Python Interview Questions:

Follow me in LinkedIn: @KEDAM MAHESH BABU

Python Programming Language:

1) Difference between a List and a Tuple?

List	Tuple
Lists are created using square brackets to store multiple values in a single variable. Ex: v=[‘bat’,’ball’,’stick’]	Tuples are written with round brackets to store multiple values in a single variable. Ex: v=(‘bat’,’ball’,’stick’)
List items are ordered, changeable, and allow duplicate values.	A tuple is a collection that is ordered and unchangeable.
Lists are slower than tuples.	Tuples are faster than lists.

2) Define Lambda Function?

A) A lambda function can only have one expression but may contain any number of arguments.

Syntax: lambda arguments: expression

```
Example: y=lambda b: b+10
         print(y(3))
```

3) Difference between Append and Extend methods in List?

Append	Extend
Append() method adds an argument at the end of the list as a single element.	Extend() method iterates over its argument adding each element to the list, and extending the list.
Ex: list1 = [‘tomato’,’potato’,’carrot’] list1.append(‘cucumber’) Output: list1 = [‘tomato’,’potato’,’carrot’, ‘cucumber’]	Ex: list1 = [‘tomato’,’potato’,’carrot’] list1.extend(‘toy’) Output: list1 = [‘tomato’,’potato’,’carrot’,’t’,’o’,’y’]

4) Difference between Pass, Continue and Break?

Pass	if you for some reason have a for loop with no content, put in the pass statement to avoid getting an error.	for x in [0, 1, 2]:
		  pass

Continue	Continue statement stops the current iteration of the loop, and continues with the next.	fruits = ["apple", "banana", "cherry"]
		for x in fruits:
		  if x == "banana":
		    continue

		print(x)

Break	Break statement stops the loop before it has looped through all the items.	fruits = ["apple", "banana", "cherry"]
		for x in fruits:
		  print(x)
		  if x == "banana":
		    break

5) Can you tell me the difference between List and Dictionary Comprehension?

List comprehension	Dictionary comprehension
Lists comprehension helps you create a new list based on the values of an existing list.	Dictionary comprehension is an elegant and concise way to create dictionaries. A dictionary comprehension takes the form {key: value for (key, value) in iterable}

y = ["cars", "bikes", "airplanes"]
newlist = []
 	
for x in y:
  if "a" in x:
    newlist.append(x)
 	
print(newlist)
	keys = ['a','b','c','d','e']
	values = [1,2,3,4,5]
	
	myDict = { k:v for (k,v) in zip(keys, values)}
	
	Print(myDict)
	
	output:{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

6) How do we drop null values in a Pandas dataframe?

A) Pandas DataFrame contains a dropna() function to remove rows and columns with Null/NaN values. This function returns a new DataFrame and the source DataFrame remains unchanged.

dropna(self, axis=0, how="any", thresh=None, subset=None, inplace=False)

Ex: df1 = df.dropna()

7) How would you replace all NaN values in a Pandas dataframe?

A) Pandas is one of those packages and makes importing and analyzing data much easier. Sometimes a csv file may contain null values, which are later displayed as NaN in Dataframe. So we use fillna() to let the user replace NaN values with some value of their own.

DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs)

Ex: df["column_name"].fillna("value", inplace = True)

8) How do you find the square root of a number using python?

A) Python has a sqrt() inbuilt function that returns the square root of any number.

Ex: import math

```
# print the square root of  4

    print(math.sqrt(4))
```

9) How do we clean a data set in Python?

A) Python offers us two libraries(Pandas, Numpy) to clean our data sets.

It is important to be able to deal with messy data, whether that means missing values, inconsistent formatting, malformed records, or nonsensical outliers.

Cleaning a dataset includes:

- Dropping unnecessary columns in a DataFrame
- Changing the index of a DataFrame
- Using .str() methods to clean columns
- Using the DataFrame.applymap() function to clean the entire dataset, element-wise
- Renaming columns to a more recognizable set of labels
- Skipping unnecessary rows in a CSV file

10) How do you print six random integers between 5 and 25 using NumPy?

A) we use the random.randint() function to generate random numbers using NumPy.

Ex: Import NumPy as np

```
    rand_r=np.random.randint(5,25,5)

    Print(rand_r)
```

11) How do we plot a sine graph using NumPy and Matplotlib Library?

```
Import NumPy as np
Import matplotlib.pyplot as plt
%matplotlib inline
x=np.arange(0,2*np.pi,0.1)
y=np.sin(x)
print(x)

plt.plot(x,y)
plt.show()
```

12) From a dataset which contains an ‘Age’ column, How do you fetch the subset of data for Age<25?

