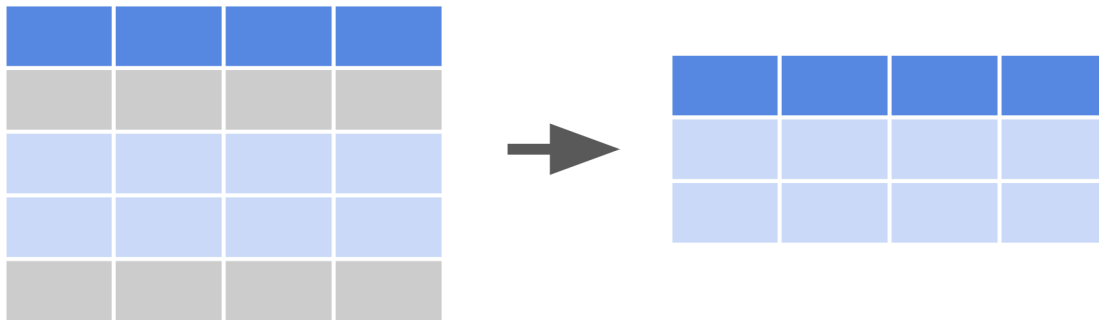
### **df[condition]**

- Description: Use 'df[condition]' to filter a data column according to selected variables.

- Example from Notebook:

df1['year' = > 2000]

Visual representation of filtering functions:



## Functions for sorting data

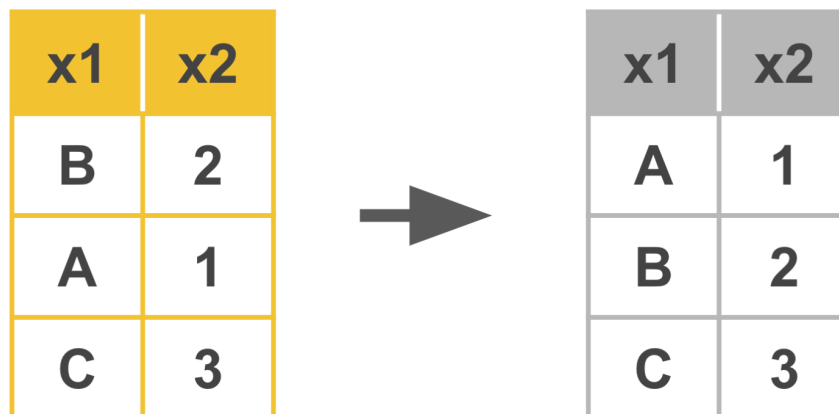
### **pd.sort\_values()**

- Description: Use 'pd.sort\_values()' to sort data according to selected parameters.

- Example from Notebook:

df.sort\_values(by=[date])

Visual representation of sorting functions:



## Functions for slicing data

### **df.iloc()**

- Description: Use 'df.iloc()' to slice data based on an integer location.

- Example from Notebook:

```
df.iloc(1, 10)
```

### **df.loc()**

- Description: Use 'df.loc()' to slice data based on label or boolean array.

- Example from Notebook:

```
df.loc(['weekday'], 10)
```

Visual representation of slicing functions:

1	2	3
---	---	---

1.5	2	3
4	5	6

1.5	2	3
4	5	6

## Key takeaways

The Python functions in this reference guide are foundational to structuring data, including filtering, sorting, merging, and slicing. You will find yourself using them throughout your career as a data professional.

## Resources for more information

Refer to these links for more details on Python functions and their various parameters.

- [Pandas documentation to describe parameters in Python functions](#)
- [W3schools provides explanations for Python functions in an easy-to-understand way](#)