# How to create dataframes from raw data

SHARP SIGHT

# What you'll learn

- How to create a new DataFrame
  - construct DataFrame from a *dictionary of lists*

- How to set the order the DataFrame columns during construction

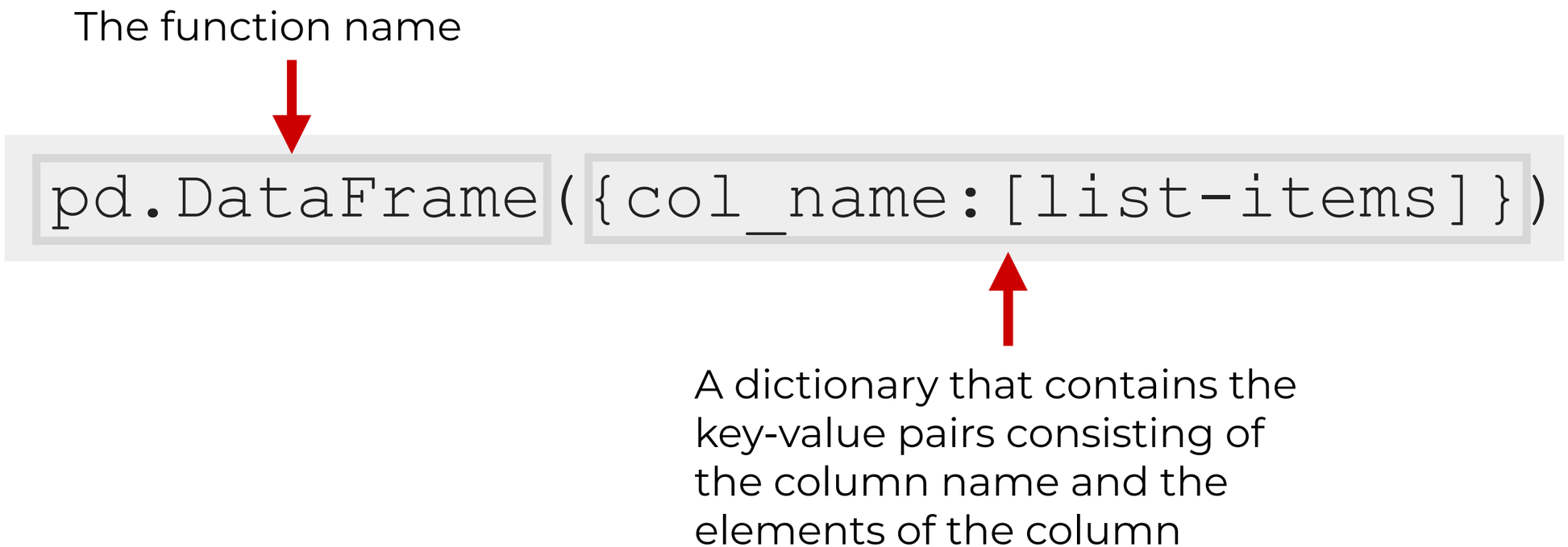
# Create a DataFrame from a dictionary of lists

# Create a DataFrame from a dictionary of lists

- You learned how to construct dictionaries and lists in previous lessons

- Now, we'll use dictionaries and lists to construct a DataFrame
  - Create a "dictionary of lists"
  - Pass the dictionary of lists to the `DataFrame()` constructor

# Syntax: create a dataframe from a dictionary of lists



# Syntax: create a dataframe from a dictionary of lists

A column name
(the "key" of the dictionary pair)

```
pd.DataFrame({col_name:[list-items]})
```

A list that contains the "elements" or data of the new column

(this is the "value" of the dictionary pair)

# Syntax: create a dataframe from a dictionary of lists

```
pd.DataFrame({col_name:[list-items]})
```

Note: here, there is only one name-list pair, but we can have as many as we want in our dataframe

# Example: create a DataFrame from a dictionary of lists

# Create a dataframe from a dictionary of lists

`raw_data` is a dictionary
(a dictionary of lists)

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
            ,'year': [1990,2004,2006,2013]
            ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data)
```

note: make sure you import Pandas with `import pandas as pd`

# Create a dataframe from a dictionary of lists

The dictionary contains several key-value pairs

Each pair consists of a column name and a list of elements

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
           ,'year': [1990,2004,2006,2013]
           ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data)
```

# Create a dataframe from a dictionary of lists

For example, `movie` is a column name

The list contains the names of the movies, which will be the elements of the `movie` column

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
            ,'year': [1990,2004,2006,2013]
            ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data)
```

# Create a dataframe from a dictionary of lists

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
            ,'year': [1990,2004,2006,2013]
            ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data)
```

We use the `pandas.DataFrame()` function to construct the dataframe from the dictionary

# This code creates a dataframe object

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
           ,'year': [1990,2004,2006,2013]
           ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data)
```

| movie | year | metascore |
|---|---|---|
| Goodfellas | 1990 | 89 |
| Aviator | 2004 | 77 |
| Departed | 2006 | 86 |
| Wolf of Wall Street | 2013 | np.NAN |

# How to set the order of the dataframe columns

# Syntax: set the order of dataframe columns

```
pd.DataFrame({col_name:[list-items]}, columns = )
```

We can use the `columns` parameter to set the column order

The argument to `columns` will be a list of the column names

# Example: set the order of dataframe columns

```
raw_data = {'movie': ['Goodfellas','Aviator','Departed','Wolf of Wall Street']
           ,'year': [1990,2004,2006,2013]
           ,'metascore': [89,77,86,np.NAN]}

movie_data = pd.DataFrame(raw_data, columns = ['movie','year','score'] )
```

We can use the `columns=` parameter to set the order or the names of the new columns

# Recap

# Recap of what we learned

- How to use the `pandas.DataFrame()` function

- How to create new dataframes from dictionaries
  - ... from a dictionary of lists

- How to change the columns of the dataframe using the `columns` parameter