**Exp.6 and 7 To add the missing value in any data set.**

Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in a real-life scenarios. Missing Data can also refer to as NA (Not Available) values in pandas. In DataFrame sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed. For Example, Suppose different users being surveyed may choose not to share their income, some users may choose not to share the address in this way many datasets went missing. None: None is a Python singleton object that is often used for missing data in Python code.

In Pandas missing data is represented by two value:

**None:** None is a Python singleton object that is often used for missing data in Python code.

**NaN :** NaN (an acronym for Not a Number), is a special floating-point value recognized by all systems

**NaN :** NaN (an acronym for Not a Number), is a special floating-point value recognized by all systems

Pandas treat None and NaN as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful **functions** for detecting, removing, and replacing null values in Pandas DataFrame :

isnull()

notnull()

dropna()

fillna()

replace()

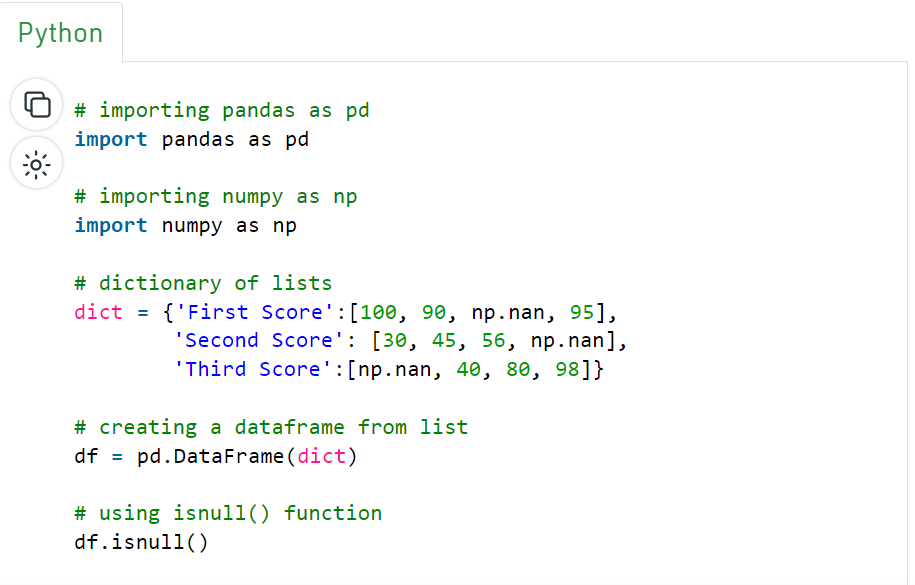
interpolate()

**A Checking for missing values using isnull() and notnull()**

In order to check missing values in Pandas DataFrame, we use a function isnull() and notnull(). Both function help in checking whether a value is NaN or not. These function can also be used in Pandas Series in order to find null values in a series.

In order to check null values in Pandas DataFrame, we use isnull() function this function return dataframe of Boolean values which are True for NaN values.

**Code #1:**



# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

'Second Score': [30, 45, 56, np.nan],

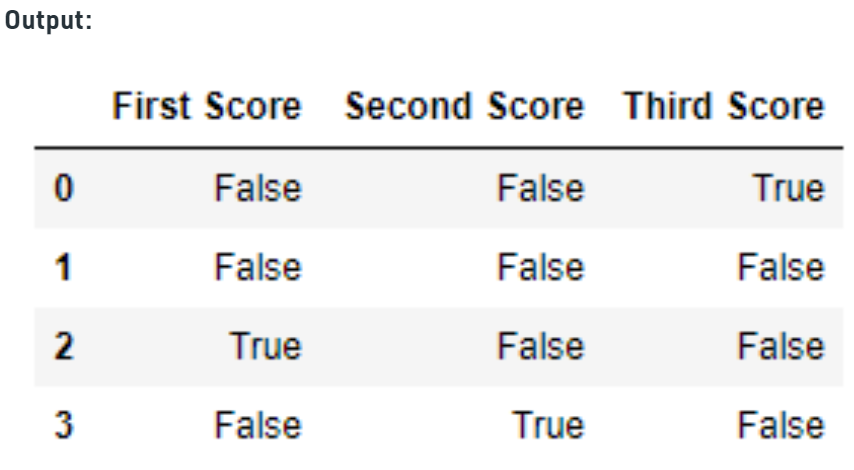
'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe from list