A: To subset the data frame we write the code as :

```
new_df=df[(df[Age]<25)]

Print(new_df)
```

13) How do we

Display the head of a dataset

Return the summary statistics

Select 2 different columns from a dataframe

Add a column to Pandas dataframe?

Display the head of a dataset	dataset_name.head()
Return the summary statistics	dataset_name.describe
Select 2 different columns from a dataframe	dataframe[[‘column1’,’column2’]]
Add a column to Pandas dataframe	ex: phone_no=[123,456,789] dataframe[‘phone_no’]=phone_no
create a Pandas data frame.	df=pd.dataframe(dataset_name)

14) Difference between global and local variables?

A) Global Variables: Global Variables are defined and declared outside a function, and we need to use them inside a function.

Local variables: A variable declared inside the function’s body or the local scope is known as a local variable.

15) What all python libraries can we use for Visualization?

A) Matplotlib: It is the standard data visualization library useful to generate two-dimensional graphs. It helps us plot histograms, pie charts, bar or column graphs, scatterplots, and non-Cartesian coordinates graphs. Many libraries are built on top of Matplotlib, and its functions are used in the backend.

Seaborn: It works really well for Numpy and Pandas and it is built based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

16) Which of the following statements are used in Exception Handling in Python?

A) 1. Except, 2. Finally, 3. Try, 4. All of the above

Answer: All of the above.

17) Python is written in

- Java language
- C+ language
- C language
- None of the above

Answer: C language

18) Date time computations in Python can be handled using:

- Date Module
- Datetime Module
- Time Module
- Timedate Module

Answer: Datetime Module

19) What will be the output of the following code snippet?

```
def is_even(number):
  message =  f"{number} is an even number" if number % 2 == 0 else  f"{number} is an odd number"
 return message
print(is_even(54))
```

- 54 is an odd number
- 54 is an even number
- Number is an even number
- Number is an odd number

Answer: 54 is an even number

20) Define enumerate() function?

The enumerate() function assigns an index to each item in an iterable object as It makes it easier to keep track of the content of an iterable object.

Ex: List2=[‘ball’,’bat’,’helmet’]

e1=enumerate(List2)

print(e1)

Output : [(0, ‘ball’), (1, ’bat’), (2, ’helmet’)]

21) Explain how Map, Reduce and Filter Functions Work?

Map	Reduce	Filter
Applies the given function to all the iterables and returns a new list.	Applies the same operation to items of a sequence	Filters item out of a sequence
Applies the same function to each element of a sequence	Uses the result of operations as first param of next operation	Used to filter the given iterables (list, sets, tuple) with the help of another function passed as an argument to test all the elements to be true or false
Returns the modified list	Returns an item, not a list	Returns the filtered list
[1,2,3,4]->def f(x):return x*2->[2,4,6,8]	List[x,y,z] function,f()->reduce->f(f(x,y),z)	List[x,y,z] condition->filter->[x,y] if condition is true

22) Difference between range, xrange, and arange?

range	xrange	arange
It returns a Python list object, which is of integers.	It returns a range object.	It is a function in the Numpy library. It can return fractional values as well.

23)Difference between del(),clear(),remove() and pop()?

del()	clear()	remove()	pop()

It deletes with respect to the position of the value. It does not return which value is deleted. It also changes the index towards the right by decreasing one value.	It clears the list.	if you know which particular value to delete. We can use the remove() function.	by default removes the last element and also returns back which value is deleted. We can store this deleted return value in a variable and use it in the future.

24) What do you know about negative indexing?

In python, the index numbering starts from 0, for the second index is 1 and continues.

The numbering for the last index is negative and starts from -1, for the second last number the index is -2 and the sequence continues in that direction.

example='string'

s[0]=s, s[1]=t, s[-1]=g, s[-2]=n

25) Can we use the OOps concept in Python?

A) Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model. However, Python can be treated as a procedural as well as a structural language.

26) Define Inheritance with an example?

A) Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, and makes it easier to create and maintain an application. The class from which we are inheriting is called superclass/parent class and the class that is inherited is called a derived / child class.

Different types of inheritance:

Single Inheritance – where a derived class acquires the members of a single super class.

