

Pandas Tutorial

Pandas is a powerful and easy-to-use open-source tool built on top of the [Python](#) programming language. It is useful for data analysis and manipulation. [Python](#) with Pandas is widely used in Statistics, Finance, Neuroscience, Economics, Web Analytics, Advertising, etc.

To work with data sets, clean them, and make them relevant for Data Science is what Pandas do. With that, easily load and read data sets in Excel, CSV, JSON, XML, etc. formats with Pandas and work on them. Easily clean the wrong format data, remove duplicates, and do other tasks with Pandas.

See more results like this page



Excel Format



DataFrames in Pandas

The Pandas DataFrame is a Two-dimensional tabular data structure in Python, which is a table with rows and columns. The `DataFrame()` method is used for this purpose and has the following parameters:

Men

- **data:** The data to be stored in the Pandas DataFrame
- **index:** The index values to be provided for the resultant frame.
- **columns:** Set the column labels for the resultant frame if data does not mention before
- **dtype:** It is the datatype and only a single type is allowed.
- **copy:** To copy the input data

In this lesson, we will see some examples to:

- Create a Pandas DataFrame
- Access a group of rows or columns in a Pandas DataFrame
- Access a group of rows or columns by integer position in Pandas DataFrame

- Create a Pandas DataFrame
- Access a group of rows or columns in a Pandas DataFrame
- Access a group of rows or columns by integer positions in a Pandas DataFrame
- Name your own indexes in a Pandas DataFrame
- Iterating a DataFrame

Menu

Create a Pandas DataFrame

To create a dataframe in pandas, use the `pandas.DataFrame()` method. Let us see an example wherein we have student records:

```
import pandas as pd

# Dataset
data = {
    'student': ["Amit", "John", "Jacob", "D"],
    'rank': [1, 4, 3, 5, 2],
    'marks': [95, 70, 80, 60, 90]
}

df = pd.DataFrame(data)

print("Student Records\n\n", df)
```

Output

```
Student Records

  student  rank  marks
0    Amit    1    95
1    John    4    70
2   Jacob    3    80
3   David    5    60
4   Steve    2    90
```

Menu

The 0, 1, 2, etc. are the index or label that gets automatically added to the table.

Access a group of rows or columns in a Pandas DataFrame

Discover related topics

Arrays



Combine



Python



Inverse Matrix



Python Pandas



The **dataframe.loc** is used in Pandas to access a group of rows or columns in a

The `dataframe.loc` is used in Pandas to access a group of rows or columns in a DataFrame. Let us see an example:

```
import pandas as pd

# Dataset
data = {
    'Student': ["Amit", "John", "Jacob", "David", "Steve"],
    'Rank': [1, 4, 3, 5, 2],
    'Marks': [95, 70, 80, 60, 90]
}

# Create a DataFrame using the DataFrame()
df = pd.DataFrame(data, index=['RowA', 'RowB', 'RowC', 'RowD', 'RowE'])

print("Student Records\n\n", df)

# Access the value in the student column
print("\nValue = ", df.loc['RowA', 'Student'])
```

Output

Student Records

	Student	Rank	Marks
RowA	Amit	1	95
RowB	John	4	70
RowC	Jacob	3	80
RowD	David	5	60
RowE	Steve	2	90

Value = Amit

Access a group of rows or columns by integer positions in a Pandas DataFrame

The `dataframe.iloc` is used to access a

group of rows or columns by integers. We

Access a group of rows or columns by integer positions in a Pandas DataFrame

The `dataframe.iloc` is used to access a group of rows or columns by integers. We have also set columns and indexes. I see an example:

Men

```
import pandas as pd

# Dataset
data = {
    'Student': ["Amit", "John", "Jacob", "David"],
    'Rank': [1, 4, 3, 5, 2],
    'Marks': [95, 70, 80, 60, 90]
}

# Create a DataFrame using the DataFrame()
df = pd.DataFrame(data, index=['RowA', 'RowB', 'RowC', 'RowD', 'RowE'])

print("Student Records\n\n", df)

# Access using rows and columns by integer positions
print("\nValue = \n", df.iloc[[1,2]])
```

Output

Student Records

	Student	Rank	Marks
RowA	Amit	1	95
RowB	John	4	70
RowC	Jacob	3	80
RowD	David	5	60
RowE	Steve	2	90

Value =

	Student	Rank	Marks
RowB	John	4	70
RowC	Jacob	3	80



Name your indexes in a Pandas DataFrame

The `index` argument is used to set and name your indexes in a DataFrame. Let us see an example:

```
import pandas as pd

# Dataset
data = {
    'Student': ["Amit", "John", "Jacob", "David", "Steve"],
    'Rank': [1, 4, 3, 5, 2],
    'Marks': [95, 70, 80, 60, 90]
}

# Create a DataFrame using the DataFrame()
# The index argument is used to set the index
df = pd.DataFrame(data, index=['Student1', 'Student2', 'Student3', 'Student4', 'Student5'])

print("Student Records\n\n", df)
```

Output

```
Student Records
```

	Student	Rank	Marks
Student1	Amit	1	95
Student2	John	4	70
Student3	Jacob	3	80
Student4	David	5	60
Student5	Steve	2	90

Iterate a DataFrame

To iterate a DataFrame and display the column names, use the `for` loop as in the below example:

```

import pandas as pd

# Dataset
data = {
    'Student': ["Amit", "John", "Jacob",
    'Rank': [1, 4, 3, 5, 2],
    'Marks': [95, 70, 80, 60, 90]
}

# Create a DataFrame using the DataFrame()
# The index argument is used to set
df = pd.DataFrame(data, index=['Student1', 'Student2', 'Student3', 'Student4', 'Student5'])

print("Student Records\n\n", df)

# Iterating to display the columns
print("\nDisplaying the columns:")
for col in df:
    print(col)

```

Output

```

Student Records

      Student  Rank  Marks
Student1    Amit     1    95
Student2   John     4    70
Student3   Jacob     3    80
Student4   David     5    60
Student5   Steve     2    90

Displaying the columns:
Student
Rank
Marks

```

If you liked the tutorial, spread the word
and share the link and our website
Studyopedia with others:



For Videos, Join Our YouTube Channel:

[Join Now](#)