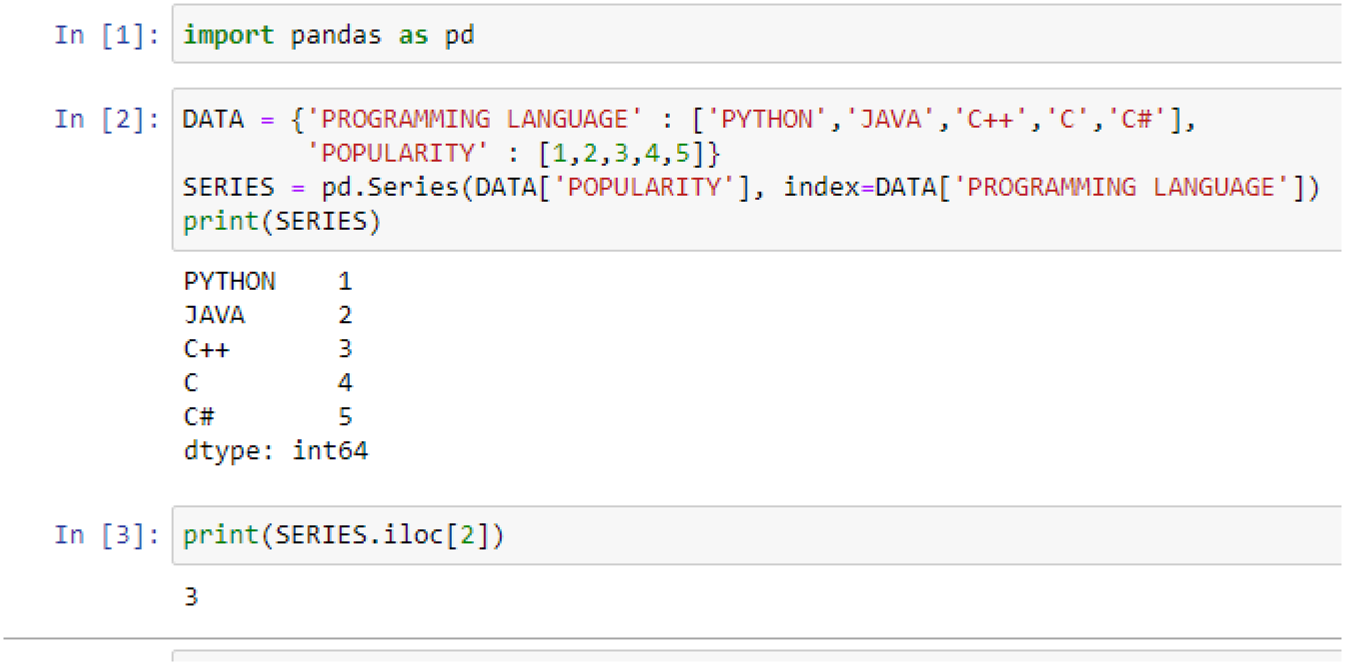
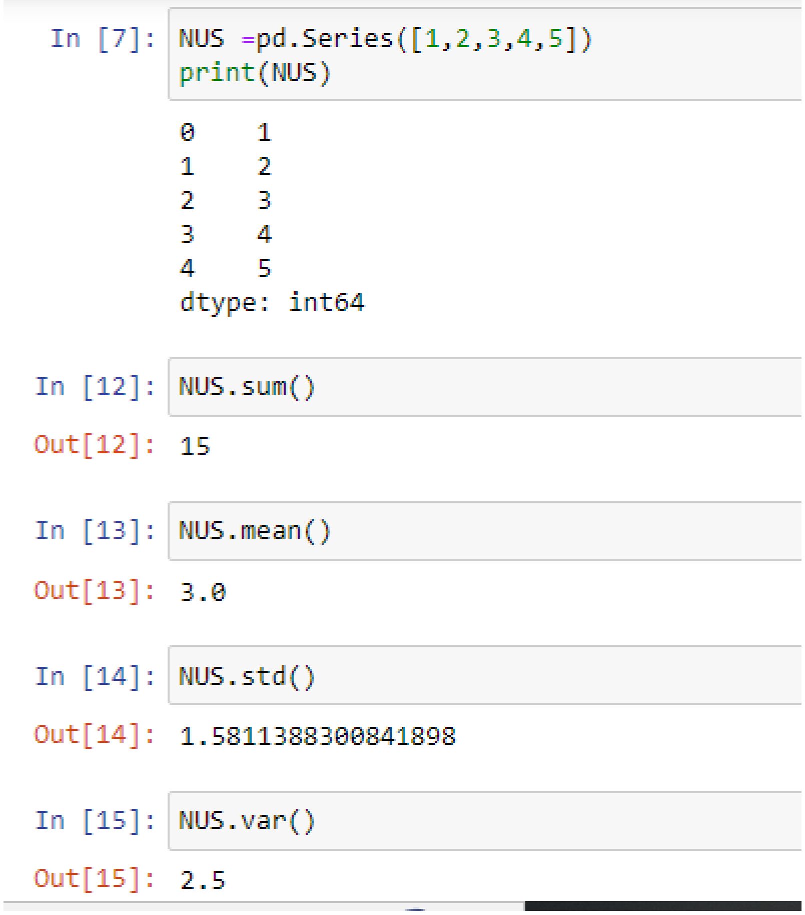
Exercise 1:
Create a pandas series containing the top 5 programming languages used in 2021, along with their respective popularity index (in descending order). Then, extract the third item from the series.