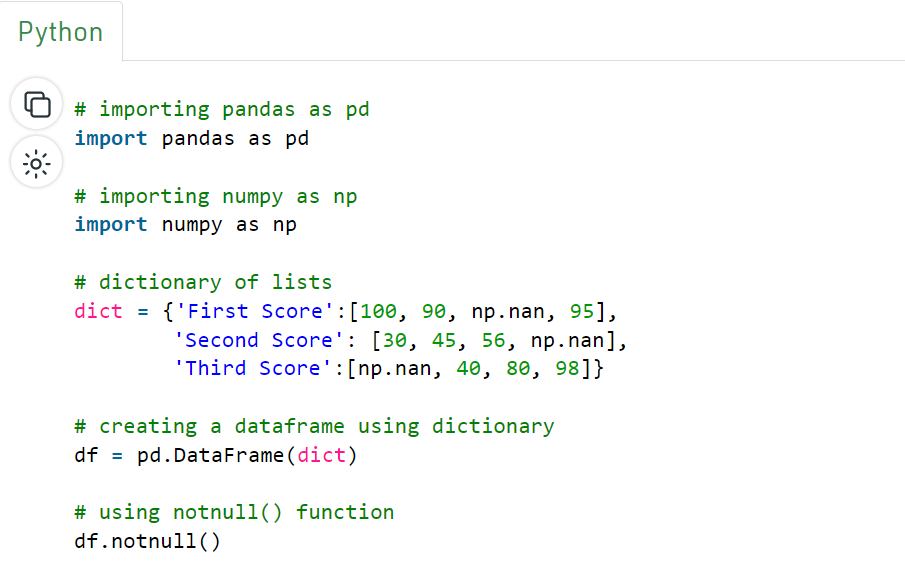
df = pd.DataFrame(dict)

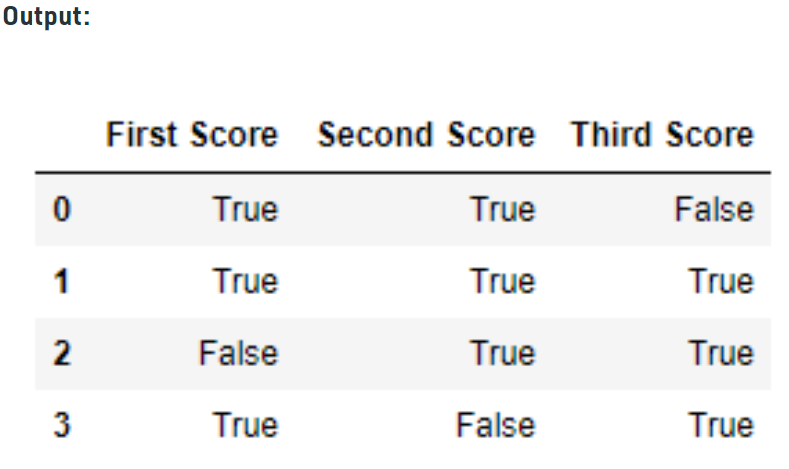
# using isnull() function

df.isnull()



