Pandas Series

A Pandas Series is like a column in a table.

It is a one-dimensional array holding data of any type.

Create a simple Pandas Series from a list:

import pandas as pd  
  
a = [1, 7, 2]  
  
myvar = pd.Series(a)  
  
print(myvar)

## Labels

If nothing else is specified, the values are labeled with their index number. First value has index 0, second value has index 1 etc.

This label can be used to access a specified value.

Return the first value of the Series:

print(myvar[0])

## Create Labels

With the index argument, you can name your own labels.

Create your own labels:

import pandas as pd  
  
a = [1, 7, 2]  
  
myvar = pd.Series(a, index = ["x", "y", "z"])  
  
print(myvar)

When you have created labels, you can access an item by referring to the label.

Return the value of "y":

print(myvar["y"])

## Key/Value Objects as Series (Dictionary)

You can also use a key/value object, like a dictionary, when creating a Series.

Create a simple Pandas Series from a dictionary:

import pandas as pd  
  
calories = {"day1": 420, "day2": 380, "day3": 390}  
  
myvar = pd.Series(calories)  
  
print(myvar)

**Note:** The keys of the dictionary become the labels.

To select only some of the items in the dictionary, use the index argument and specify only the items you want to include in the Series.

Create a Series using only data from "day1" and "day2":

import pandas as pd  
  
calories = {"day1": 420, "day2": 380, "day3": 390}  
  
myvar = pd.Series(calories, index = ["day1", "day2"])  
  
print(myvar)

## DataFrames

Data sets in Pandas are usually multi-dimensional tables, called DataFrames.

Series is like a column, a DataFrame is the whole table.

Create a DataFrame from two Series:

import pandas as pd  
  
data = {  
  "calories": [420, 380, 390],  
  "duration": [50, 40, 45]  
}  
  
myvar = pd.DataFrame(data)  
  
print(myvar)

## What is a DataFrame?

A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

Create a simple Pandas DataFrame:

import pandas as pd  
  
data = {  
  "calories": [420, 380, 390],  
  "duration": [50, 40, 45]  
}  
  
#load data into a DataFrame object:  
df = pd.DataFrame(data)  
  
print(df)

## Locate Row

As you can see from the result above, the DataFrame is like a table with rows and columns.

Pandas use the loc attribute to return one or more specified row(s)

Return row 0:

#refer to the row index:  
print(df.loc[0])

Return row 0 and 1:

#use a list of indexes:  
print(df.loc[[0, 1]])

**Note:** When using [], the result is a Pandas **DataFrame**.

## Named Indexes

With the index argument, you can name your own indexes.

Add a list of names to give each row a name:

import pandas as pd  
  
data = {  
  "calories": [420, 380, 390],  
  "duration": [50, 40, 45]  
}  
  
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])  
  
print(df)

## Locate Named Indexes

Use the named index in the loc attribute to return the specified row(s).

Return "day2":

#refer to the named index:  
print(df.loc["day2"])

#### REMEMBER

* Getting data in to pandas from many different file formats or data sources is supported by read\_\* functions.
* Exporting data out of pandas is provided by different to\_\*methods.
* The head/tail/info methods and the dtypes attribute are convenient for a first check.