Pandas library Functions

Pandas is a Python library used for working with data sets. It helps in analysing, cleaning, exploring, and manipulating data. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

Create Alias with Pandas and framing data

Import pandas as pd

```
import pandas as pd

mydataset = {

  'cars': ["BMW", "Volvo", "Ford"],

  'passings': [3, 7, 2]

}

myvar = pd.DataFrame(mydataset)

print(myvar)
```

Checking Pandas Version

The version string is stored under `__version__` attribute.

```
import pandas as pd
print(pd.__version__)
```

Pandas Series

A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any type.

```
import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a)

print(myvar)
```

Labels

with index argument you can create your own labels.

```
import pandas as pd

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

Key/Value Objects as Series

You can also use a key/value object, like a dictionary, when creating a Series. Key of dictionary becomes the label.

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

Data Frames

Data sets in Pandas are usually multi-dimensional tables, called DataFrames. Series is like a column, a DataFrame is the whole table.

```
import pandas as pd

data = {

  "calories": [420, 380, 390],

  "duration": [50, 40, 45]

}

myvar = pd.DataFrame(data)

print(myvar)
```

Read CSV file

A simple way to store big data sets is to use CSV files (comma separated files).

CSV files contains plain text and is a well know format that can be read by everyone including Pandas.

In our examples we will be using a CSV file called 'data.csv'. To csv data into a DataFrame:

```
import pandas as pd

df = pd.read_csv('data.csv')
```

To print loaded data frame use `print(df.to_string())`

or simply use `print(df)`

To print maximum number of rows we use `print(pd.options.display.max_rows)`

To declare maximum number of rows `pd.options.display.max_rows = 9999`

Read JSON

Big data sets are often stored, or extracted as JSON.JSON is plain text, but has the format of an object, and is well known in the world of programming, including Pandas. In our examples we will be using a JSON file called 'data.json'.

```
Import pandas as pd

df=pd.read_json('data.json')

print(df)
```

Viewing the Data in DataFrame

One of the most used method for getting a quick overview of the DataFrame, is the `head()` method.

The `head()` method returns the headers and a specified number of rows, starting from the top.

Printing the first 10 rows of the DataFrame:

```
import pandas as pd

df = pd.read_csv('data.csv')

print(df.head(10))
```