**Code #2:**

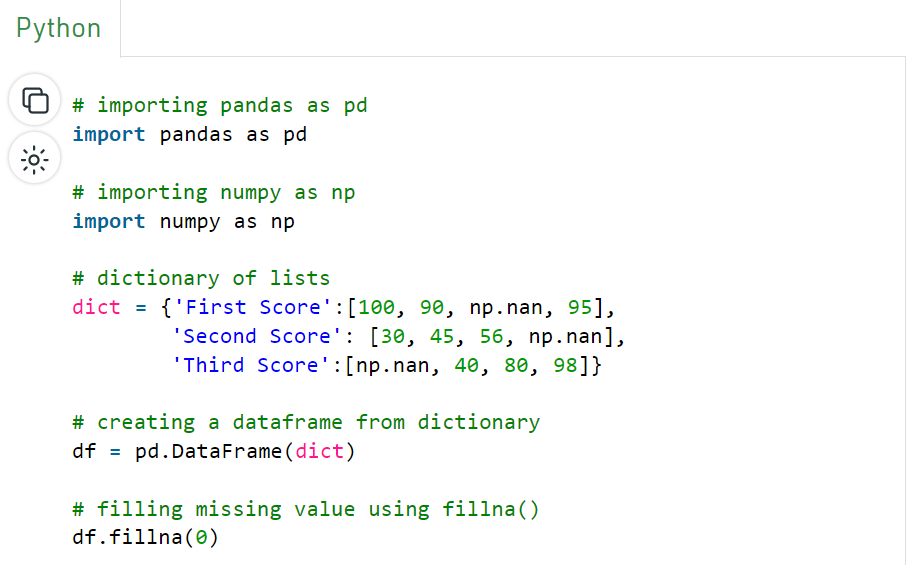


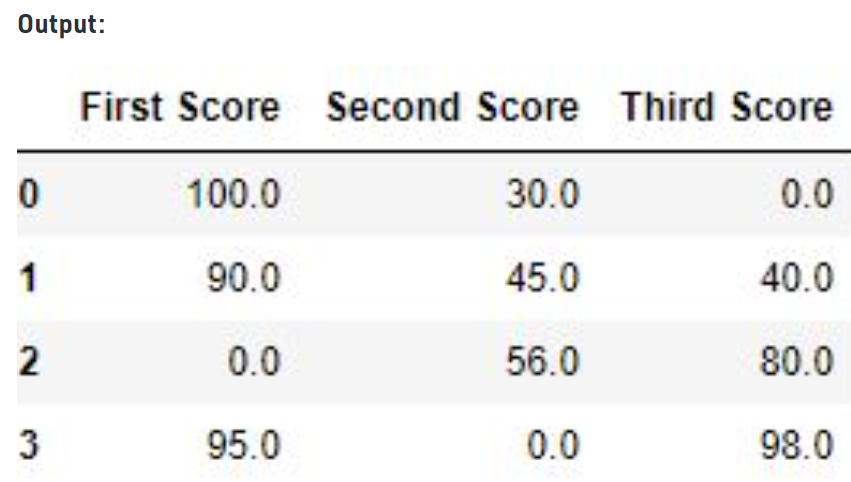


**B Filling missing values using fillna(), replace() and interpolate()**

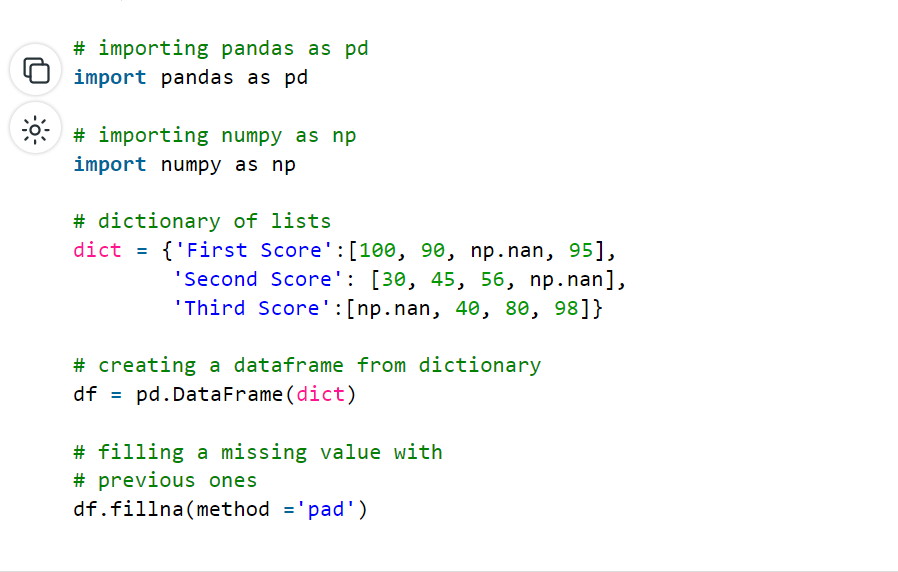
In order to fill null values in a datasets, we use fillna(), replace() and interpolate() function these function replace NaN values with some value of their own. All these function help in filling a null values in datasets of a DataFrame. Interpolate() function is basically used to fill NA values in the dataframe but it uses various interpolation technique to fill the missing values rather than hard-coding the value.

