How to Drop Entries?

- Pandas provide pandas.DataFrame.drop() method to delete and filter data frame.
- **Rows or columns can be removed using index label or column name using this drop() method.**
- **axis param is used to specify what axis you would like to remove. By default axis = 0 meaning to remove rows. Use axis=1 or columns param to remove columns.**
- By default it doesn't remove on the existing DataFrame instead it returns a new DataFrame after dropping the columns specified with the drop method.
- In order to remove columns on the existing DataFrame object use inplace=True param.

Syntax:

> **DataFrame.drop(labels, axis, index, columns, inplace=False)**

Parameters:

1. **labels:** single label or list-like. referring row or column name.
2. **axis:** Use 1 to drop columns and 0 to drop rows from DataFrame.
3. **index or columns:** Single label or list. index or columns are an alternative to axis and cannot be used together.
4. **level:** Used to specify level in case data frame is having multiple level index.
5. **inplace:** Makes changes in original Data Frame if True.

Now, let's see the drop() syntax and how to delete or drop columns (two or more) from DataFrame with examples.

First, create a pandas DataFrame with a dictionary of lists. On our DataFrame, we have column names Courses, Fee and Duration.

```
import pandas as pd
technologies = ({
    'Courses':["Spark","PySpark","Hadoop","Python","pandas","Oracle","Java"],
    'Fee' :[20000,25000,26000,22000,24000,21000,22000],
    'Duration':['30day', '40days' ,'35days', '40days', '60days', '50days', '55days']
              })
df = pd.DataFrame(technologies)
print(df)
```

```
   Courses    Fee Duration
0    Spark  20000    30day
1  PySpark  25000   40days
2   Hadoop  26000   35days
3   Python  22000   40days
4   pandas  24000   60days
5   Oracle  21000   50days
6     Java  22000   55days
```

A) Drop Column

pandas drop() method remove the column by name and index from the DataFrame.

By default it doesn't remove on the existing DataFrame instead it returns a new DataFrame without the columns specified with the drop method.

In order to remove columns on the existing DataFrame object use inplace=True

Drop Column by Name

This example removes a column by name Fee from a DataFrame. Note that to use axis=1 in order to delete columns.

```
# Drops 'Fee' column
df2=df.drop(["Fee"], axis = 1)
print(df2)

----------------OR----------------

# Alternatively you can also use columns instead of labels.
df2=df.drop(columns=["Fee"], axis = 1)
```

```
   Courses Duration
0    Spark    30day
1  PySpark   40days
2   Hadoop   35days
3   Python   40days
4   pandas   60days
5   Oracle   50days
6     Java   55days
```

Drop Two or More Columns By Label Name

When you have a list of column names to drop, create a list object with the column names and use it with drop() method or directly use the list. The Below examples delete columns Courses and Fee from DataFrame.

```
df2=df.drop(["Courses", "Fee"], axis = 1)
print(df2)
```

```
  Duration
0    30day
1   40days
2   35days
3   40days
4   60days
5   50days
6   55days
```

Drop Column by Index

In order to remove the DataFrame columns by Index, first, we should get the DataFrame column as a list by using df.columns and then pick the column by index.

Note that the index starts from 0 in Python. On below example df.columns[1] represents the second column on DataFrame which is Fee.

```
# Drop column by index.
print(df.drop(df.columns[[1]], axis = 1))
```

```
   Courses Duration
0    Spark    30day
1  PySpark   40days
2   Hadoop   35days
3   Python   40days
4   pandas   60days
```

```
5   Oracle   50days
6     Java   55days
```

Drop Two or More Columns by Index

If you wanted to drop two or more columns by index, unfortunately, the drop() method doesn't take an index as parameter.

But we can overcome this by getting column names by index using df.columns[]. Use the below example to delete columns 0 and 1 (index starts from 0) index.

```
df2=df.drop(df.columns[[0,1]], axis = 1)
print(df2)
```

```
  Duration
0    30day
1   40days
2   35days
3   40days
4   60days
5   50days
6   55days
```

B) Drop Rows

Let's create a DataFrame, run some examples and explore the output.

```
import pandas as pd
import numpy as np

technologies = {
    'Courses':["Spark","PySpark","Hadoop","Python"],
    'Fee' :[20000,25000,26000,22000],
    'Duration':['30day','40days',np.nan, None],
    'Discount':[1000,2300,1500,1200]
               }

indexes=['r1','r2','r3','r4']
df = pd.DataFrame(technologies,index=indexes)

print(df)
```

```
     Courses    Fee Duration  Discount
r1     Spark  20000    30day      1000
r2   PySpark  25000   40days      2300
r3    Hadoop  26000      NaN      1500
r4    Python  22000     None      1200
```

1. Drop rows by Index Labels or Names

```
# Drop rows by Index Label
df = pd.DataFrame(technologies,index=indexes)

df1 = df.drop(['r1','r2'])
print(df1)
```

```
    Courses    Fee Duration  Discount
r3   Hadoop  26000      NaN      1500
```

```
     r4  Python  22000     None      1200
```

```
# Delete Rows by Index Labels
df1 = df.drop(index=['r1','r2'])
```

2. Drop Rows by Index Number (Row Number)

```
# Delete Rows by Index numbers
df = pd.DataFrame(technologies,index=indexes)

df1=df.drop(df.index[[1,3]])
print(df1)
```

```
        Courses    Fee Duration  Discount
     r1   Spark  20000    30day      1000
     r3  Hadoop  26000      NaN      1500
```

```
# Removes First Row
df=df.drop(df.index[0])

# Removes Last Row
df=df.drop(df.index[-1])
```

3. Delete Rows by Index Range

```
# Delete Rows by Index Range
df = pd.DataFrame(technologies,index=indexes)

df1=df.drop(df.index[2:])
print(df1)
```

```
         Courses    Fee Duration  Discount
     r1    Spark  20000    30day      1000
     r2  PySpark  25000   40days      2300
```

```
df1=df[4:]     # Returns rows from 4th row
df1=df[1:-1]   # Removes first and last row
df1=df[2:4]    # Return rows between 2 and 4
```