Task 14
Task for class on 23-10-2022:
File Handling:

Exercise 1:

Write a Python program to read a file and display its contents

Exercise 2:

Write a Python program to count the number of lines in a file

Exercise 3:

Write a Python program to write some text to a file

Exercise 4:

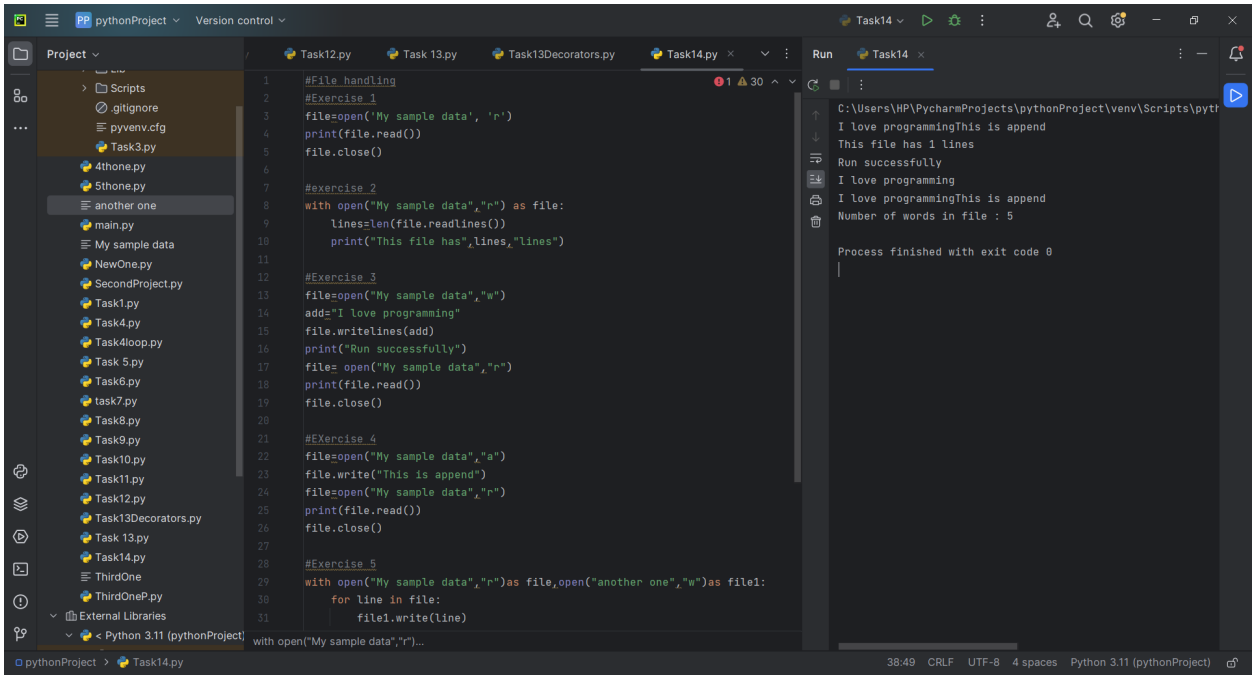
Write a Python program to append some text to a file.

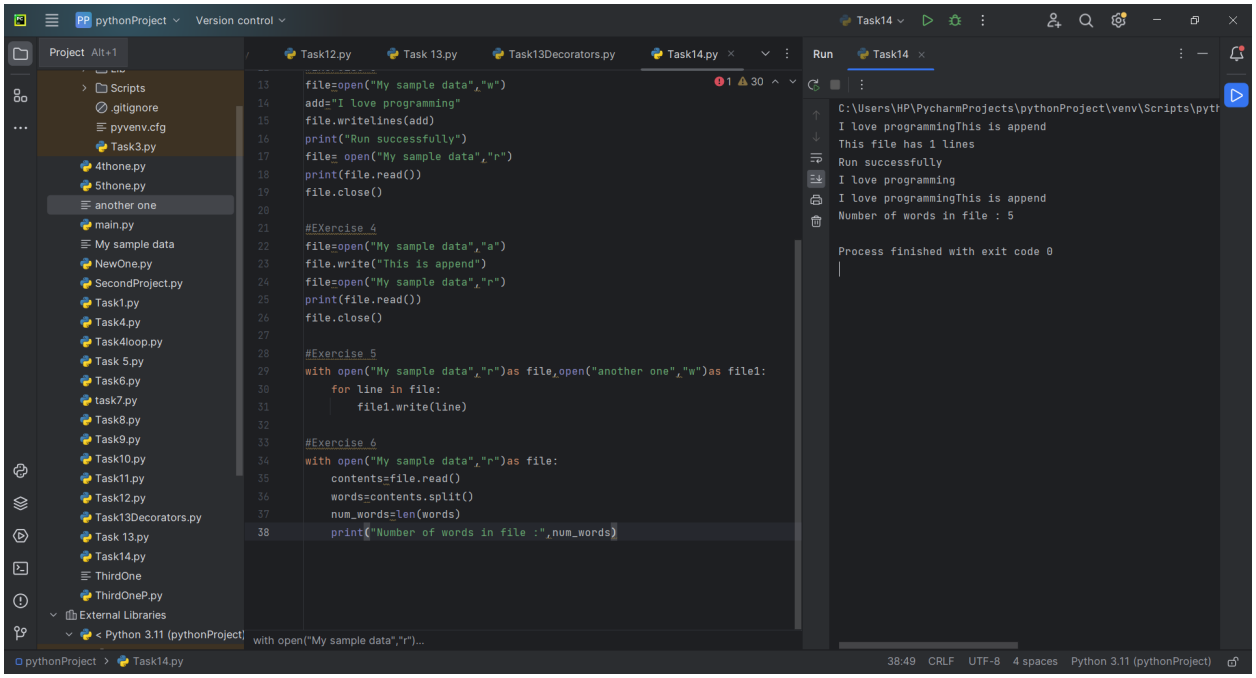
Exercise 5:

Write a Python program to copy the contents of one file to another file

Exercise 6:

Write a Python program to read the content of a file and count the total number of words in that file.