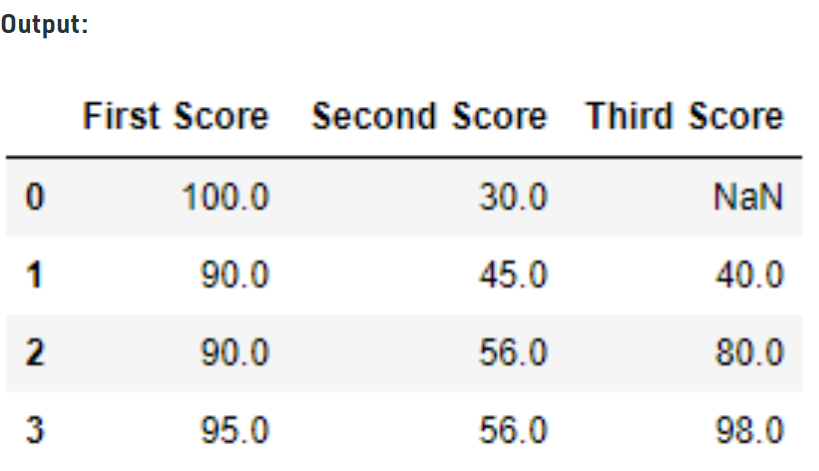
**Code #1: Filling null values with a single value**



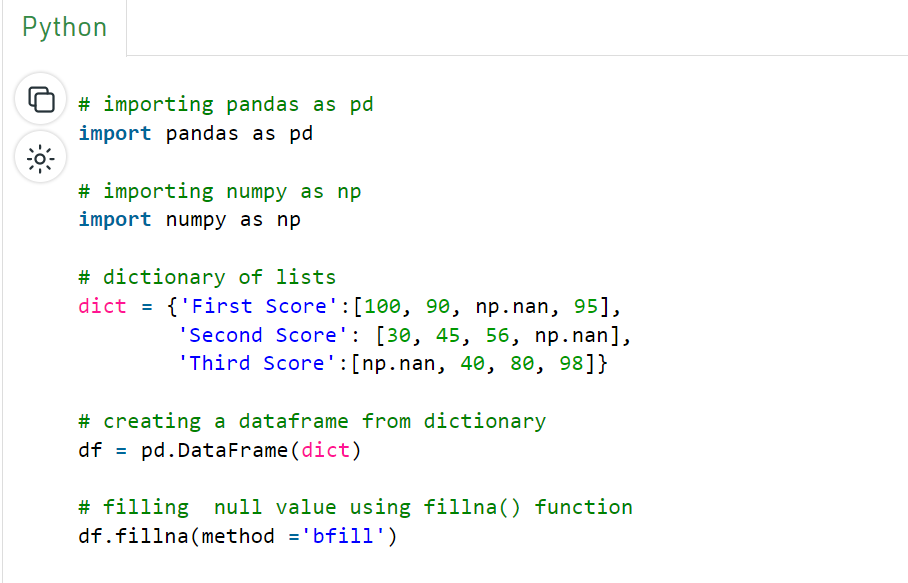


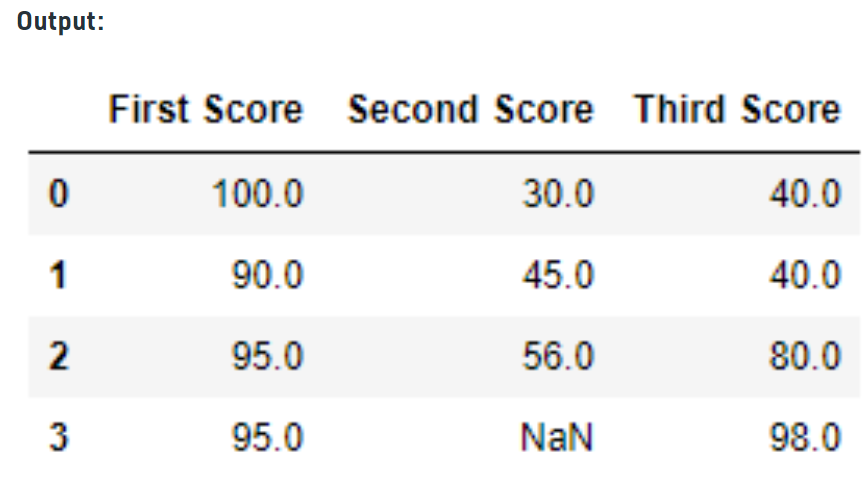
**Code #2: Filling null values with the previous ones**

****

****

**Code #3: Filling null value with the next ones**

****

****

**Further reading:** Filling null values in CSV file fillna() and replace()

**Link:** https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/