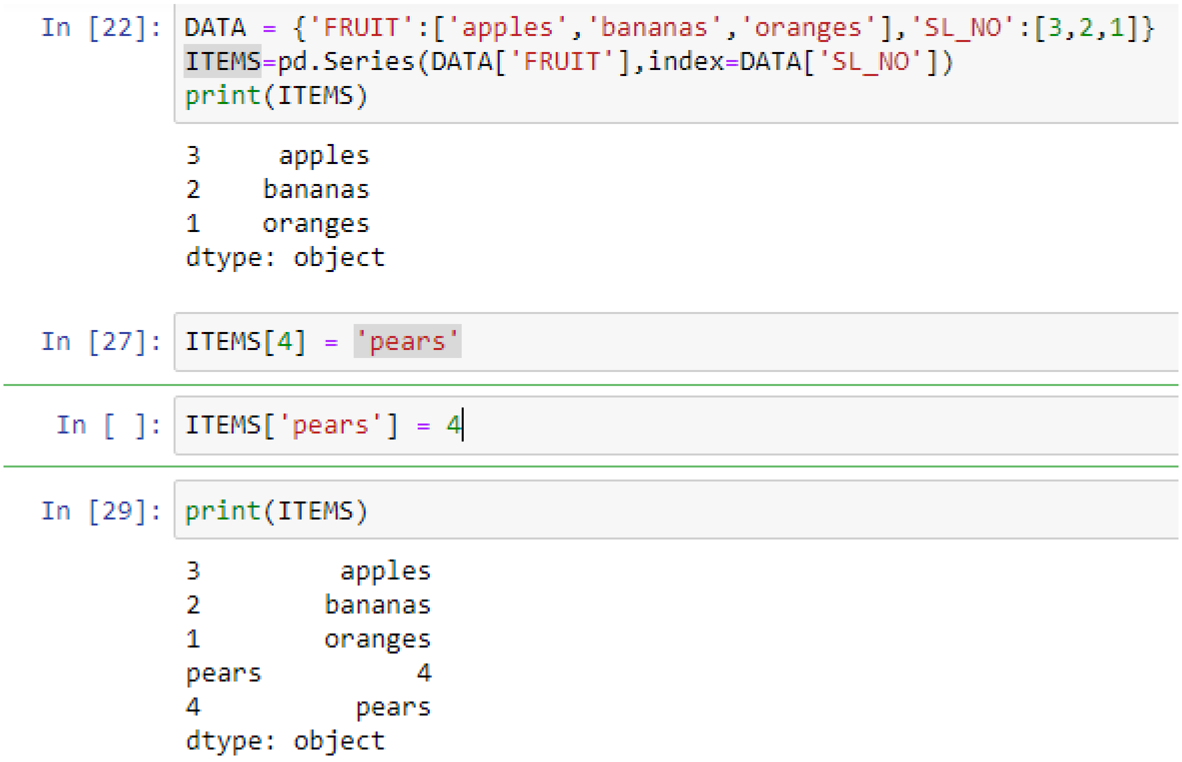
```
In [1]: import pandas as pd

In [2]: DATA = {'PROGRAMMING LANGUAGE' : ['PYTHON','JAVA','C++','C','C#'],
                'POPULARITY' : [1,2,3,4,5]}
        SERIES = pd.Series(DATA['POPULARITY'], index=DATA['PROGRAMMING LANGUAGE'])
        print(SERIES)

        PYTHON    1
        JAVA      2
        C++       3
        C         4
        C#        5
        dtype: int64

In [3]: print(SERIES.iloc[2])

        3
```

Exercise 2:
Create a Pandas series with the following data: [1, 2, 3, 4, 5]. Then, calculate the sum, mean, and standard deviation of the series.