Task15

Task for class on 27-10-2023

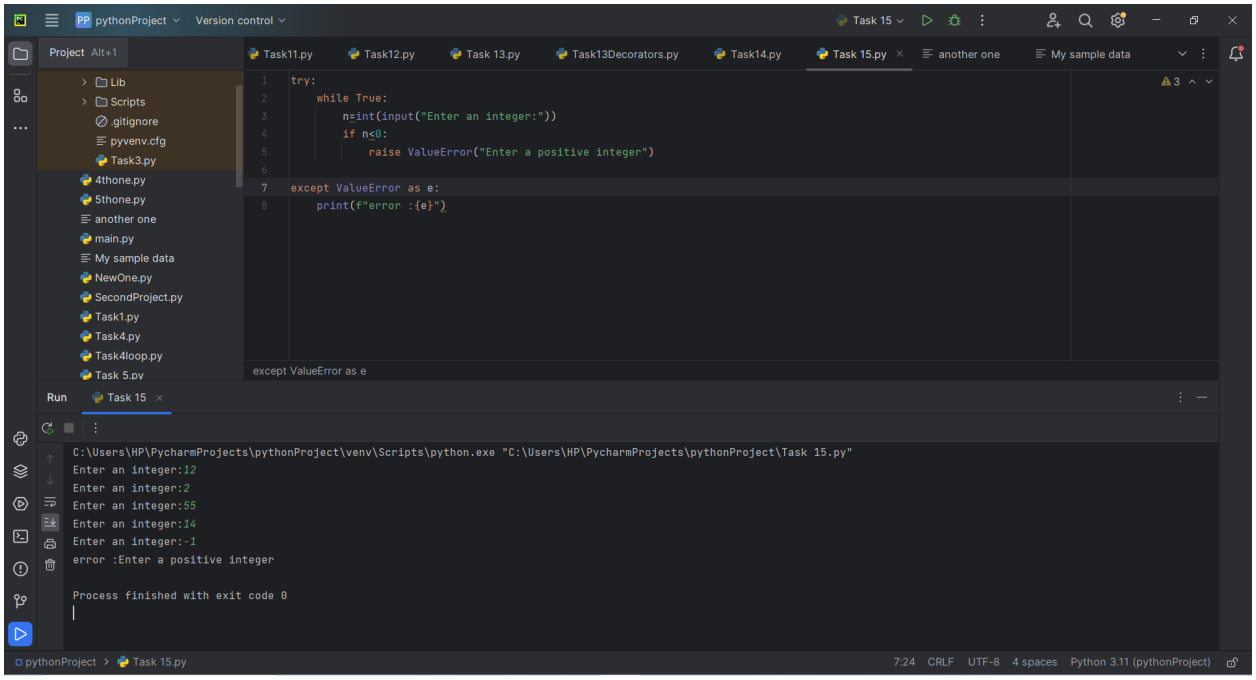
Exception Handling

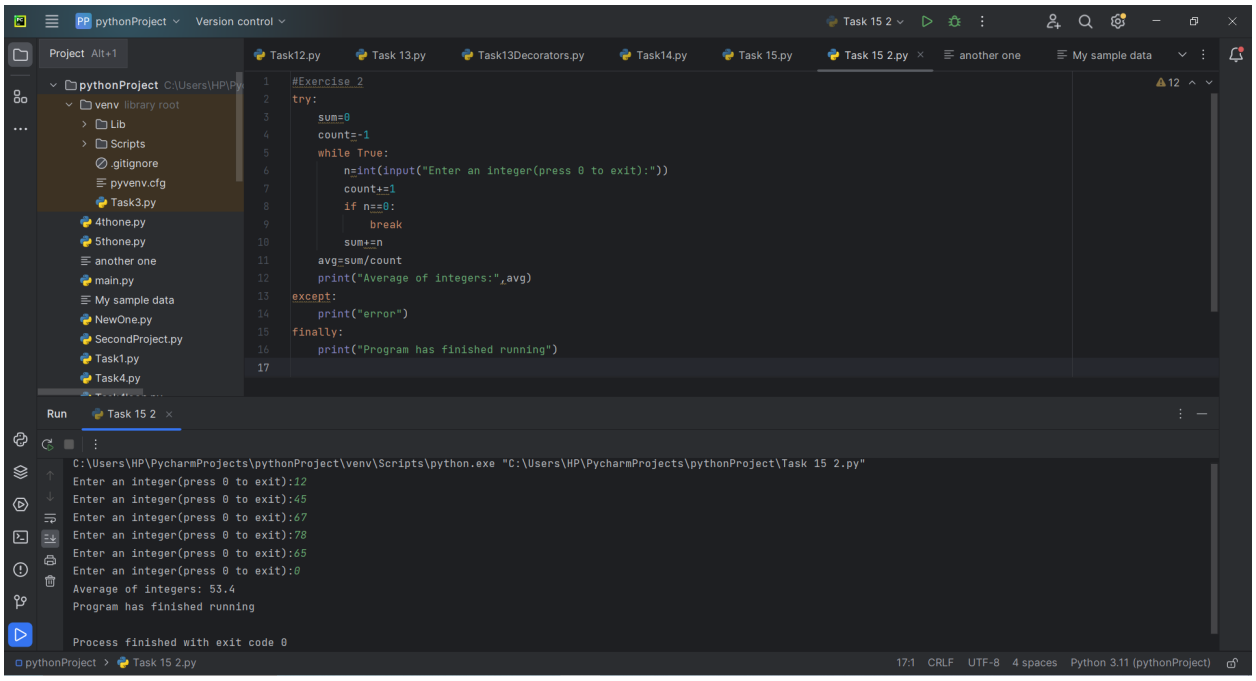
Exercise 1

Write a Python program that prompts the user to input a list of integers and raises an exception if any of the integers in the list are negative.

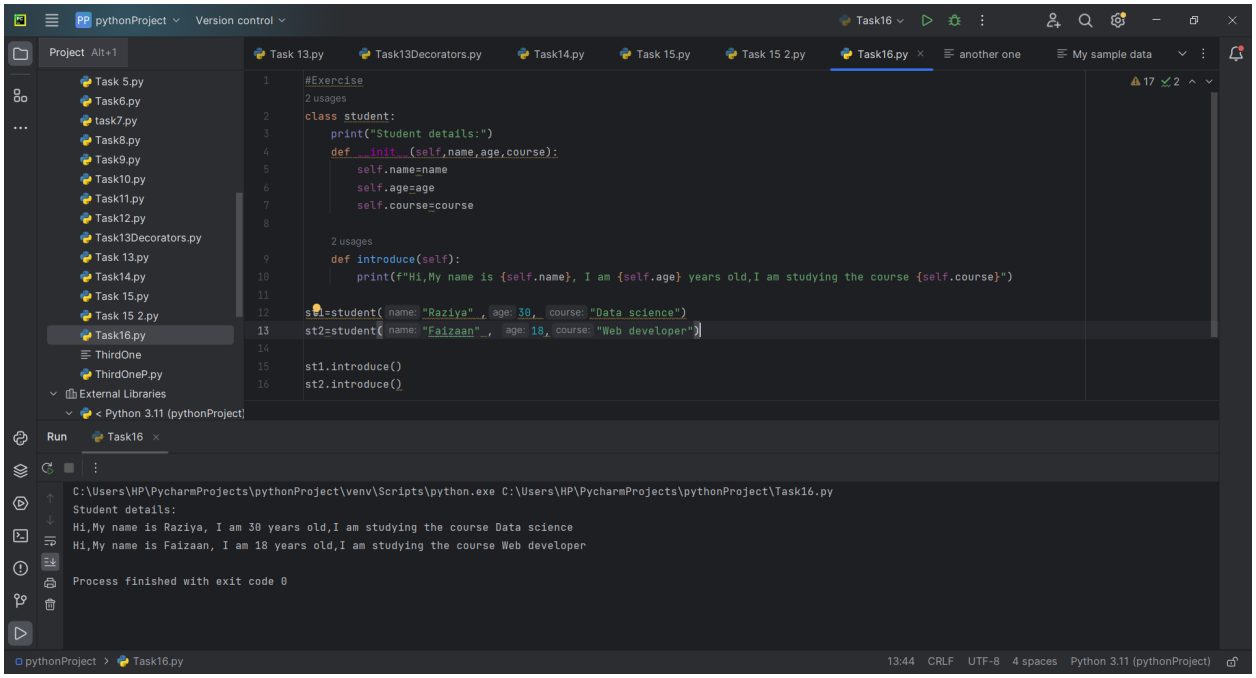
Exercise 2

Write a Python program that prompts the user to input a list of integers and computes the average of those integers. Use try-except blocks to handle any exceptions that might occur. Use the finally clause to print a message indicating that the program has finished running.





Task 16
Task for class on 30-10-2023
Classes and Objects:
Create a Python class called Student with a constructor that initializes attributes for a student's name, age, and course. Then, write a method that allows a student to introduce themselves. Finally, create two instances of the Student class and demonstrate how to use them.

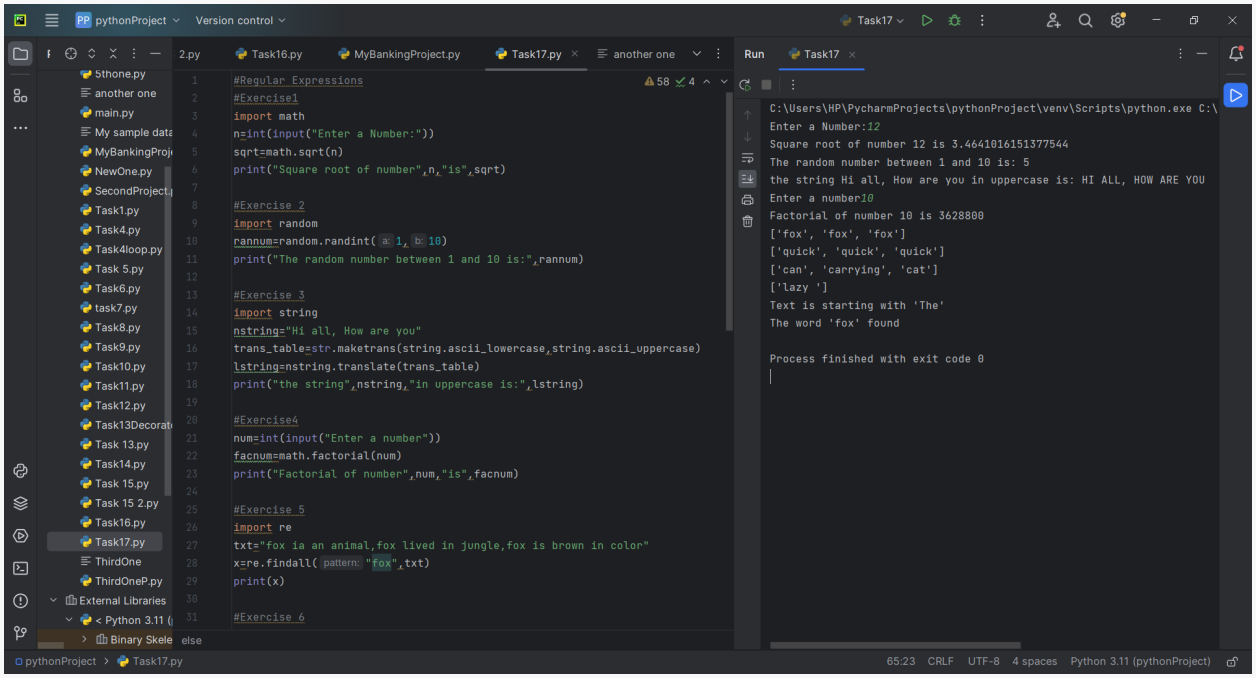


Task17

Task for class conducted on 06-11-2023:

