DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects. It is generally the most commonly used pandas object.

A pandas DataFrame can be created in the following way:
pandas.DataFrame(data, index, columns, dtype, copy)

Creating a DataFrame
A pandas DataFrame can be created using various inputs like:
Lists
dictionary
Series
Numpy ndarrays
DataFrame

Creating a empty DataFrame

```
import pandas as pd
df1 = pd.DataFrame()
print(df1)
```

o/p:

```
Empty DataFrame
Columns: []
Index: []
```

Creating a DataFrame using a List object:

```
lst1 = [1,2,3,4,5]
df2 = pd.DataFrame(lst1)
print(df2)
```

o/p:

```
  0
0 1
1 2
2 3
3 4
4 5
```

 # providing index to the dataframe:

```
df3 = pd.DataFrame(lst1, index = ['a','b','c','d','e'])
print(df3)
```

o/p:

```
   0
a  1
b  2
c  3
d  4
e  5
```

Note: index represents rows

 # assigning column names:

```
df4 = pd.DataFrame(lst1, index = ['a','b','c','d','e'], columns = ['col1'])
print(df4)
```

o/p:

```
    col1
a     1
b     2
c     3
d     4
e     5
```

 # creating a emp dataframe:

```
lst2 = [['Amar', 30], ['Akbar', 31], ['Anthony', 32]]
df5 = pd.DataFrame(lst2, index = ['Emp1', 'Emp2','Emp3'], columns =
['Emp_Name', 'Age'])
print(df5)
```

o/p:

```
      Emp_Name   Age
Emp1     Amar     30
Emp2    Akbar     31
Emp3  Anthony    32
```

<u>Create a DataFrame from Dict of ndarrays / Lists</u>

All the **ndarrays** must be of same length. If index is passed, then the length of the index should equal to the length of the arrays.
If no index is passed, then by default, index will be range(n), where **n** is the array length.

Example 1

```
import pandas as pd
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'], 'Age':[28,34,29,42]}
df6 = pd.DataFrame(data)
print(df6)
```

o/p:

```
  Age Name
0 28 Tom
1 34 Jack
2 29 Steve
3 42 Ricky
```

Note: Observe the values 0,1,2,3. They are the default index assigned to each using the
function range(n).

Example 2

Let us now create an indexed DataFrame using arrays.

```
import pandas as pd
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]}
df7 = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
print(df7)
```

o/p:

```
      Age Name
rank1 28 Tom
rank2 34 Jack
```

rank3 29 Steve
rank4 42 Ricky

Create a DataFrame from Dictionary object:

List of Dictionaries can be passed as input data to create a DataFrame. The dictionary keys are by default taken as column names.

Example 1

The following example shows how to create a DataFrame by passing a list of dictionaries.

```
import pandas as pd
data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]
df7 = pd.DataFrame(data)
print(df7)
```

o/p:

```
   a b c
0 1 2 NaN
1 5 10 20.0
```

Note: NaN (Not a Number) is used to represent a missing value.

Example 2

The following example shows how to create a DataFrame by passing a list of dictionaries and the row indices.

```
import pandas as pd
data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]
df8 = pd.DataFrame(data, index=['first', 'second'])
print(df8)
```

o/p:

```
      a b c
first 1 2 NaN
second 5 10 20.0
```

Example 3

The following example shows how to create a DataFrame with a list of dictionaries, row indices, and column indices.

```
import pandas as pd
data = [{'a': 1, 'b': 2},{'a': 5, 'b': 10, 'c': 20}]

#With two column indices, values same as dictionary keys
df9 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b'])
```

o/p:

```
       a b
first  1 2
second 5 10
```

```
df10 = pd.DataFrame(data, index=['first', 'second'], columns=['a', 'b1'])
```

o/p:

```
       a b1
```

first 1 NaN
second 5 NaN

Create a DataFrame from Dictionary of Series

Dictionary of Series can be passed to form a DataFrame. The resultant index is the union of all the series indexes passed.

Example

```
import pandas as pd
d = {'one':pd.Series([1, 2, 3], index=['a', 'b', 'c']),
'two' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}
df11 = pd.DataFrame(d)
print(df11)
```

o/p:

```
  one two
a 1.0 1
b 2.0 2
c 3.0 3
d NaN 4
```