Multilevel inheritance – a derived class d1 is inherited from base class base1, and d2 is inherited from base2.

Hierarchical inheritance – from one base class you can inherit any number of child classes.

Multiple inheritance – a derived class is inherited from more than one base class.

27) What is Polymorphism in Python?

A) Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.

28) What is Encapsulation in Python?

A) Encapsulation means binding the code and the data together. A Python class in an example of encapsulation.

29) How do we create a class in Python?

A) Class in Python is created using the class keyword.

Example:

class Employee:	
	def __init__(self, name):
	self.name = name
	E1=Employee("python")
	print(E1.name)
	Output: python

30) How do you do data abstraction in Python?

A) Data Abstraction is providing only the required details and hides the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

31) What Is Init Method In Python?

The init method works similarly to the constructors in Java. The method is run as soon as an object is instantiated. It is useful for initializing any attributes or default behavior of the object at the time of instantiation.

For example:

```
class InterviewbitEmployee:

# init method / constructor

def __init__(self, emp_name):

        self.emp_name = emp_name

# introduce method

def introduce(self):

      print('Hello, I am ', self.emp_name)

emp = InterviewbitEmployee('Mr Employee')

 emp.introduce()
```

32) How Does Python Handle Memory Management?

A) Python uses private heaps to maintain its memory. So the heap holds all the Python objects and the data structures. This area is only accessible to the Python interpreter; programmers can’t use it.

And it’s the Python memory manager that handles the Private heap. It does the required allocation of the memory for Python objects

Python employs a built-in garbage collector, which salvages all the unused memory and offloads it to the heap space.

33) What Are Errors And Exceptions In Python Programs?

A) Errors are coding issues in a program that may cause it to exit abnormally.

On the contrary, exceptions happen due to the occurrence of an external event that interrupts the normal flow of the program.

34) How Do You Handle Exceptions With Try/Except/Finally In Python?

A) Python lay down Try, Except, Finally constructs to handle errors as well as Exceptions. We enclose the unsafe code indented under the try block. And we can keep our fall-back code inside the except block. Any instructions intended for execution last should come under the finally block.

try:	
	print("Executing code in the try block")
	print(exception)
	except:
	print("Entering in the except block")
	finally:
	print("Reached to the final block")
	The output is:
	Executing code in the try block
	Entering in the except block
	Reached to the final block

35) What Are Python Iterators?

Iterators in Python are array-like objects which allow moving to the next element. We use them in traversing a loop, for example, in a “for” loop.

Python library has a no. of iterators. For example, a list is also an iterator and we can start a for loop over it.

36) What Is The Difference Between An Iterator And Iterable?

A) The collection type like a list, tuple, dictionary and set are all iterable objects whereas they are also iterable containers that return an iterator while traversing.

37) What Are Python Generators?

A) A Generator is a kind of function which lets us specify a function that acts like an iterator and hence can get used in a “for” loop.

In a generator function, the yield keyword substitutes the return statement.

```
 # Simple Python function

Def fn():

  return "Simple Python function."

 # Python Generator function

 def generate():

 Yield ‘Python Generator function’

 print(next(generate()))
```

38) What are decorators in Python?

A) Python decorator gives us the ability to add new behavior to the given objects dynamically. In the example below, we’ve written a simple example to display a message pre and post the execution of a function.

39) What is a Try Block?

A) A block that is preceded by the try keyword is known as a try block

try{	
	//statements that may cause an exception
	}

40) Write a program to count the number of capital letters in a file.

```
A) with open(SOME_LARGE_FILE) as countletter:

  count=0

  text = countletter.read()

  for character in text:

  if character.isupper():

  count += 1
```

41) Write a program to display the Fibonacci sequence in Python.

n = 10	
	# first two terms
	n0 = 0
	n1 = 1

```
#Count
x = 0
# check if the number of terms is valid
if n <= 0:
  print("Enter positive integer")
elif n == 1:
  print("Numbers in Fibonacci sequence upto",n,":")
  print(n0)
else:
  print("Numbers in Fibonacci sequence upto",n,":")
  while x < n:
      print(n0,end=', ')
      nth = n0 + n1
      n0 = n1
      n1 = nth
      x += 1
Output:


0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

42) Write a program in Python to produce a Star triangle?

```
def pattern🖓:
     k = 2 * n - 2
     for i in range(0,n):
          for j in range(0,k):
              print(end=" ")
          k = k - 1
          for j in range(0, i+1):
               print("*", end=" ")
          print("\r")
pattern(5)
```

43) Write a program to check whether the given number is prime or not?

```
A) num = 11
# If given number is greater than 1
if num > 1:
```

```
    # Iterate from 2 to n / 2
    for i in range(2, int(num/2)+1):
        # If num is divisible by any number between
        # 2 and n / 2, it is not prime
        if (num % i) == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

44) Write Python code to check if the given sequence is a palindrome or not?

```
A) def isPalindrome(s):
    return s == s[::-1]
 # Driver code
s = "malayalam"
ans = isPalindrome(s)
 if ans:
    print("Yes")
else:
    print("No")
```

45) Write Python code to sort a numerical dataset?

```
A) list = [ "13", "16", "1", "5" , "8"]
list = [int(x) for x in the list]
list.sort()
```

```
print(list)
```

Output : 1, 5, 8, 13, 16

46) Write a Python Function Which Takes A Variable Number Of arguments.

S) A function that takes variable arguments is called a function prototype.

Syntax:

```
def function_name(*arg_list)
```

Example:

```
def func(*var): for i in var:

print(i) func(1)

func(20,1,6)
```

The * in the function argument represents variable arguments in the function.

47) Write a Program to add two integers>0 without using the plus operator.

A) We can use bitwise operators to achieve this.

```
def add_nums(num1, num2):
    while num2 != 0:
       data = num1 & num2
      num1 = num1 ^ num2
      num2 = data << 1
      return num1
print(add_nums(2, 10))
```

48) Write a Program To Combine Two Different Dictionaries. While combining, if you find the same keys, you can add the values of these same keys. Output the new dictionary.

A) We can use the Counter method from the collections module

```
from collections import Counter
d1 = {'key1': 50, 'key2': 100, 'key3':200}
d2 = {'key1': 200, 'key2': 100, 'key4':300}
new_dict = Counter(d1) + Counter(d2)
print(new_dict)
```

49) Write a Program to match a string that has the letter ‘a’ followed by 4 to 8 'b’s.

A) We can use the re module of python to perform regex pattern comparison here.

```
import re
def match_text(txt_data):
```

```
pattern = 'ab{4,8}'

if re.search(pattern, txt_data): #search for pattern in txt_data

return 'Match found'

else:

return('Match not found')

print(match_text("abc")) #prints Match not found

print(match_text("aabbbbbc")) #prints Match found
```

50) How Will Access the dataset publicly shared spreadsheet in CSV format stored in Google Drive?

A) We can use the StringIO module from the io module to read from the Google Drive link and then we can use the pandas library using the obtained data source.

```
 from io import StringIO

import pandas

csv_link = "https://docs.google.com/spreadsheets/d/..."

data_source = StringIO.StringIO(requests.get(csv_link).content))

 dataframe = pd.read_csv(data_source)

print(dataframe.head())
```

51) Write Program convert date from yyyy-mm-dd format to dd-mm-yyyy format.

A) We can again use the re module to convert the date string as shown below:

```
 import re

def transform_date_format(date):

return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', date)

 date_input = "2021-08-01"

print(transform_date_format(date_input))
```

You can also use the datetime module as shown below:

```
from datetime import datetime

new_date = datetime.strptime("2021-08-01", "%Y-%m-%d").strftime("%d:%m:%Y") print(new_data)
```

52) Write Program for counting the Number Of Every Character of a given text file.

A) The idea is to use collections and pprint module as shown below:

```
 import collections

import pprint

with open("sample_file.txt", 'r') as data:

count_data = collections.Counter(data.read().upper())
```

```
count_value = pprint.pformat(count_data)
 print(count_value)
```

53) Write Program to solve the given equation assuming that a,b,c,m,n,o are constants:

A) ax + by = c

mx + ny = o

By solving the equation, we get:

```
a, b, c, m, n, o = 5, 9, 4, 7, 9, 4
temp = a*n - b*m
if n != 0:
x = (c*n - b*o) / temp
 y = (a*o - m*c) / temp
print(str(x), str👍)
```

54) WAP(Write A Program) which takes a sequence of numbers and check if all numbers are unique.

A) You can do this by converting the list to set by using set() method and comparing the length of this set with the length of the original list. If found equal, return True.

```
 def check_distinct(data_list):
if len(data_list) == len(set(data_list)):
return True
else:
return False;
print(check_distinct([1,6,5,8])) #Prints True
print(check_distinct([2,2,5,5,7,8])) #Prints False
```

55) How do we read CSV data into array NumPy?