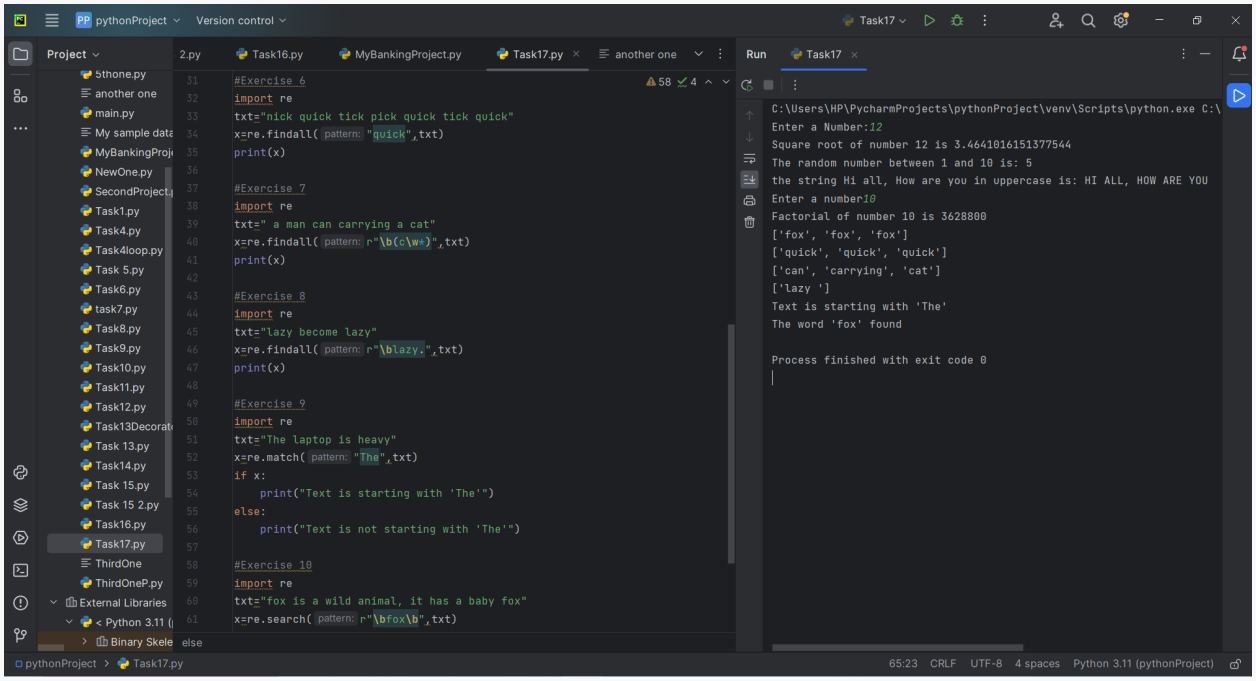
Regex:

1 Write a Python program to calculate the square root of a number using the math module.
2 Write a Python program to generate a random number between 1 and 10 using the random module.
3 Write a Python program to convert a string to uppercase using the string module.
4 Write a Python program to calculate the factorial of a number using the math module.
5 Write a regular expression to find all occurrences of the word "fox" in the given text
6 Write a regular expression to find all occurrences of the word "quick" in the given text
7 Write a regular expression to find all occurrences of words that start with the letter "c" in the given text.
8 Write a regular expression to find all occurrences of the word "lazy" followed by any one character in the given text.
9 Use match() to find if the text starts with "The"
10 Use search() to find the first occurrence of the word "fox"



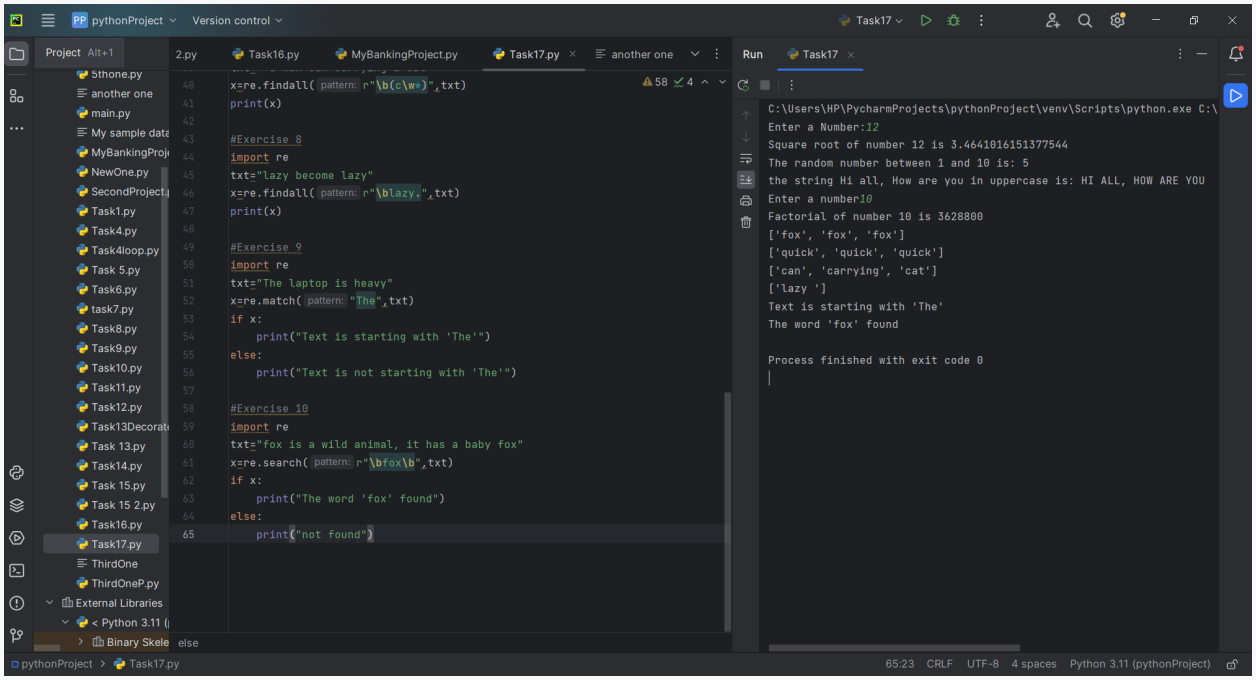
```
#Regular Expressions
#Exercise1
import math
n=int(input("Enter a Number:"))
sqrt=math.sqrt(n)
print("Square root of number",n,"is",sqrt)

#Exercise 2
import random
rannum=random.randint( a: 1, b: 10)
print("The random number between 1 and 10 is:",rannum)

#Exercise 3
import string
nstring="Hi all, How are you"
trans_table=str.maketrans(string.ascii_lowercase,string.ascii_uppercase)
lstring=nstring.translate(trans_table)
print("the string",nstring,"in uppercase is:",lstring)

#Exercise4
num=int(input("Enter a number"))
facnum=math.factorial(num)
print("Factorial of number",num,"is",facnum)

#Exercise 5
import re
txt="fox ia an animal,fox lived in jungle,fox is brown in color"
x=re.findall( pattern: "fox",txt)
print(x)

#Exercise 6
```

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\
Enter a Number:12
Square root of number 12 is 3.4641016151377544
The random number between 1 and 10 is: 5
the string Hi all, How are you in uppercase is: HI ALL, HOW ARE YOU
Enter a number10
Factorial of number 10 is 3628800
['fox', 'fox', 'fox']
['quick', 'quick', 'quick']
['can', 'carrying', 'cat']
['lazy ']
Text is starting with 'The'
The word 'fox' found

Process finished with exit code 0
```



```
#Exercise 6
import re
txt="nick quick tick pick quick tick quick"
x=re.findall( pattern: "quick",txt)
print(x)

#Exercise 7
import re
txt=" a man can carrying a cat"
x=re.findall( pattern: r"\b(c\w*)",txt)
print(x)

#Exercise 8
import re
txt="lazy become lazy"
x=re.findall( pattern: r"\blazy.",txt)
print(x)

#Exercise 9
import re
txt="The laptop is heavy"
x=re.match( pattern: "The",txt)
if x:
    print("Text is starting with 'The'")
else:
    print("Text is not starting with 'The'")

#Exercise 10
import re
txt="fox is a wild animal, it has a baby fox"
x=re.search( pattern: r"\bfox\b",txt)
```

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\
Enter a Number:12
Square root of number 12 is 3.4641016151377544
The random number between 1 and 10 is: 5
the string Hi all, How are you in uppercase is: HI ALL, HOW ARE YOU
Enter a number10
Factorial of number 10 is 3628800
['fox', 'fox', 'fox']
['quick', 'quick', 'quick']
['can', 'carrying', 'cat']
['lazy ']
Text is starting with 'The'
The word 'fox' found

Process finished with exit code 0
```



Project

Task for class conducted on 03-11-2023:

OOPs:

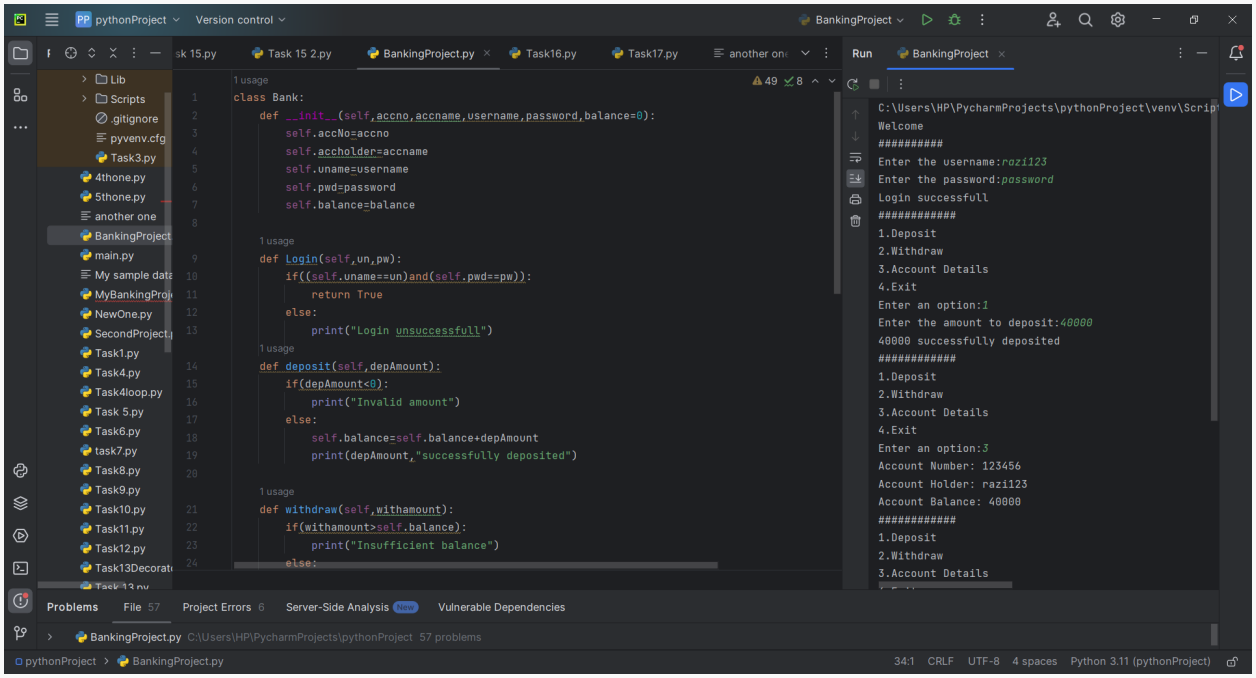
We have covered core python, now its time to test your coding and logical skills, also to know how confident you are with the syntax, i'm giving you a small project:

Write a python program to replicate a Banking system. The following features are mandatory:

1.Account login

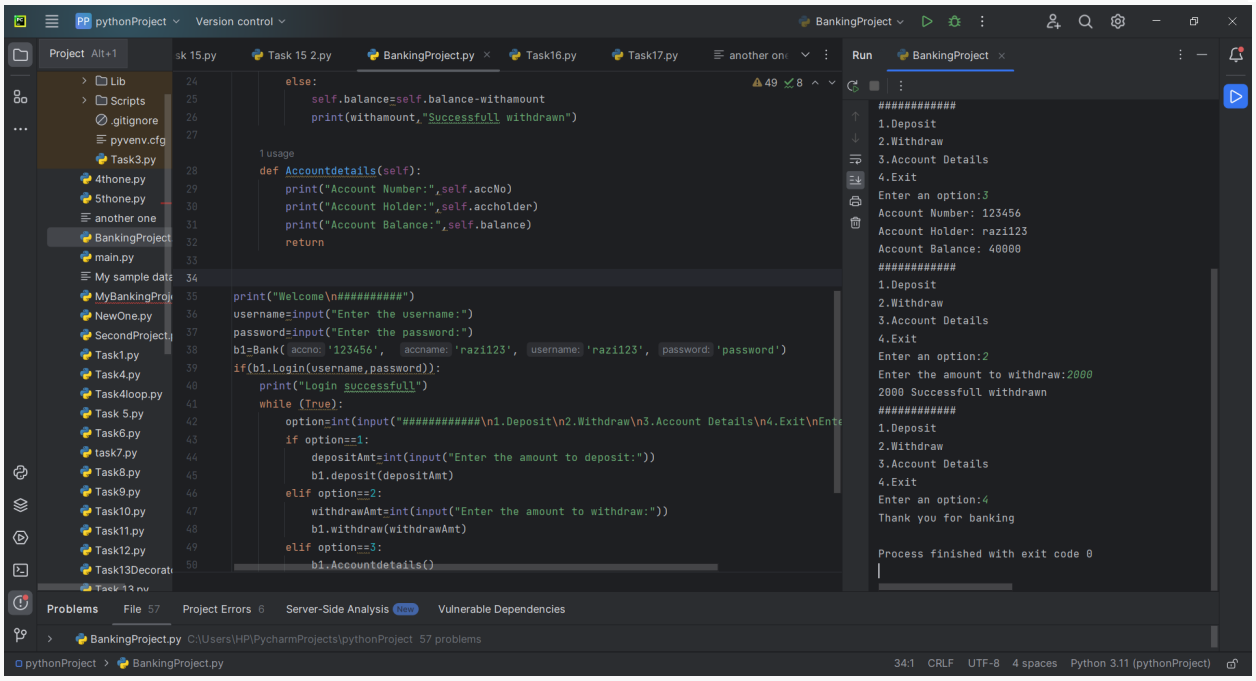
2. Amount Depositing

3. Amount Withdrawal



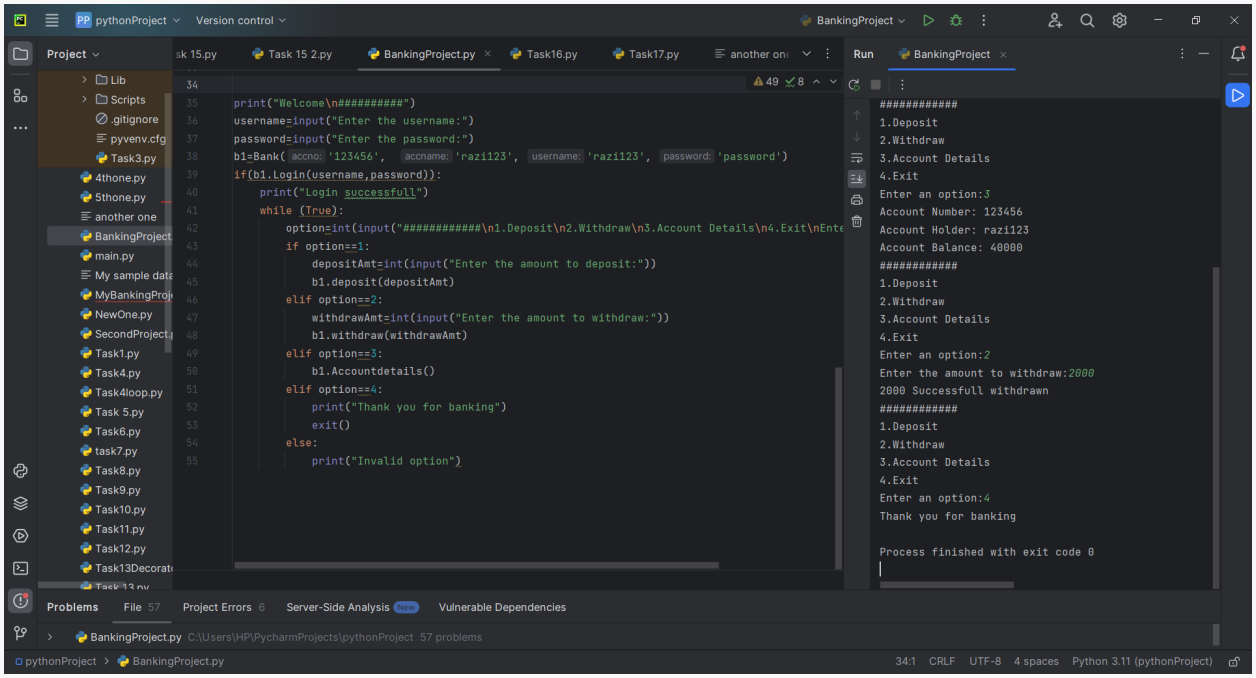
```python
class Bank:
    def __init__(self,accno,accname,username,password,balance=0):
        self.accNo=accno
        self.accholder=accname
        self.uname=username
        self.pwd=password
        self.balance=balance

    def Login(self,un,pw):
        if((self.uname==un)and(self.pwd==pw)):
            return True
        else:
            print("Login unsuccessfull")
    def deposit(self,depAmount):
        if(depAmount<0):
            print("Invalid amount")
        else:
            self.balance=self.balance+depAmount
            print(depAmount,"successfully deposited")

    def withdraw(self,withamount):
        if(withamount>self.balance):
            print("Insufficient balance")
        else:
```

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scrip
Welcome
##########
Enter the username:razi123
Enter the password:password
Login successfull
############
1.Deposit
2.Withdraw
3.Account Details
4.Exit
Enter an option:1
Enter the amount to deposit:40000
40000 successfully deposited
############
1.Deposit
2.Withdraw
3.Account Details
4.Exit
Enter an option:3
Account Number: 123456
Account Holder: razi123
Account Balance: 40000
############
1.Deposit
2.Withdraw
3.Account Details
```