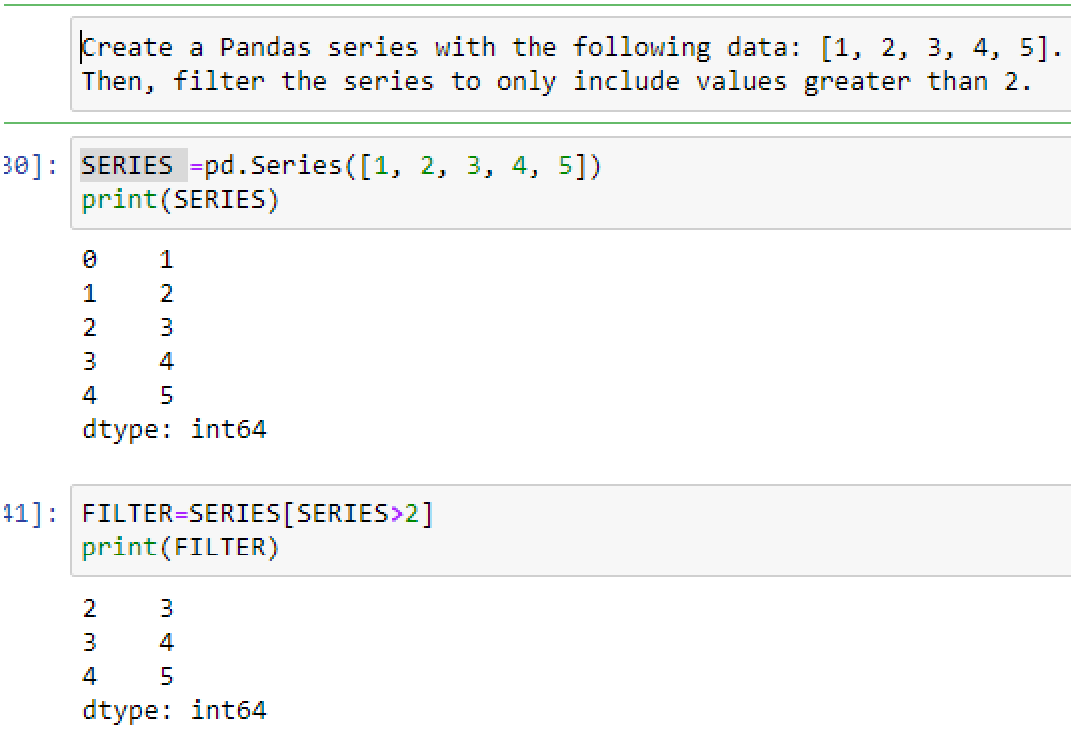
```
In [7]: NUS =pd.Series([1,2,3,4,5])
        print(NUS)

        0    1
        1    2
        2    3
        3    4
        4    5
        dtype: int64

In [12]: NUS.sum()

Out[12]: 15

In [13]: NUS.mean()

Out[13]: 3.0

In [14]: NUS.std()

Out[14]: 1.5811388300841898

In [15]: NUS.var()

Out[15]: 2.5
```

Exercise 3:
Create a Pandas series with the following data: {'apples': 3, 'bananas': 2, 'oranges': 1}. Then, add a new item to the series with the key 'pears' and the value 4.



```
In [22]: DATA = {'FRUIT':['apples','bananas','oranges'],'SL_NO':[3,2,1]}
         ITEMS=pd.Series(DATA['FRUIT'],index=DATA['SL_NO'])
         print(ITEMS)

         3     apples
         2    bananas
         1    oranges
         dtype: object

In [27]: ITEMS[4] = 'pears'

In [ ]: ITEMS['pears'] = 4

In [29]: print(ITEMS)

         3          apples
         2         bananas
         1         oranges
         pears           4
         4           pears
         dtype: object
```