A) This can be achieved by using the genfromtxt() method by setting the delimiter as a comma.

```
 from numpy import genfromtxt
csv_data = genfromtxt('sample_file.csv', delimiter=',')
```

56) How will you reverse the numpy array using one line of code?

A) This can be done as shown in the following:

```
reversed_array = arr[::-1]
```

where arr = original given array, reverse_array is the resultant after reversing all elements in the input.

57) What are some of the most commonly used built-in modules in Python?

A) Python modules are the files having python code which can be functions, variables or classes. These go by .py extension. The most commonly available built-in modules are:

- os
- math
- sys
- Random
- Re
- Date time
- JSON

58) How can you generate random numbers?

A) Python provides a module called random using which we can generate random numbers.

We have to import a random module and call the random() method as shown below:

The random() method generates float values lying between 0 and 1 randomly.

```
 import random
print(random.random())
```

To generate customized random numbers between specified ranges, we can use the randrange() method

Syntax: randrange(beginning,end,step)

For example:

```
import random
 print(random.randrange(5,100,2))
```

59) What is Pythonpath?

A)It is an environment variable used for incorporating additional directories during the import of a module or a package. PYTHONPATH is used for checking if the imported packages or modules are available in the existing directories. Not just that, the interpreter uses this environment variable to identify which module needs to be loaded.

60) Differentiate between deep and shallow copies?

A) Shallow copy does the task of creating new objects storing references of original elements. This does not undergo recursion to create copies of nested objects. It just copies the reference details of nested objects.

Deep copy creates an independent and new copy of an object and even copies all the nested objects of the original element recursively.

61) Are there any tools for identifying bugs and performing static analysis in python?

A) Yes, there are tools like PyChecker and Pylint which are used as static analysis and linting tools respectively. PyChecker helps find bugs in python source code files and raises alerts for code issues and their complexity.

Pylint checks for the module’s coding standards and supports different plugins to enable custom features to meet this requirement.

62) What Is the Main Function of Python? How do you invoke it?

In the world of programming languages, the main is considered as an entry point of execution for a program. But in python, it is known that the interpreter serially interprets the file line-by-line.This means that python doesn't provide main() function explicitly. But this doesn't mean that we cannot simulate the execution of main. This can be done by defining user-defined main() function and by using the

__name__ property of python file. This __name__ variable is a special built-in variable that points to the name of the current module. This can be done as shown below:

Example:

```
def main():

print("Hi Interviewbit!")

if __name__=="__main__":

main()
```

63) What is the difference between a Package And a Module In Python?

The module is a single python file. A module can import other modules (other python files) as objects. Whereas, a package is the folder/directory where different sub- packages and the modules reside.

A python module is created by saving a file with the extension of .py . This file will have classes and functions that are reusable in the code as well as across modules.

A python package is created by following the below steps:

Create a directory and give a valid name that represents its operation.

Place modules of one kind in this directory.

Create __init__.py file in this directory. This lets python know the directory we created is a package. The contents of this package can be imported across different modules in other packages to reuse the functionality.

64) How will you find the nearest value in a given numpy array?

A) We can use the argmin() method of numpy as shown below:

```
 import numpy as np

def find_nearest_value(arr, value):

arr = np.asarray(arr)

idx = (np.abs(arr - value)).argmin()

return arr[idx]

#Driver code

arr = np.array([ 0.21169, 0.61391, 0.6341, 0.0131, 0.16541, 0.5645, 0.5742])

value = 0.52
```

```
print(find_nearest_value(arr, value)) # Prints 0.5645
```

65). What do you understand by NumPy?

A) NumPy is one of the most popular, easy-to-use, versatile, open-source, python-based, general-purpose package that is used for processing arrays. NumPy is short for NUMerical PYthon. This is very famous for its highly optimized tools that result in high performance and powerful N-Dimensional array processing feature that is designed explicitly to work on complex arrays. Due to its popularity and powerful performance and its flexibility to perform various operations like trigonometric operations, algebraic and statistical computations, it is most commonly used in performing scientific computations and various broadcasting functions. The following image shows the applications of NumPy:

1) Arithmetic operations

2) Broadcasting

3) Bitwise Operators

4) Stacking

5) Matrix Operation

6) Searching, Sorting and Counting

7) Mathematical operation

8) Copying and viewing arrays

9) Statistical operations

10) Algebra

66) What are the steps to create 1D,2D and 3D arrays?

A) **1D array creation:**

```
import numpy as np

one_dimensional_list = [1,2,4]

one_dimensional_arr = np.array(one_dimensional_list)

print("1D array is : ",one_dimensional_arr)
```

2D array creation:

```
import numpy as np

 two_dimensional_list=[[1,2,3],[4,5,6]]

 two_dimensional_arr = np.array(two_dimensional_list)

 print("2D array is : ",two_dimensional_arr)
```

3D array creation:

```
import numpy as np

three_dimensional_list=[[[1,2,3],[4,5,6],[7,8,9]]]
```

```
 three_dimensional_arr = np.array(three_dimensional_list)

 print("3D array is : ",three_dimensional_arr)
```

ND array creation: This can be achieved by giving the ndmin attribute. The below example demonstrates the creation of a 6D array:

```
import numpy as np

ndArray = np.array([1, 2, 3, 4], ndmin=6)

print(ndArray)

print('Dimensions of array:', ndArray.ndim)
```

67) What is PIP in Python?

A) PIP stands for Python Installer Package. As the name indicates, it is used for installing different python modules. It is a command-line tool providing a seamless interface for installing different python modules. It searches over the internet for the package and installs them into the working directory without the need for any interaction with the user. The syntax for this is:

```
 pip install <package_name>
```

68) How To Add A New Column To Pandas Dataframe?

A) A new column can be added to a pandas dataframe as follows:

```
import pandas as pd

data_info = {'first' : pd.Series([1, 2, 3], index=['a', 'b', 'c']),

'second' : pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(data_info)

#To add new column third

df['third']=pd.Series([10,20,30],index=['a','b','c'])

print (df)

#To add new column fourth

df['fourth']=df['first']+info['third']

print (df)
```

69) How will you identify and deal with missing values in a dataframe?

A) We can identify if a dataframe has missing values by using the isnull() and isna() methods.

```
missing_data_count=df.isnull().sum()
```

We can handle missing values by either replacing the values in the column with 0 as

follows:

```
df[‘column_name’].fillna(0)
```

Or by replacing it with the mean value of the column

```
df[‘column_name’] = df[‘column_name’].fillna((df[‘column_name’].mean()))
```

70) How will you combine different pandas dataframes?

A) The data frames can be combined using the below approaches:

append() method: This is used to stack the data frames horizontally. Syntax: df1.append(df2)

concat() method: This is used to stack data frames vertically. This is best used when the data frames have the same columns and similar fields. Syntax: pd.concat([df1, df2])

join() method: This is used for extracting data from various data frames having one or more common columns.

```
df1.join(df2)
```

71) Explain Pandas?

A) Pandas is an open-source, python-based library used in data manipulation applications requiring high performance. The name is derived from “Panel Data” having multidimensional data. This was developed in 2008 by Wes McKinney and was developed for data analysis.

Pandas are useful in performing 5 major steps of data analysis - Load the data, clean/manipulate it, prepare it, model it, and analyze the data.

72) What is a pandas dataframe?

A data frame is a 2D mutable and tabular structure for representing data labeled with axes - rows and columns.

The syntax for creating dataframe:

```
import pandas as pd

dataframe = pd.DataFrame( data, index, columns, dtype)
```

where:

data - Represents various forms like series, map, ndarray, lists, dict etc.

index - Optional argument that represents an index to row labels.

columns - Optional argument for column labels.

Dtype - the data type of each column. Again optional.

73) How Will I Get the items that are not common to both The given series A and B?

A) We can achieve this by first performing the union of both series, then taking the intersection of both series. Then we follow the approach of getting items of union that are not there in the list of the intersection.

74) While Importing Data From Differentsources, can the pandas library recognize dates?

A) Yes, they can, but with some bit of help. We need to add the parse_dates argument while we are reading data from the sources. Consider an example where we read data from a CSV file, we may encounter different date-time formats that are not readable by the pandas library. In this case, pandas provide flexibility to build our custom date parser with the help of lambda functions as shown below:

```
import pandas as pd

from datetime import datetime
```

```
dateparser = lambda date_val: datetime.strptime(date_val, '%Y-%m-%d %H:%M:%S')
df = pd.read_csv("some_file.csv", parse_dates=['datetime_column'], date_parser=dateparser
```

Follow me in LinkedIn: @KEDAM MAHESH BABU