```python
            self.balance=self.balance-withamount
            print(withamount,"Successfull withdrawn")

    def Accountdetails(self):
        print("Account Number:",self.accNo)
        print("Account Holder:",self.accholder)
        print("Account Balance:",self.balance)
        return


print("Welcome\n##########")
username=input("Enter the username:")
password=input("Enter the password:")
b1=Bank( accno: '123456',  accname: 'razi123',  username: 'razi123',  password: 'password')
if(b1.Login(username,password)):
    print("Login successfull")
    while (True):
        option=int(input("############\n1.Deposit\n2.Withdraw\n3.Account Details\n4.Exit\nEnte
        if option==1:
            depositAmt=int(input("Enter the amount to deposit:"))
            b1.deposit(depositAmt)
        elif option==2:
            withdrawAmt=int(input("Enter the amount to withdraw:"))
            b1.withdraw(withdrawAmt)
        elif option==3:
            b1.Accountdetails()
```

```
############
1.Deposit
2.Withdraw
3.Account Details
4.Exit
Enter an option:3
Account Number: 123456
Account Holder: razi123
Account Balance: 40000
############
1.Deposit
2.Withdraw
3.Account Details
4.Exit
Enter an option:2
Enter the amount to withdraw:2000
2000 Successfull withdrawn
############
1.Deposit
2.Withdraw
3.Account Details
4.Exit
Enter an option:4
Thank you for banking