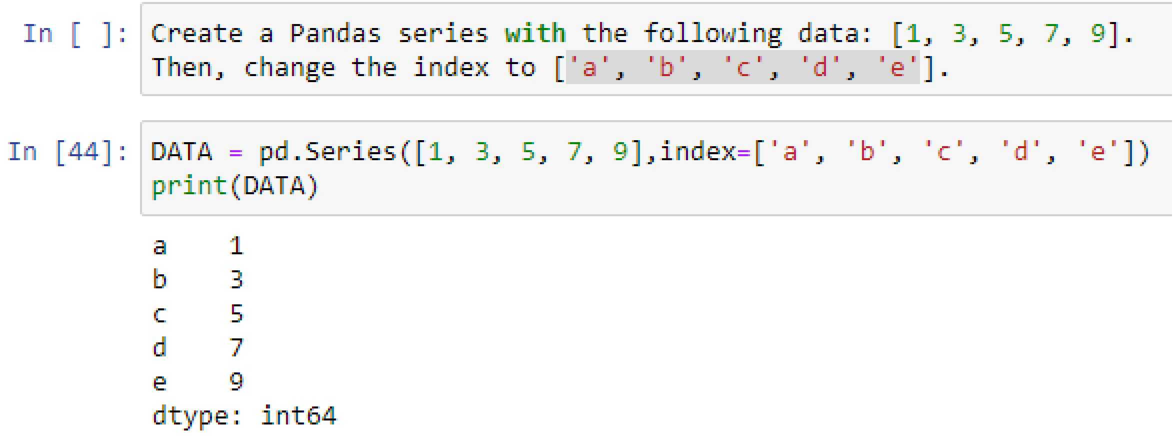
Exercise 4:
Create a Pandas series with the following data: [1, 2, 3, 4, 5]. Then, filter the series to only include values greater than 2.



```
      Create a Pandas series with the following data: [1, 2, 3, 4, 5].
      Then, filter the series to only include values greater than 2.

30]:  SERIES =pd.Series([1, 2, 3, 4, 5])
      print(SERIES)

      0    1
      1    2
      2    3
      3    4
      4    5
      dtype: int64

41]:  FILTER=SERIES[SERIES>2]
      print(FILTER)

      2    3
      3    4
      4    5
      dtype: int64
```

Exercise 5:

.Create a Pandas series with the following data: [1, 3, 5, 7, 9]. Then, change the index to ['a', 'b', 'c', 'd', 'e'].

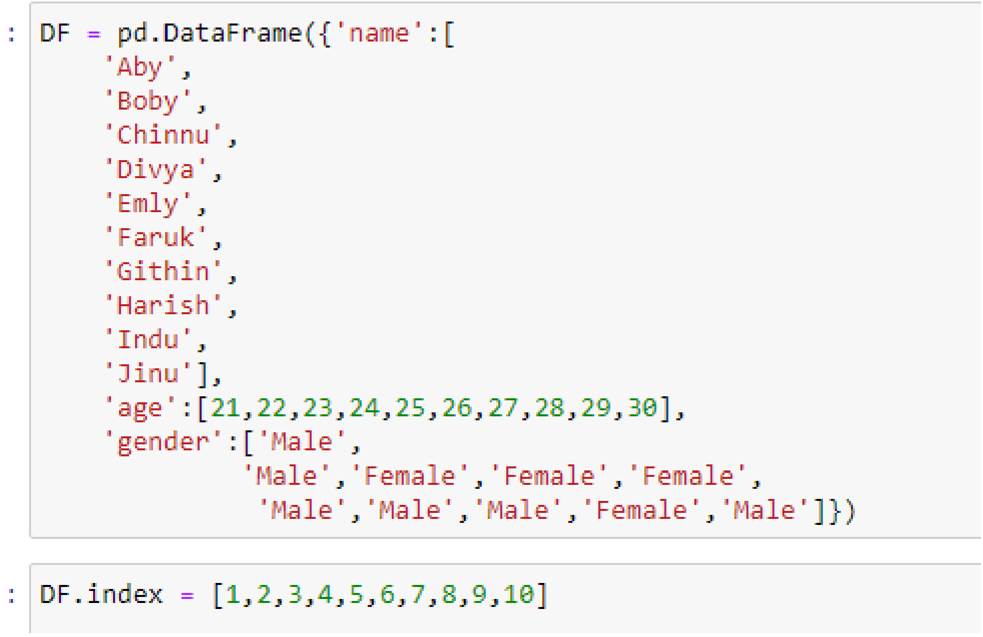


```
Create a Pandas series with the following data: [1, 3, 5, 7, 9].
Then, change the index to ['a', 'b', 'c', 'd', 'e'].
```

```
DATA = pd.Series([1, 3, 5, 7, 9],index=['a', 'b', 'c', 'd', 'e'])
print(DATA)
```

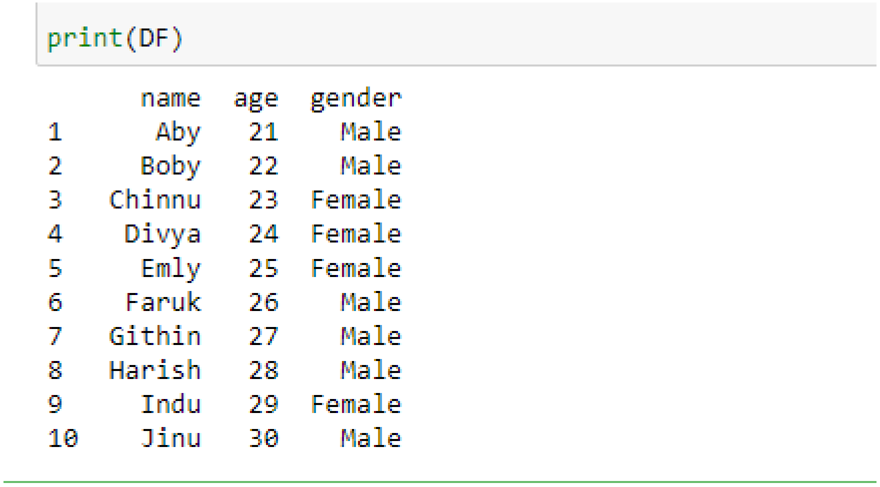
```
a    1
b    3
c    5
d    7
e    9
dtype: int64
```

exercise 1:
Create a dataframe with the following columns: name, age, and gender. The dataframe should have 10 rows of data, with the following values



```
DF = pd.DataFrame({'name':[
    'Aby',
    'Boby',
    'Chinnu',
    'Divya',
    'Emly',
    'Faruk',
    'Githin',
    'Harish',
    'Indu',
    'Jinu'],
    'age':[21,22,23,24,25,26,27,28,29,30],
    'gender':['Male',
             'Male','Female','Female','Female',
              'Male','Male','Male','Female','Male']})
```

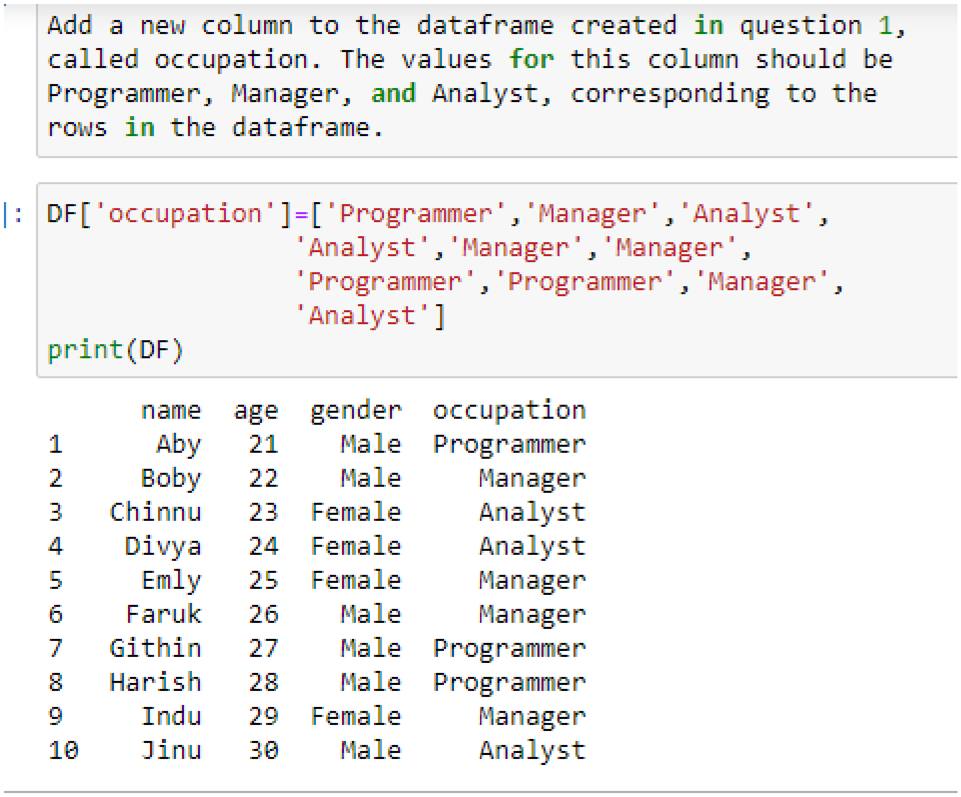
```
DF.index = [1,2,3,4,5,6,7,8,9,10]
```



```
print(DF)
```

```
      name  age  gender
1      Aby   21    Male
2     Boby   22    Male
3   Chinnu   23  Female
4    Divya   24  Female
5     Emly   25  Female
6    Faruk   26    Male
7   Githin   27    Male
8   Harish   28    Male
9     Indu   29  Female
10    Jinu   30    Male
```

Exercise 2:
Add a new column to the dataframe created in question 1, called occupation. The values for this column should be Programmer, Manager, and Analyst, corresponding to the rows in the dataframe.

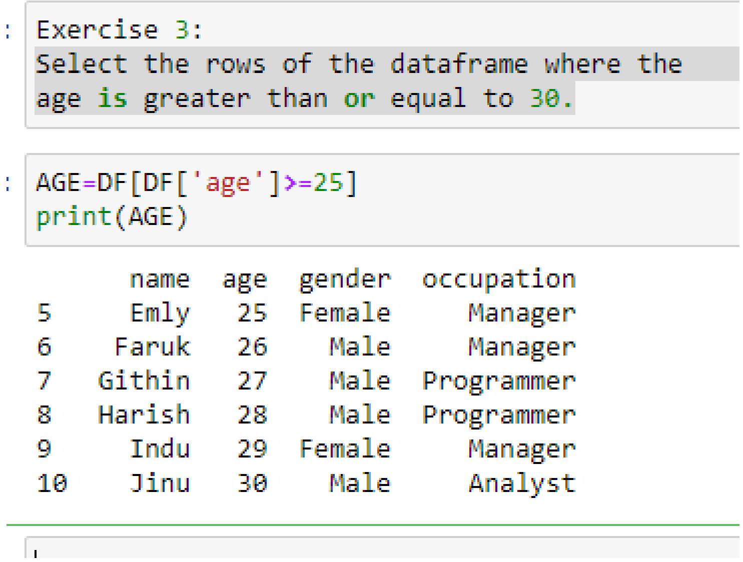


Add a new column to the dataframe created in question 1, called occupation. The values for this column should be Programmer, Manager, and Analyst, corresponding to the rows in the dataframe.

```
DF['occupation']=['Programmer','Manager','Analyst',
                  'Analyst','Manager','Manager',
                  'Programmer','Programmer','Manager',
                  'Analyst']
print(DF)
```

```
      name  age  gender  occupation
1      Aby   21    Male  Programmer
2     Boby   22    Male     Manager
3  Chinnu   23  Female     Analyst
4    Divya   24  Female     Analyst
5     Emly   25  Female     Manager
6    Faruk   26    Male     Manager
7  Githin   27    Male  Programmer
8  Harish   28    Male  Programmer
9     Indu   29  Female     Manager
10    Jinu   30    Male     Analyst
```

Exercise 3:
Select the rows of the dataframe where the age is greater than or equal to 25.

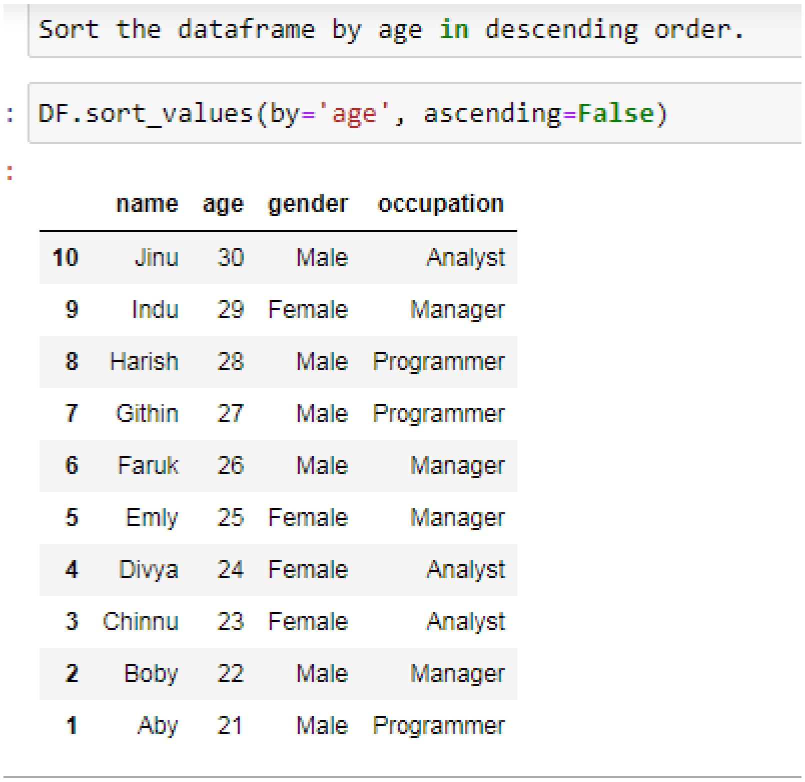


Exercise 3:
Select the rows of the dataframe where the age is greater than or equal to 30.

```
AGE=DF[DF['age']>=25]
print(AGE)
```

```
      name  age  gender  occupation
5     Emly   25  Female     Manager
6    Faruk   26    Male     Manager
7   Githin   27    Male  Programmer
8   Harish   28    Male  Programmer
9     Indu   29  Female     Manager
10    Jinu   30    Male     Analyst
```

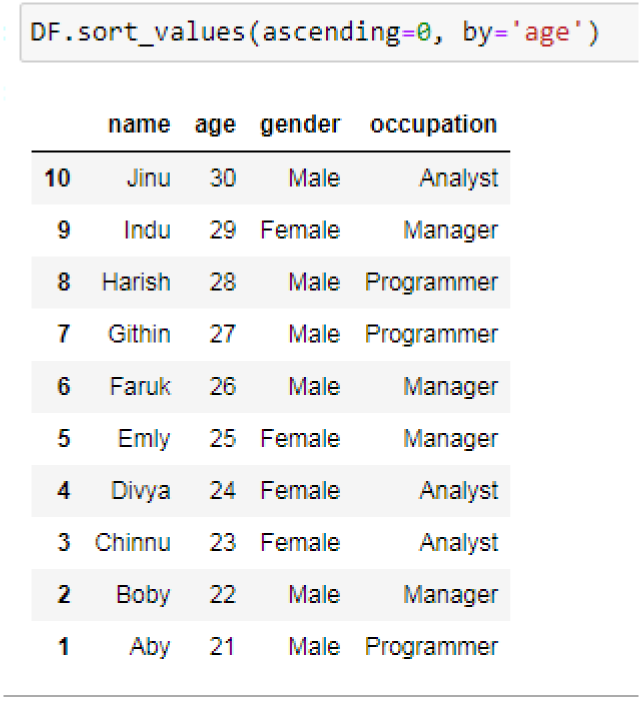
Exercise 4:
Sort the dataframe by age in descending order.



Sort the dataframe by age in descending order.

```
DF.sort_values(by='age', ascending=False)
```

	name	age	gender	occupation
10	Jinu	30	Male	Analyst
9	Indu	29	Female	Manager
8	Harish	28	Male	Programmer
7	Githin	27	Male	Programmer
6	Faruk	26	Male	Manager
5	Emly	25	Female	Manager
4	Divya	24	Female	Analyst
3	Chinnu	23	Female	Analyst
2	Boby	22	Male	Manager
1	Aby	21	Male	Programmer



```
DF.sort_values(ascending=0, by='age')
```

	name	age	gender	occupation
10	Jinu	30	Male	Analyst
9	Indu	29	Female	Manager
8	Harish	28	Male	Programmer
7	Githin	27	Male	Programmer
6	Faruk	26	Male	Manager
5	Emly	25	Female	Manager
4	Divya	24	Female	Analyst
3	Chinnu	23	Female	Analyst
2	Boby	22	Male	Manager
1	Aby	21	Male	Programmer