Process finished with exit code 0
```



Task 18 Numpy

Task for class on 14-10-2023:

Numpy:

Exercise 1:

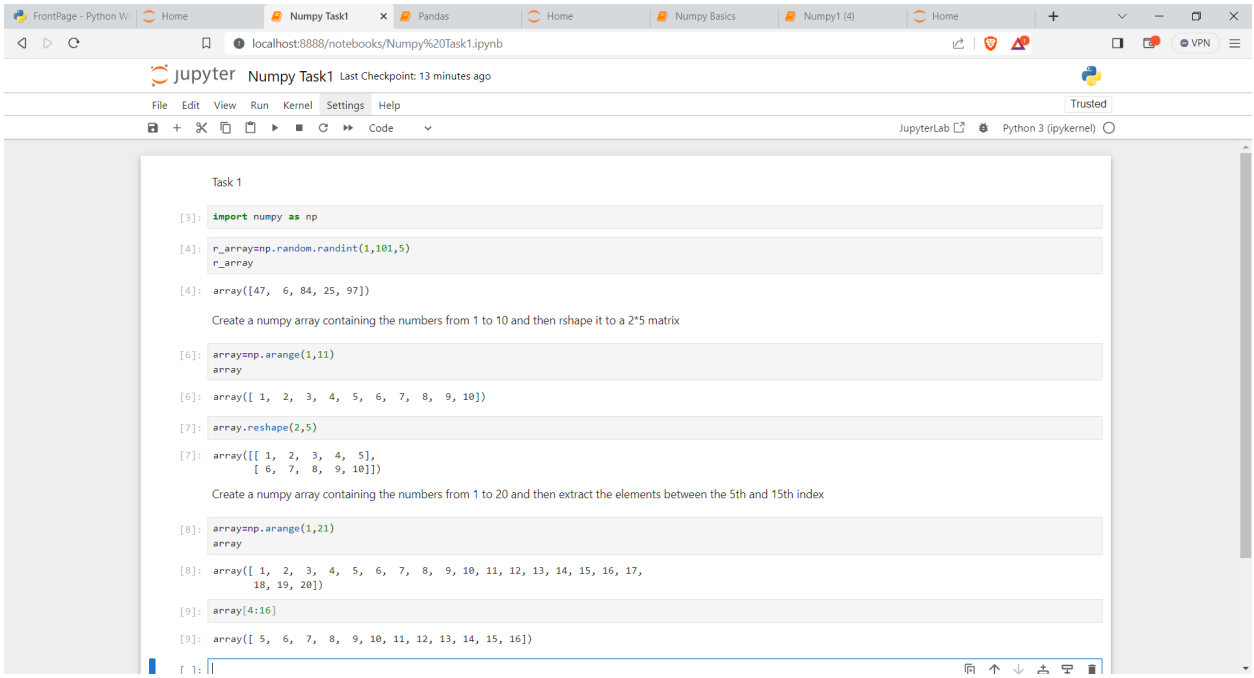
Create a numpy array with 5 elements consisting of random integers between 1 and 100.

Exercise 2:

Create a numpy array containing the numbers from 1 to 10, and then reshape it to a 2x5 matrix.

Exercise 3:

Create a numpy array containing the numbers from 1 to 20, and then extract the elements between the 5th and 15th index.



Task 19 Pandas

Task for class on 17-10-2022:

Pandas:

Series

Exercise 1:

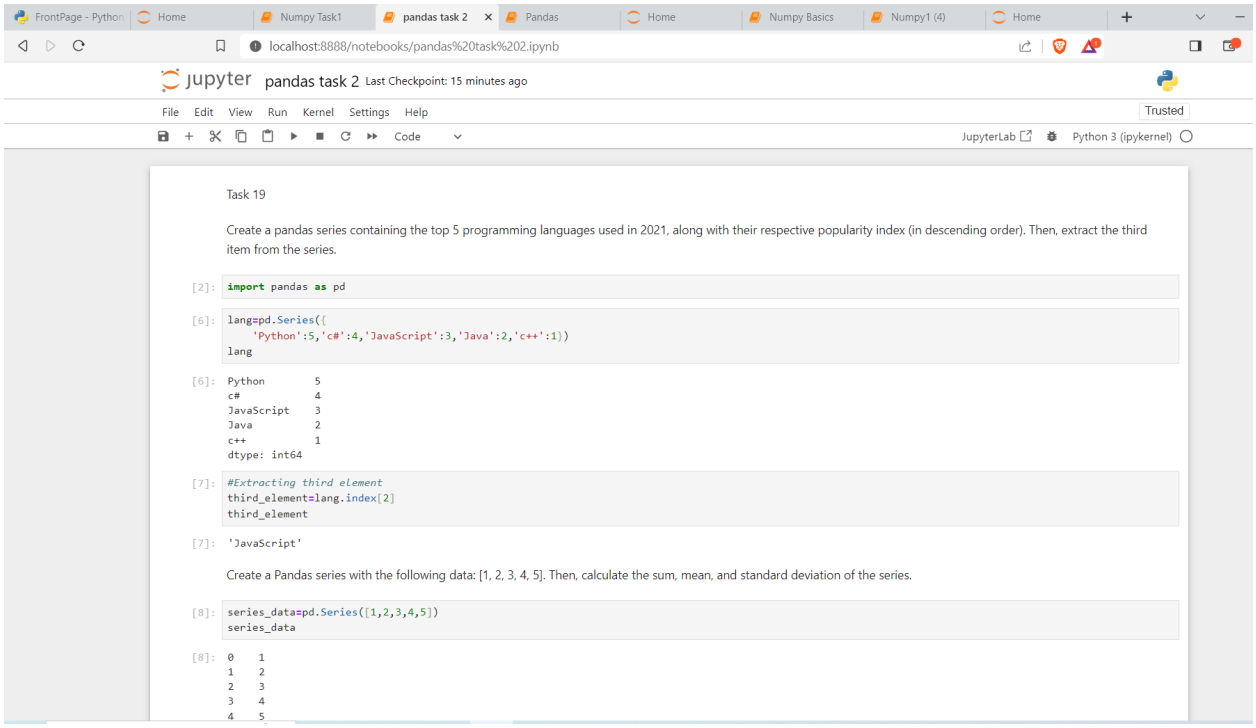
Create a pandas series containing the top 5 programming languages used in 2021, along with their respective popularity index (in descending order). Then, extract the third item from the series.

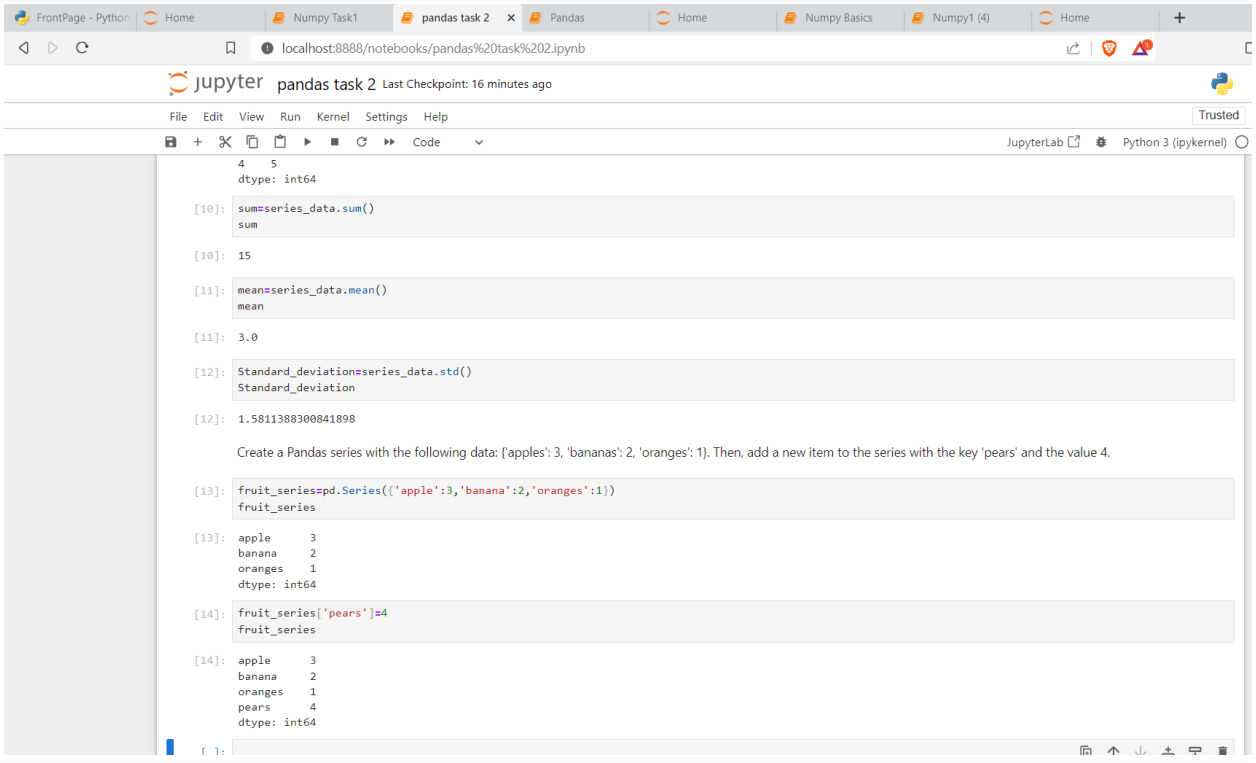
Exercise 2:

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

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.





Task 20 Data Frame

Tasks for class on 21-11-2023:

Dataframe

Exercise 1:

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

Exercise 2:

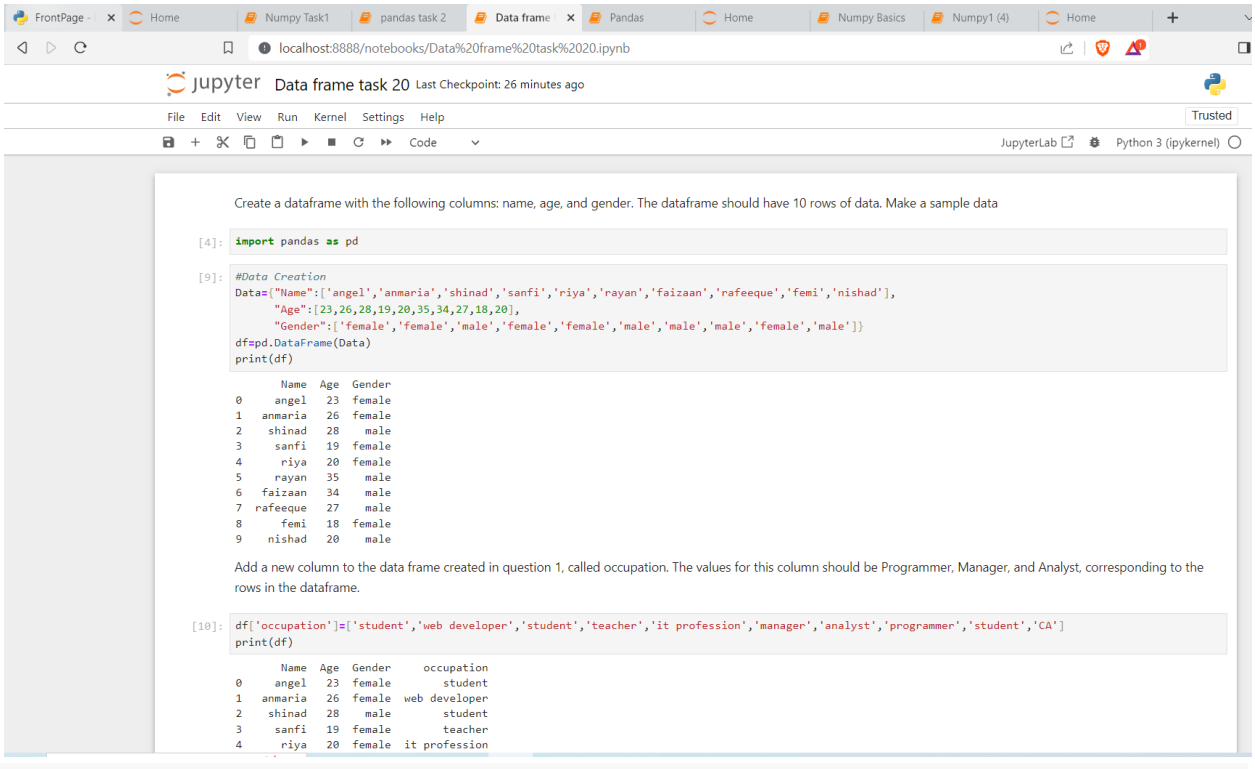
Add a new column to the data frame created in question 1, called occupation. The values for this column should be Programmer, Manager, and Analyst, corresponding to the rows in the dataframe.

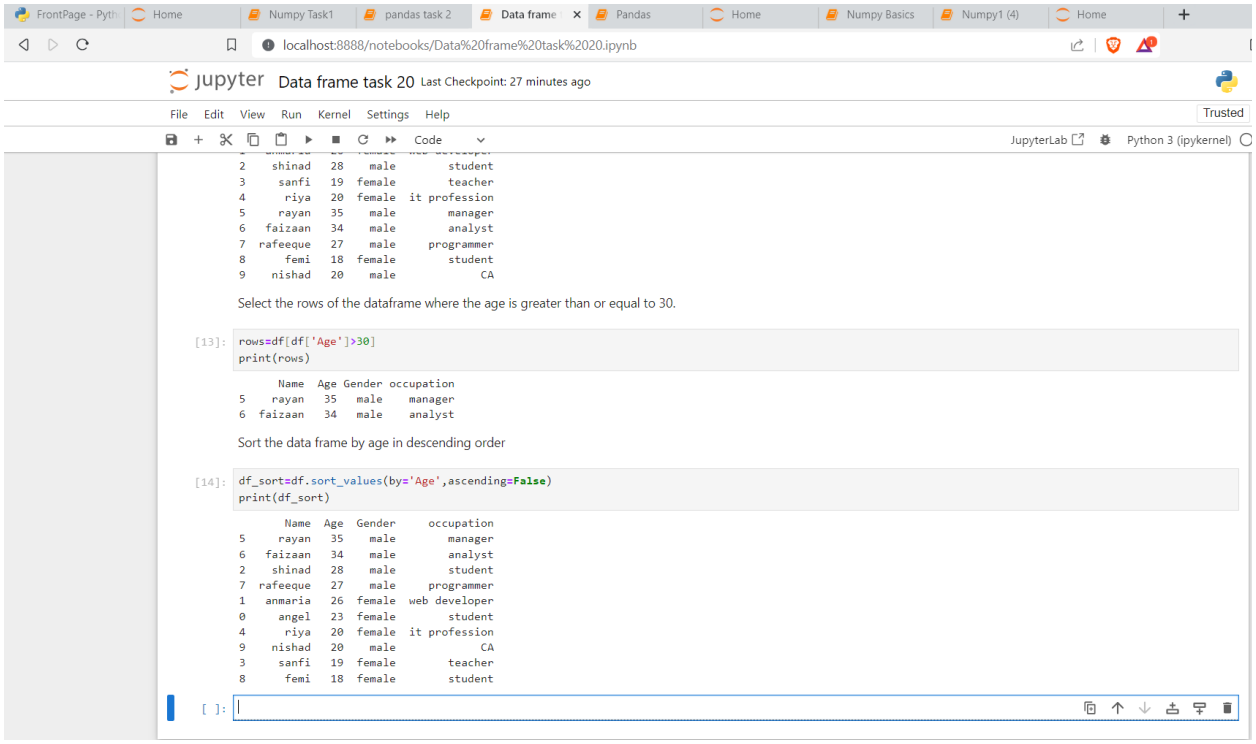
Exercise 3:

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

Exercise 4:

Sort the data frame by age in descending order





Task 21 Matplotlib

Task for class on 28-11-2023:

Matplotlib:

Exercise 1:

Create a line plot using matplotlib pyplot that displays the population of four different cities over time. Each city should have its own line, and the x-axis should represent years (e.g. 2010, 2011, 2012, etc.) while the y-axis should represent the population.

The data for the four cities is provided below:

City A: [500000, 550000, 600000, 650000, 700000, 750000, 800000]

City B: [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]

City C: [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]

City D: [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]

Exercise 2:

Create a scatter plot using matplotlib pyplot that shows the relationship between the number of hours studied and the test scores obtained by a group of students. Use the following data:

Hours Studied: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Test Scores: [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

Exercise 3:

Create a bar chart using matplotlib pyplot that shows the total sales for each month of the year. Use the following data:

Month: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]

Sales: [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346, 6675]

localhost:8888/notebooks/matplotlib.ipynb

jupyter matplotlib Last Checkpoint: 30 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

Matplotlib

[8]:
```
import matplotlib.pyplot as plt
```

[ ]:
```
#Exercise
Create a line plot using matplotlib pyplot that displays the population of four different cities over time. Each city should have its own line, and the x
The data for the four cities is provided below:
City A: [500000, 550000, 600000, 650000, 700000, 750000, 800000]
City B: [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]
City C: [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]
City D: [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]
```

[9]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[10]:
```
#x axis represent years
years=[2010,2011,2012,2013,2014,2015,2016]
#y axis represent population for city a,b,c,d
City_A=[500000, 550000, 600000, 650000, 700000, 750000, 800000]
City_B=[800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]
City_C=[1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]
City_D=[1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]
```

[11]:
```
plt.title('Population of cities over time')
plt.xlabel('year')
plt.ylabel('population')
plt.plot(years,City_A,label='City A')
plt.plot(years,City_B,label='City B')
plt.plot(years,City_C,label='City C')
plt.plot(years,City_D,label='City D')
plt.legend()
```

localhost:8888/notebooks/matplotlib.ipynb

jupyter matplotlib Last Checkpoint: 30 minutes ago

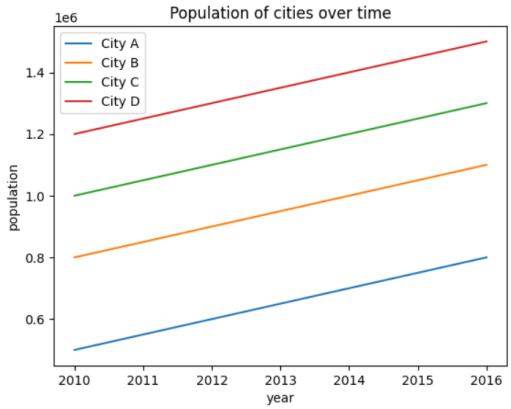
File Edit View Run Kernel Settings Help

Trusted

Code

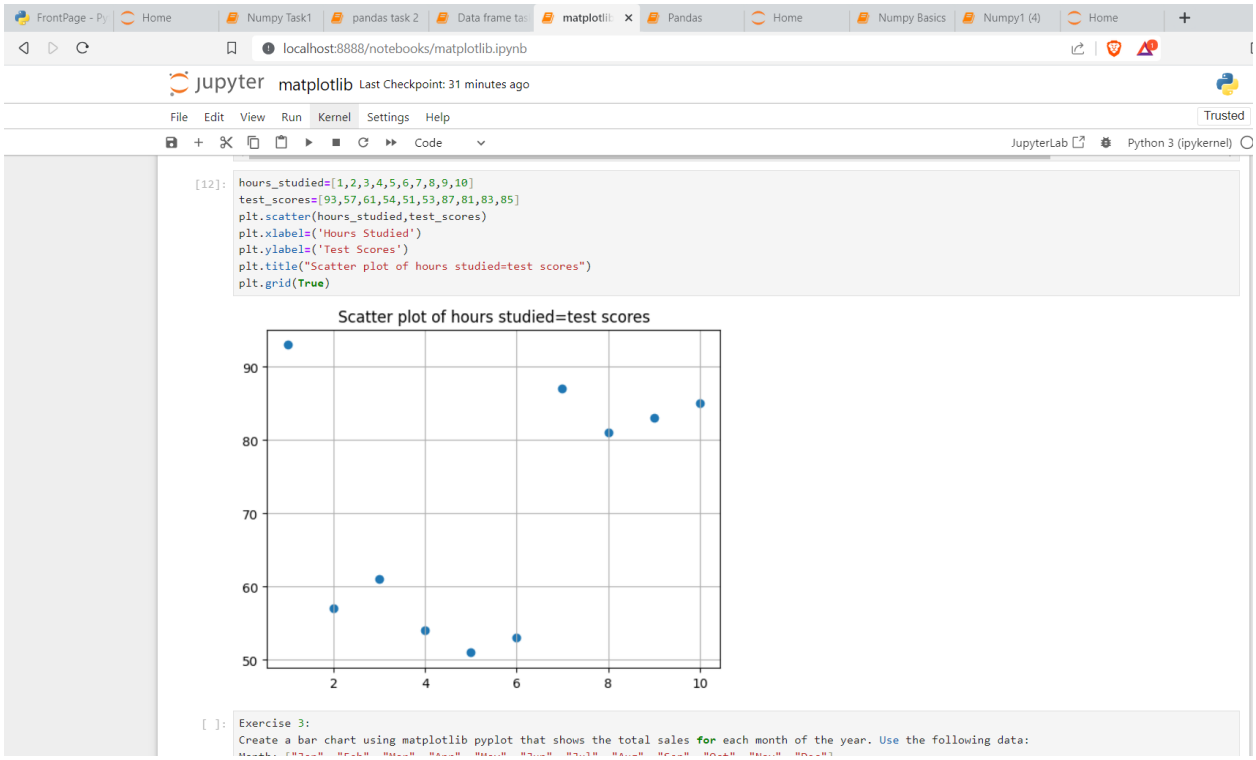
JupyterLab Python 3 (ipykernel)

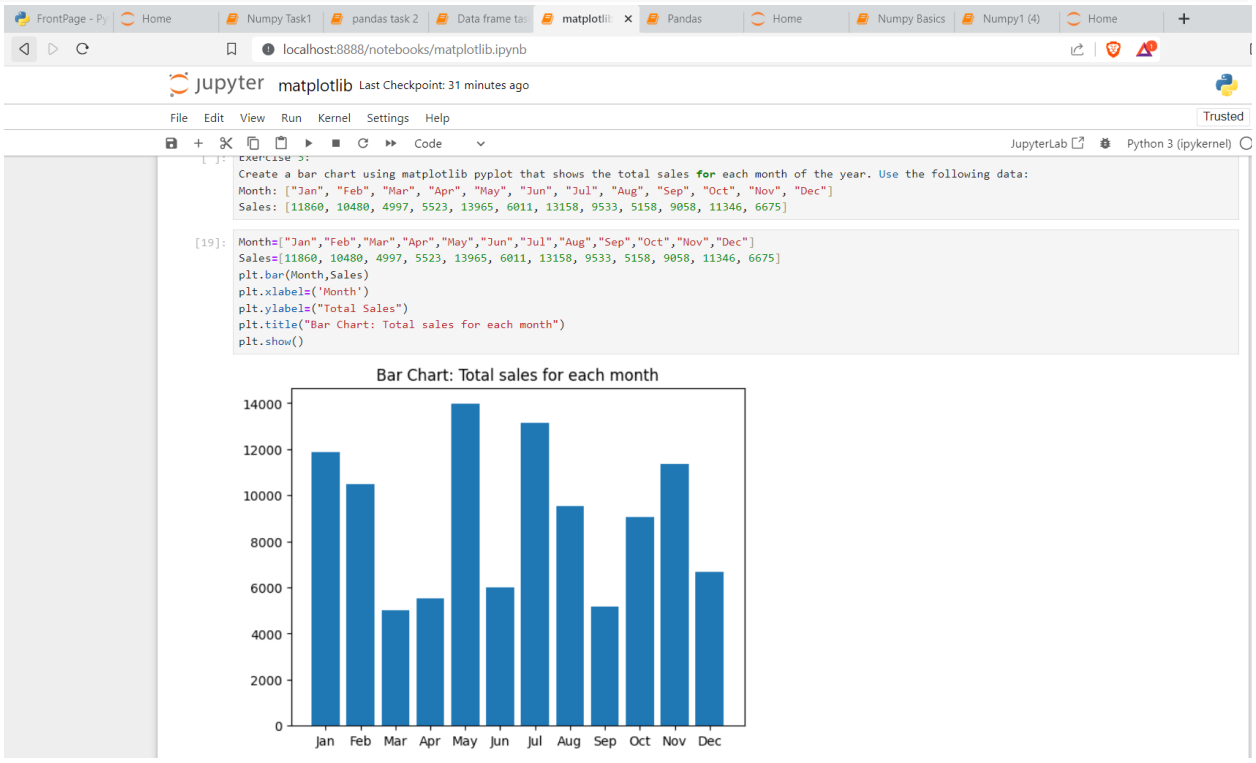
```
plt.legend()
plt.show()
```



[ ]:
```
Exercise 2:
Create a scatter plot using matplotlib pyplot that shows the relationship between the number of hours studied and the test scores obtained by a group of
Hours Studied: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Test Scores: [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]
```

[12]:
```
hours_studied=[1,2,3,4,5,6,7,8,9,10]
```





Task 22 Matplotlib and Seaborn

Task for class on 05-12-2023:

Matplotlib and Seaborn:

Using the below dataset of fifa players dataset. Perform Exploratory data analysis and find the following insights:
1.Which country has the most number of players
2.Plot a bar chart of 5 top countries with most number of players
3.Which player has the highest salary?
4.Plot a histogram to get the salary range of the players.
5.Who is the tallest player in the fifa players?
6.Which club has most number of players?
7.Which foot is most preferred by the players?Draw a bar chart for preferred foot.
https://drive.google.com/file/d/10oyIT1KPdwUqeU9-2LX0xE5-ZytNn9su/view?usp=share_link

FrontPage - Python Wiki | Home | Matplotlib and seaborn | fifa_data.csv - Google Drive | New Tab | fifa_data.csv - Google Drive

localhost:8888/notebooks/Matplotlib%20and%20seaborn.ipynb

jupyter Matplotlib and seaborn Last Checkpoint: 23 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[10]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

[18]:
```
fifa_data=pd.read_csv("C:\\jupyternotebook\\fifa_data.csv")
fifa_data
```

[18]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	94	FC Barcelona
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	94	Juventus
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	93	Paris Saint-Germain
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	93	Manchester United
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	92	Manchester City
...	...	...	...	...	...	...	...	...	...	...
18202	18202	238813	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png	England	https://cdn.sofifa.org/flags/14.png	47	65	Crewe Alexandra
18203	18203	243165	N. Christoffersson	19	https://cdn.sofifa.org/players/4/19/243165.png	Sweden	https://cdn.sofifa.org/flags/46.png	47	63	Trelleborgs FF
18204	18204	241638	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png	England	https://cdn.sofifa.org/flags/14.png	47	67	Cambridge United
18205	18205	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	England	https://cdn.sofifa.org/flags/14.png	47	66	Tranmere Rovers

FrontPage - Python Wiki | Home | Matplotlib and seaborn | fifa_data.csv - Google Drive | New Tab | fifa_data.csv - Google Drive

localhost:8888/notebooks/Matplotlib%20and%20seaborn.ipynb

jupyter Matplotlib and seaborn Last Checkpoint: 23 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

Which country has the most numbers of players

[19]:
```
Most_players=fifa_data["Nationality"].value_counts().idxmax()
Most_players
```

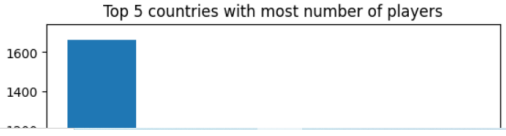
[19]:
```
'England'
```

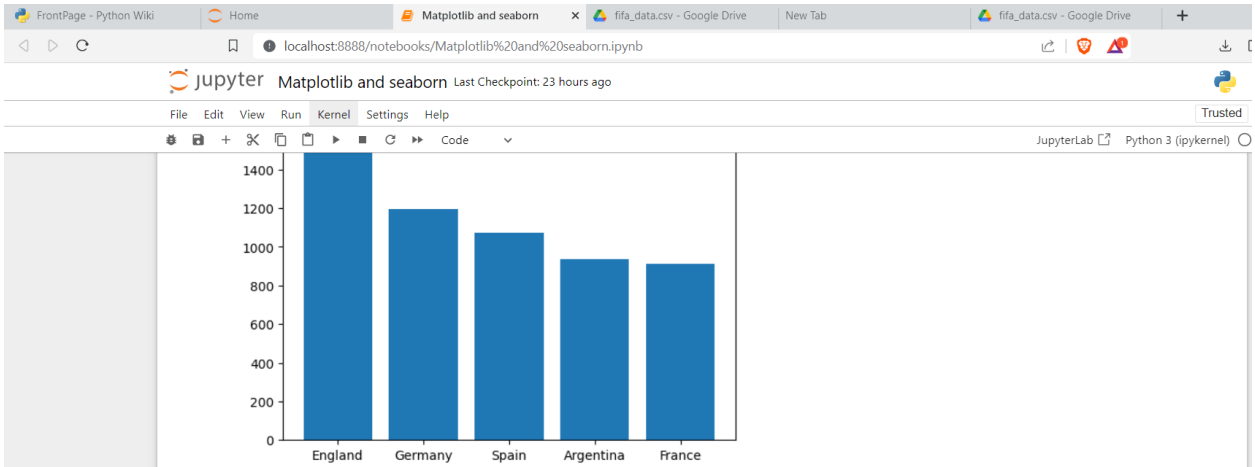
Plot a bar chart of 5 top countries with most numbers of players

[20]:
```
top5_countries=fifa_data['Nationality'].value_counts().head()
top5_countries
```

[20]:
```
Nationality
England      1662
Germany      1198
Spain        1072
Argentina     937
France        914
Name: count, dtype: int64
```

[21]:
```
top_countries=['England','Germany','Spain','Argentina','France']
players=[1662,1198,1072,937,914]
plt.bar(top_countries,players)
plt.title('Top 5 countries with most number of players')
plt.xlabel('Top 5 countries')
plt.ylabel('No: of players')
plt.show()
```





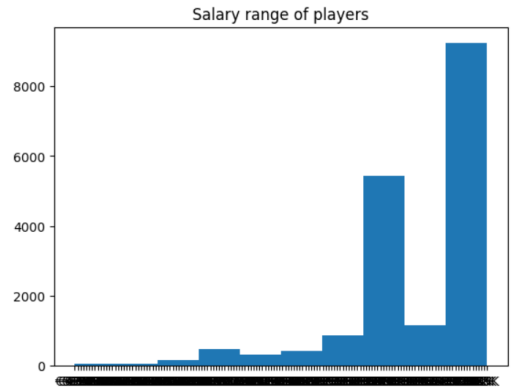
Which player has the highest salary?

[24]:
```
highsalary_player=fifa_data.loc[fifa_data['Wage'].idxmax()]['Name']
highsalary_player
```

[24]:
```
'F. Marchetti'
```

Plot a histogram to get the salary range of players

[25]:
```
Wage=fifa_data['Wage']
plt.hist(Wage)
plt.title('Salary range of players')
plt.xlabel('Wage')
plt.ylabel('Number of players')
plt.show()
```



Who is the tallest player in the fifa players?

[26]:
```
players_height=fifa_data['Height'].copy()
players_height
```

[26]:
```
0        5'7
1        6'2
2        5'9
3        6'4
4       5'11
        ...
```

```
3         6'4
4        5'11
        ...
18202     5'9
18203     6'3
18204     5'8
18205    5'10
18206    5'10
Name: Height, Length: 18207, dtype: object
```

[28]:
```
def feetto_cm(height):
    if isinstance(height,str):
        feet,inches=map(int,height.split("'"))
        return feet*12+inches
    else:
        return height
fifa_data['Height(inches)']=fifa_data['Height'].apply(feetto_cm)
fifa_data[['Height','Height(inches)']]
```

[28]:

	Height	Height(inches)
0	5'7	67.0
1	6'2	74.0
2	5'9	69.0
3	6'4	76.0
4	5'11	71.0
...	...	...
18202	5'9	69.0
18203	6'3	75.0
18204	5'8	68.0
18205	5'10	70.0
18205	5'10	70.0
18206	5'10	70.0

18207 rows × 2 columns

To find the tallest player

[30]:
```
tallest_player=fifa_data.loc[fifa_data['Height(inches)'].idxmax()]
tallest_player[["Name","Height"]]
```

[30]:
```
Name      T. Holý
Height        6'9
Name: 11614, dtype: object
```

Which club has most number of players?

[31]:
```
most_num_players=fifa_data['Club'].value_counts().max()
club_most_no_players=fifa_data['Club'].value_counts()[fifa_data['Club'].value_counts()==most_num_players]
print(club_most_no_players)
```

```
Club
FC Barcelona               33
Valencia CF                33
Fortuna Düsseldorf         33
Cardiff City               33
Rayo Vallecano             33
CD Leganés                 33
Frosinone                  33
Newcastle United           33
Southampton                33
Burnley                    33
Eintracht Frankfurt        33
Wolverhampton Wanderers    33
TSG 1899 Hoffenheim        33
Everton                    33
AS Monaco                  33
```

```
Frosinone                 33
Newcastle United          33
Southampton               33
Burnley                   33
Eintracht Frankfurt       33
Wolverhampton Wanderers   33
TSG 1899 Hoffenheim       33
Everton                   33
AS Monaco                 33
RC Celta                  33
Empoli                    33
Manchester City           33
Manchester United         33
Borussia Dortmund         33
Real Madrid               33
Atlético Madrid           33
Tottenham Hotspur         33
Chelsea                   33
Liverpool                 33
Arsenal                   33
Name: count, dtype: int64
```

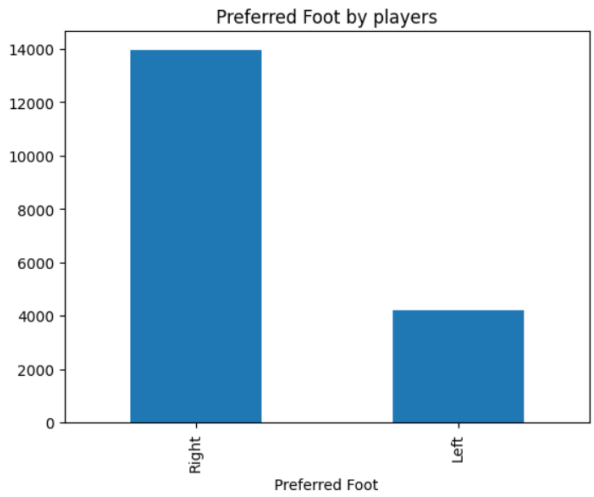
Whcih foot is most preferred by hte players ? Draw a bar chart for preferred foot.

[32]:
```
preferred_foot=fifa_data.value_counts('Preferred Foot')
preferred_foot
```

[32]:
```
Preferred Foot
Right    13948
Left      4211
Name: count, dtype: int64
```

[37]:
```
preferred_foot.plot.bar()
plt.title('Preferred Foot by players')
plt.xlabel=('Preferred foot')
plt.ylabel=('Number of players')
plt.show()
```

```
plt.ylabel=('Number of players')
plt.show()
```



[ ]: