We at The Data Monk hold the vision to make sure everyone in the IT industry has an equal stand to work in an open domain such as analytics. Analytics is one domain where there is no formal under-graduation degree and which is achievable to anyone and everyone in the World.

We are a team of 30+ mentors who have worked in various product-based companies in India and abroad, and we have come up with this idea to provide study materials directed to help you crack any analytics interview.

Every one of us has been interviewing for at least the last 6 to 8 years for different positions like Data Scientist, Data Analysts, Business Analysts, Product Analysts, Data Engineers, and other senior roles. We understand the gap between having good knowledge and converting an interview to a top product-based company.

Rest assured that if you follow our different mediums like our blog cum questions-answer portal www.TheDataMonk.com , our youtube channel - The Data Monk, and our e-books, then you will have a very strong candidature in whichever interview you participate in.

There are many blogs that provide free study materials or questions on different analytical tools and technologies, but we concentrate mostly on the questions which are asked in an interview. We have a set of 100+ books which are available both on Amazon and on The Data Monk e-shop page

We would recommend you to explore our website, youtube channel, and e-books to understand the type of questions covered in our articles. We went for the question-answer approach both on our website as well as our e-books just because we feel that the best way to go from beginner to advance level is by practicing a lot of questions on the topic.

We have launched a series of 50 e-books on our website on all the popular as well as niche topics. Our range of material ranges from SQL, Python, and Machine Learning algorithms to ANN, CNN, PCA, etc.

We are constantly working on our product and will keep on updating it. It is very necessary to go through all the questions present in this book.

Give a rating to the book on Amazon, do provide your feedback and if you want to help us grow then please subscribe to our Youtube channel.

100 Python interview Questions

(Questions are based on practical and theory knowledge of python)

Q1. Write a sorting function without using the list.sort function (descending order)

```
Code: ls=[25,59,80,100,156,12]
          new_ls=[ ]
          while ls:
             min=ls[0]
             for x in ls:
                   if x>min:
                      min=x
             new_ls.append(min)
             ls.remove(min)
        print(new_ls)
Output: 156,100,80,59,25,12
```

- ls is the list containing random numbers
- new_ls contains empty list, apply while loop and make min as 0 and pick one element one after the other and compare with the previous number if the number is minimum put it into min and look for the other numbers which is less than the current minimum value.

Q2. Write a Python program to print set of duplicates in a list.

```
list=[1,2,3,1,5,6,2,5,7]
print(set[x for x in list if list.count(x)>1]))
Output: {1,2,5}
```

Q3. Given an array of n element write a python function to search a given element x in array

#Given an array of n elements write a python function to search a given element x in array

Code:
```
def search(array,z):
        for i in range(len(array)):
                if array[i] = = z:
                        return i #return the index number if number is found
        return -1 #if number is not [resent return -1
    array = [1,2,3,14,35,66,7,87,94]
    search(array, 35)
```

Output: 4

- Here we can use linear search on the given array
- Search z in the given array
- If z if found in the given array then print the index of the z and if the z is not present in the array return -1.

Q4. Write a Python program to extract digits from given string.

Code:
```
import re #re: Regular Expression
    string = ‘data1monk23’
    print(“The original string : “ +string)
    abc = re.sub(“\D”, “”, string)
    print(“The digits string is: “ +abc)
```

Output:
```
The original string : data1monk23
The original string is: 123
```

- Python has built-in regular expression (re) means if a given regular expression matches a particular string
- The re.sub() function is used to replace circumstance of a specific sub-string with another sub-string.
- \D is used to fetch the numbers from the string.

Q5. Write a program to check whether the email id provided in the code is valid or invalid.

- Validating an email means to check whether the given email id is in perfect format or not.
- And this code doesn’t tell about spam mail or fake mail this code tells about the format of email id is valid or invalid.

Code:
```
import re
def isValidEmail(email):
        regex= “^[A-Z0-9.%+-]+[A-Z0-9]+\.[A-Z]{2, }$”
        if len(email) > 7:
                if re.match(regex,email,re.IGNORECASE) is not
None:
                        return True
if isValidEmail(“datamonk@gmail.com”):
        print(“Valid Email”)
else:
        print(“Invalid email”)
```
Output: Valid Email

- When re is imported, we can start using the regular expression.
- A Regular expression or regex which has sequence of characters which makes search pattern.
- Python’s function re.match( ) searches for the regular expression patterns and it returns the first instance.

Q6. Explain Garbage collection in python briefly.

- Garbage can be cleaned manually that is reclaiming the memory this can be a traditional method. But modern languages like Python, Java etc. In which the language runtime environment continues using garbage collector, it will always check for memory location which are not in use, such memory locations are reclaimed.
- Python deletes unnecessary objects automatically to free memory space. This process is termed as garbage collection.
- Python's garbage collector runs during program execution and is triggered when an object reference count reaches zero.
- An objects reference count increases when it is assigned a new name or placed in a container (list, tuple or dictionary)
- An objects reference count decreases when it's deleted with del, it's reference is reassigned or it's reference goes out of scope.
- For example, recently created objects are more likely to be dead. As objects are created, the garbage collector assigns them to generations. Each object gets one generation, and younger generations are dealt with first.

Q7. What is multithreading achieved in python?

- Multithreading means continuously executing multiple threads by rapidly switching the control of the CPU between threads also known as context switching.
- Python has GIL that is Global Interpreter Lock which allows only one thread to execute one at a time. The process makes sure that a thread acquires the GIL, does a little work, then passes the GIL onto the next thread.
- This process happen very quickly that every human being thinks that the threads are executing working simultaneously or parallel but this is not the case it goes one by one by just taking turns using the same CPU core.
- Multithreading leads to maximum utilization of the CPU by multitasking. The advantage of multitasking in python would be: Multithreaded programs can run faster on computer systems with multiple CPUs, because theses threads can be executed truly concurrent.

Q8. Write Python logic to count the number of capital letters in a file.

```
Code: import os
        os.chdir('C\\Users\\Manali\\Desktop')
        with open("Para.txt") as para:
                count=0
                for i in para.read( ):
                        if i.isupper( ):
                                count+=1
        print(count)
```

- upper() is a built-in method used for string handling. The upper() methods returns the uppercased string from the given string. It converts all lowercase characters to uppercase.

Q9. **Can Python be used for web client and web server side programming? Which one is best suited to Python?**

- Python is more suitable for web server-side application development due to its numerous set of features for creating business logic, database interactions, web server hosting, etc.
- However, Python can be used as a web client-side application that needs some conversions for a browser to interpret the client-side logic.
- Also Python can be used to create desktop applications that can run as a standalone application such as utilities for test automation.

Q10. What are virtualenvs?

- Virtualenvs is the virtual environment, it is used to isolate a Python interpreter together with a set of libraries and settings. Together with pip, it allows us to develop, deploy and run multiple applications on a single host.

We can use virtualenvs in these following steps:

1. Open a terminal.
2. Setup the pip package manager.
3. Install the virtualenv package.
4. Create the virtual environment.
5. Activate the virtual environment.

Q11. What is pickling and unpickling?

- Pickling means convert object to stream (object can be list, dictionary or tuple) and again converting it in reverse manner that is stream to object is called unpickling.
- Pickling is also called serialization and unpickling is called unserialization
- File can be seen only after deserialization.
- Pickling has writable format and unpickling has readable format.
- Python provide predefined pickle method
- Pickling has predefined module called dump( ) and Unpickling has predefined module called load( )

Here is the example of pickling and unpickling:

```
import pickle
f=open('d://data.txt,'wb')
dct={"name:"Datamonk","age":23,"Gender":"F","marks":75}
pickle.dump(dct,f)
f.close( )
```

- wb means write binary file, as pickling has writable format
- As soon as you run the code, data.txt file is created and a serializable data is created, it not understood to the human till its unpickled, this was the example of pickling.
- Here is the example of unpickling:

  Code: import pickle

  ```
  f=open('d://data.txt,'rb')

  d=pickle.load(f)

  print(d)
  ```

- When you run the code the data is printed successfully.

Q12. How does Lambda function differ from a normal function in Python?

- Lambda is similar to the inline function in C programming. It returns a function object. It contains only one expression and can accept any number of arguments.
- In case of a normal function, you can define a function name, pass the parameter, and compulsorily have a return statement.
- But the Lambda function is typically used for simple operations without the use of function names. It can also be used in the place of a variable.

Q13. How can we know the total number of references pointing to a particular object?

- The system module has getrefcount() function which gives the number of references pointing to a particular object including its own reference.

  Code: import sys

  ```
  a="DataMonk"

  b=a
  ```

```
sys.getrefcount(a)
```

Output: 3

- Here the object DataMonk is referred by a,b and getrefcount() function itself. So the output is 3

Q14. What are *args and **kwargs in python?

- In Python, we can pass a variable number of arguments to a function using special symbols. When we are unsure about the number of arguments to be passed in a function we use *args and **kwargs
  There are two special symbols:

1. *args also known as Non Keyword Arguments: Python has *args which allow us to pass the variable number of non-keyword arguments to function.
2. **kwargs also known as Keyword Arguments: **kwargs works the same way, but for named parameters

```
Code: def func1(*args,**kwargs):
                print(args)
                print(kwargs)
        func1(1,'efgh','mona',year='second)
Output: (1, 'efgh','mona')
          {'year':'second}
```

Here 1, efgh, mona -> args

“year”=”second’ -> kwargs

Q15. What is the difference between Python 2 and Python 3?

1. Function print:

Python 2 had print “hello” that is, it didn’t had parenthesis while Python 3 has parenthesis in print statement print (“hello”)

2. Division of integers:

In Python 2 when two integers are divided, you always provide a integer value, while in Python 3 you get a floor value when two integers are divided.

3. Unicode:

In Python 2, to store Unicode string value, you require to define them with “u” and in Python 3, default storing of string is Unicode.

Q16. Explain Inheritance in Python with an example and does python support multiple inheritance?

- Inheritance is when a child class inherits the property or gene of parent class. From the programming point of view when a class inherits the property from other class. Parents class contains various attributes and method which can be used number of times.
- The class which inherits property is known as child class or the derived class and the other class is known as the parent class or the base class. Child class inherits every attribute and method which are present in parent class. Also new we can create new methods in child class if required.

Advantages of using Inheritance in Python:

- Error identification: If we use inheritance in python, the error identification becomes easy as the code is dividing into number of class.
- Code repetition: If we use inheritance in python, we can put all the main and standard attribute in the parent class which can be used by child class.
- Less maintenance cost: As inheritance allows the code to reuse, so it has less maintenance.

Let’s take a simple example to understand inheritance.

- We have taken two classes named it as parent and child and from the child class we are inheriting the properties of parent class.

```
class Parent:

        def function1(self):

                print(“This is parent function”)

class Child(Parent):

        def function2(self):

                print(“This is child function”)

ob=Child( )

ob.function1( )

ob.function2( )

Output: This is parent function

         This is child function
```

- A class can be derived from more than one parent classes is called multiple inheritance and yes python does support multiple inheritance.

Q17. What is the difference between NumPy and SciPy?

- NumPy and SciPy make easy to implement the concept of machine learning, data science with their various modules and packages.
- NumPy stands for Numerical Python while SciPy stands for Scientific Python

- Processing speed: NumPy has a faster processing speed than others while SciPy has slow speed as compared to numpy.
- NumPy does the basic operations like sorting, indexing and array manipulation while SciPy contains all algebraic functions.

Q18. What is python zip( ) function?

- The python zip( ) function is used to transform multiple list that is list1 list2 list3 to a single list of tuples and it returns a zip object.
- Syntax of zip( ): zip(*iterator1, iterator2, iterator3 ...*)

  a= (“Python”,”Java”,”C”)

  b=(“List”,”Map”,”Pointer”,”Dataset”)

  x=zip(a,b)

  print(tuple(x))

  Output: ((‘Python’,’List’),(‘Java’,’Map’),(‘C’,’Pointer’))

- Also unzipping coverting zipped values back to individual as they were.
- We can pass multiple iterators to zip function of same or different types.

Q19. What is web scrapping, how would you achieve web scrapping in python.

- Web scrapping is a method of extracting the large amount of information which is available on the website and saving it onto the local machine or onto the database tables.
- Some websites allow web scraping and some don’t therefore you should check weather the website is allowing you for getting their own data or do web scrapping.

  Python is good for web scrapping because it is:

1. Easy to use
2. Dynamically typed
3. Easily understandable syntax
4. Low maintenance
5. Cost effective

   In python following are some modules for web scraping:

   Urllib2, Scrappy, Pyquery, BeautifulSoup

Q20. What is PYTHONPATH?

- It is an environment variable which is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the presence of the imported modules in various directories. The interpreter uses it to determine which module to load.
- You can set PYTHONPATH permanently by:
- Go to the windows menu, right click on My computer and select property,then click advanced system settings on the left and then choose Environment variables button then click new button to make new0020user variable. Give the variable name as PYTHONPATH and then press ok to save the variable.
- And to confirm whether the environment variable is correctly set: Open a cmd and then type echo % PYTHONPATH%

Q 21. What is the difference between range and xrange?

1. Range( ): This function returns a range object

   Xrange( ): This function returns generator object

2. Syntax: range( ): range(start, stop, step)

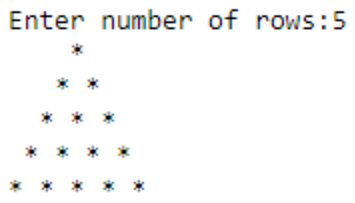
xrange( ): xrange(start, stop, step)

3. Memory: Variable storing the range created by range function requires more memory while variable storing the range using xrange( ).
4. Speed: xrange( ) returns only generator object containing only the values that are required by lazy evaluation, that is why xrange( ) is faster in speed.
5. Operations: As the return type of range( ) is range object, operations like addition multiplication, deletion can be performed on range( ), while these operations can't be applied on xrange( ) as the return type is generator object.

Q22. Write a program to print an equilateral triangle of stars.

Code: num=int(input("Enter number of rows: "))

```
for i in range (1, num+1):

print(" "*(num-i)+"* "*i)
```



```
Enter number of rows:5
    *
   * *
  * * *
 * * * *
* * * * *
```

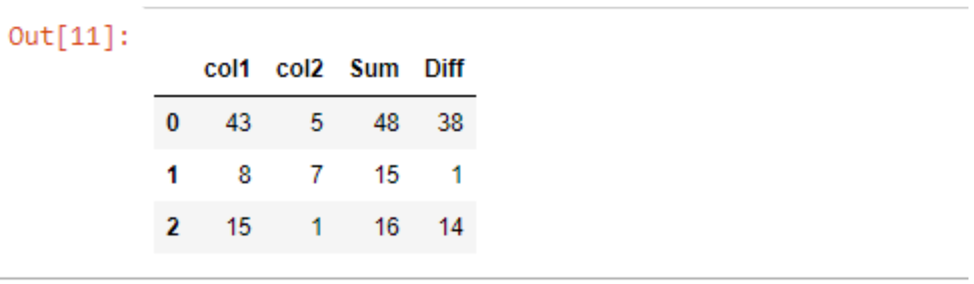
Q23. **How to create a new column in pandas by using values from other columns?**

```
import pandas as pd

a=[43,8,15]

b=[5,7,1]

d={"col1":a, "col2": b}

df.pd.DataFrame(d)

#Perform sum and addition using col1 and col2
```

df[“Sum”]=df[“col1”]+df[“col2”]

df[“Diff”]=df[“col1]-df[“col2]

df



```
Out[11]:
   col1  col2  Sum  Diff
0    43     5   48    38
1     8     7   15     1
2    15     1   16    14
```

- New column sum and difference is made from col1 and col2.

Q24. What are Literals in Python and explain about different Literals?

- Literals can be defined as data given to a variable or a value given to a particular variable.

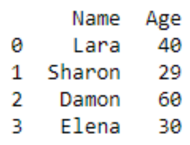
There are various kinds of literals:

1. Single literal: A string literal is collection of zero or more words or characters enclosed in single quotation marks. There can be single, double and triple strings. The following is the example of string literal: ‘The Data Monk’ ’19 Aug 2021’
2. Numeric Literals: Numeric literal is a character string which is collection of digits 0-9, the decimal point and the single character plus sign or minus sign. A literal must contain at least one digit also it must not contain more than one decimal point.
3. Boolean Literals: Boolean Literals are the literals which have only true or false which can be represented as either 0 or 1.
4. Special Literals: It is represented by the value ‘none’ and used to classify fields which are not created.

Q25. What are DataFrame and panda series?

- DataFrame is simply represented by a table containing rows and columns, it is a two dimensional data structure.
- Here is the example of creating DataFrame from dictionary of n arrays or list:

```
Code: import as pd

        Data={‘Name’:[‘Lara’,’Sharon,’Damon’,Elena’],
        ‘Age’:[40,29,60,30]}

        #here we are creating DataFrame

        df=pd.DataFrame(data)

        #printing the output

        print(df)
```

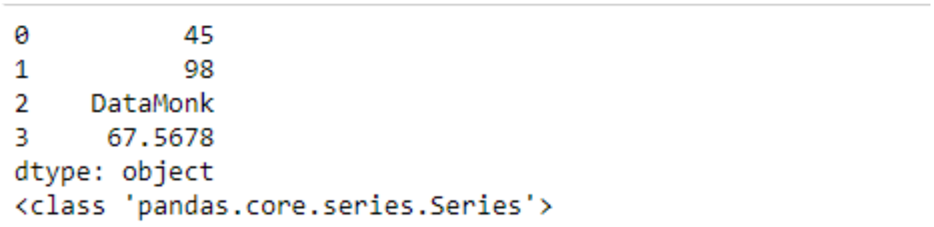


```
     Name  Age
0    Lara   40
1  Sharon   29
2   Damon   60
3   Elena   30
```

- Pandas is a library which is used for data manipulation and data analysis. Pandas series is a one dimensional array, the axis labels are called as index. It can hold any type of data like integer, float, string, object etc.
- Pandas Series can be created from the lists, dictionary, and from a scalar value etc.
- Here is the example of creating a series from data:

```
Code: import pandas as pd

        data=[“45”,98,”DataMonk”,67.5678]

        series=pd.Series(data)

         print(series)
```

```
# here type of series is printed

print(type(series))
```



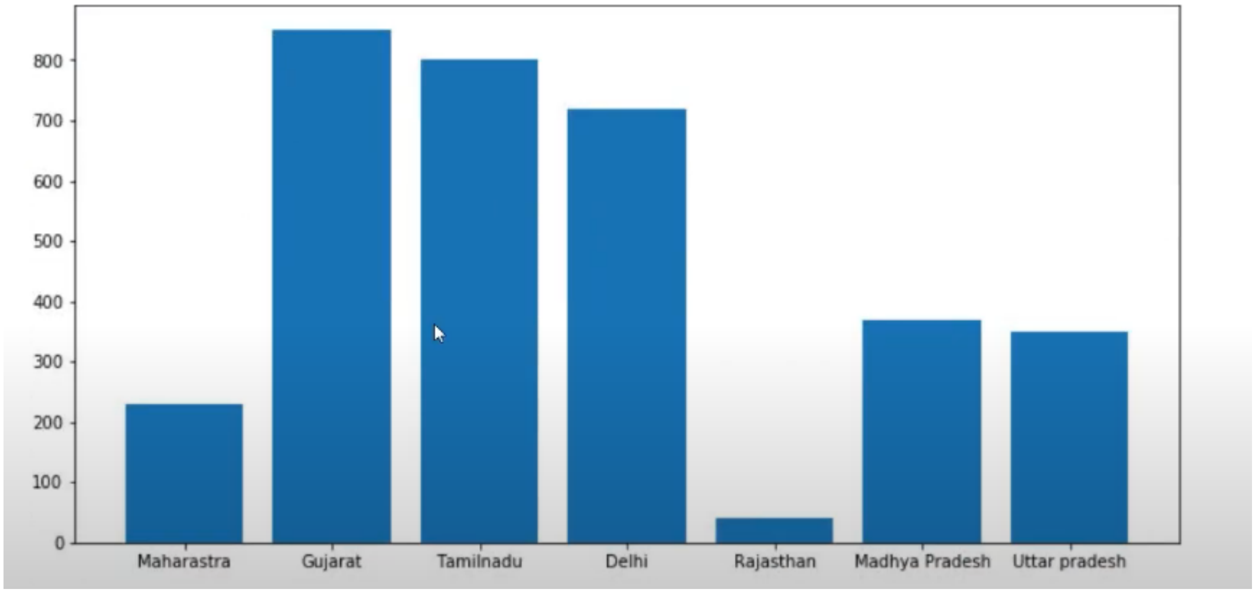
```
0         45
1         98
2   DataMonk
3    67.5678
dtype: object
<class 'pandas.core.series.Series'>
```

Q26. Create bar plot using Matplotlib, Seaborn, Plotly.

- Matplotlib: Matplotlib is a plotting library for the Python programming language for creating static, animated, and interactive visualizations in Python, it makes thing easy and used to make bar plots.
- Here is the first example of making bar plot using matplotlib:

```
Code: import matplotlib as plt

       %matplotlib inline

      location=[‘Tamil
Nadu’,’Chennai’,’Kolkatta’,’Mumbai’,’Delhi’,’Gujrat’]

      mobile_stock=[500,548,365,200,589,324]

      # barplot using matplotlib

      plt.figure(figsize=(12,6))

      plt.bar(x=location, height=mobile_stock)
```

- Here we have taken two variables location and mobile_stock where location has all the states and mobile_stock present in every respective states.
- Here is the output of barplot using Matplotlib:

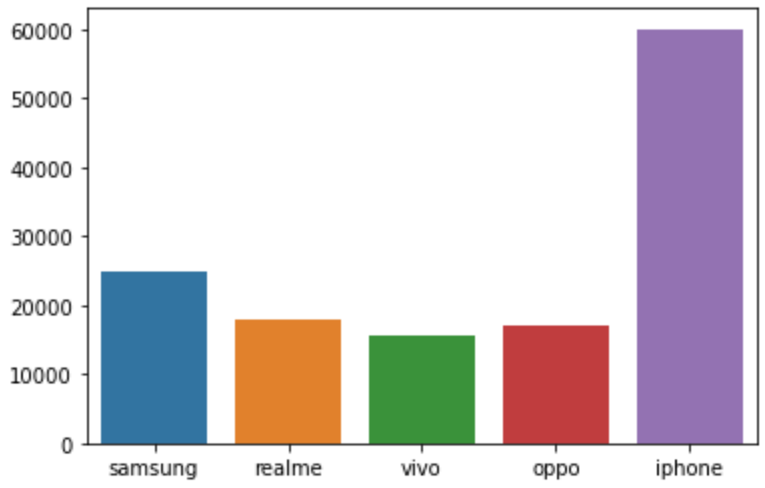


- Seaborn: Seaborn is a plotting visualization library based on matplotlib, it has fewer syntax and different default themes and seaborn requires matplotlib to be installed first.

  Code:
  ```
  import seaborn as sns

      brand=['samsung','realme','vivo','oppo','iphone']

      price=[25000,18000,15500,17000,60000]

      sns.barplot(x=brand, y=price)
  ```

- Here we have two variables brand and price, brand variable has various phone brands and price variable has price of respective brand.
- The seaborn plot looks better than matplotlib because of colour change in each bar column.

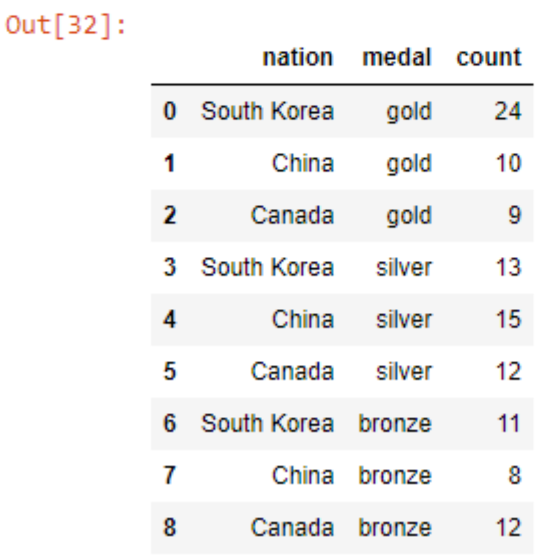
```
<matplotlib.axes._subplots.AxesSubplot at 0x2ff5fbd520>
```



This is the output of bar plot created by seaborn .

- Plotly Express is a built-in part of the plotly library, and easy to use
- px.bar or px.scatter is used to operate on column type data, on matrix like data px.choropleth and px.choropleth_mapbox can operate on geographic data
- Plotly Express works with Long-, Wide-, and Mixed-Form Data:
  long-form data has one row per observation, and one column per variable. This is suitable for storing and displaying dimensions having greater than 2.
- Here is the example of data represented in long-form:

Code:
```
import plotly.express as px
long_df=px.data.medals_long( )
long_df
```


Out[32]:

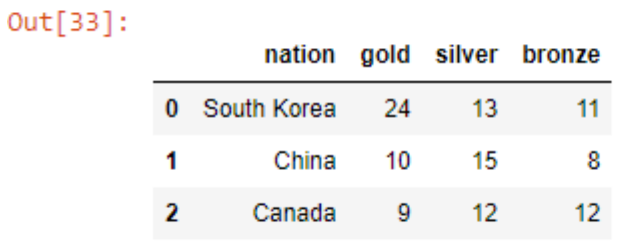
	nation	medal	count
0	South Korea	gold	24
1	China	gold	10
2	Canada	gold	9
3	South Korea	silver	13
4	China	silver	15
5	Canada	silver	12
6	South Korea	bronze	11
7	China	bronze	8
8	Canada	bronze	12

- wide-form data has one row per value of one of the first variable, and one column per value of the second variable.
- Here is the example of data represented in wide-form:

Code: import plotly.express as px

wide_df= px.data.medals_wide( )

wide_df


Out[33]:

	nation	gold	silver	bronze
0	South Korea	24	13	11
1	China	10	15	8
2	Canada	9	12	12

- This is the code for making bar plot using pyplot express:

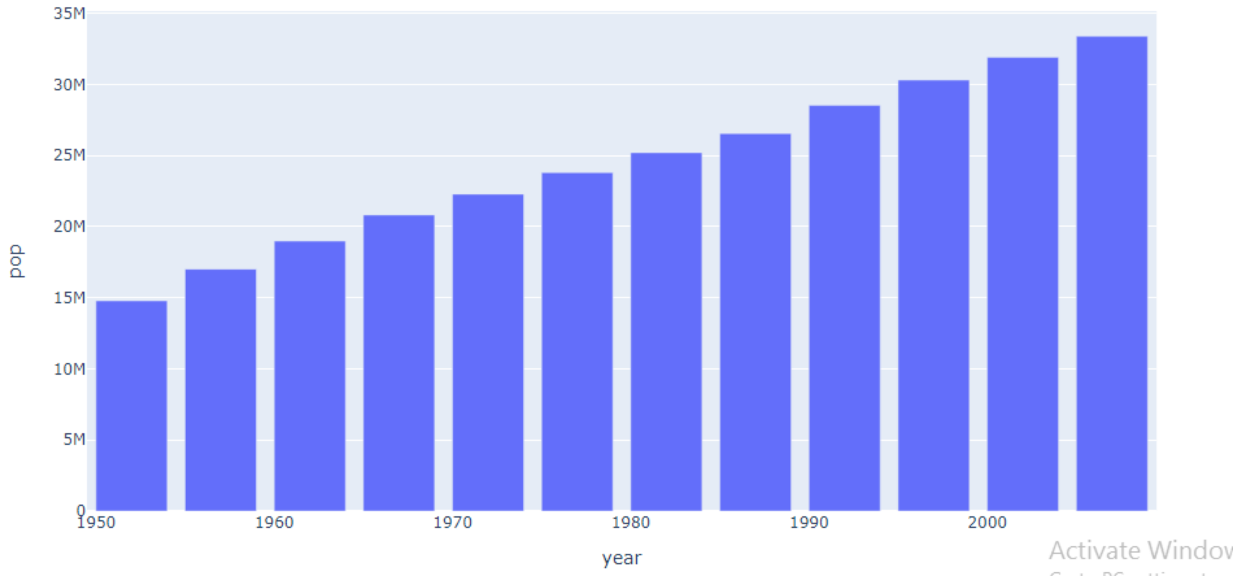
Code: import plotly.express as px

data=px.data.gapminder( ).query("country=='Canada'")

fig=px.bar(data, x='year', y='pop')

fig.show( )

- Output:



Q27. What will be the output of the below Python code –

def multipliers ():

return [lambda x: i * x for i in range (4)]

print [m (2) for m in multipliers ()]

- The output of the following code would be [6,6,6,6]

Q28. In which sectors we can use Python?

- Python is a programming language which is very easy to use as well as understand, it is a high-level programming language that has English-like syntax, there are various applications of Python programming.
- In today's life we can say that python is the most trending programming language.

The applications of Python programming is:

- Network Programming
- Games, 3D Graphics
- Other Python Applications
- Applications in Business
- Database Access
- Desktop GUI
- Scientific and Numeric Applications
- Software Development Applications
- Applications in Education
- Web and Internet Development

Q29. Python's math module has ceil( ) and floor( ) division function, what is the difference between them?

- Python has a round( ) function, which round off to the given number of digits and returns floating point number.
- For example you have 4 digit number and if the next number exceeds 4.50 then its rounded off to 5 and if the next number is not exceeded than 4.50 then the number is rounded off to 4.

The example would be:

```
> round(3.99)

4

> round(2.48)

2
```

- ceil( ) stands for the ceiling function which return number greater than the given number. For if the we if number 7.89 then by apply ceil( ) to it, it will give 8 number in the output even if the number is 7.02 it will return 8 because it returns one number greater than the given number. The example would be:

```
> import math
```

```
>math.ceil(3.33)

4

>math.ceil(3.65)

4
```

- floor( ) return number smaller than the given number. For example if we take number 7.89 then by apply floor( ) to it, it will give 7 number in the output even if the number is 7.02 it will return 7 because it returns one number smaller than the given number. The example would be:

```
>import math

>math.floor(3.33)

3

>math.floor(3.65)

3
```

Q30. Python's standard library has a build-in function next( ). For what purpose it's used?

- The built-in next() function calls __next__() method of iterator object to retrieve next item in it. Here the data is collected in iterator.
- One by one the element is displayed one after the other and at the end when string or object ends it will give Traceback error of StopIterator.

The example of next( ) function would be:

```
a=iter(["Python","is","a","language"])

a

Output: <list_iterator at 0x8f3f518190>

next(a)

Output: 'Python'
```

```
next(a)
```

Output: 'is'

```
next(a)
```

Output: 'a'

```
next(a)
```

Output: ' language'

```
next(a)
```

Output:

```
StopIteration Traceback (most recent call last)
<ipython-input-13-15841f3f11d4> in <module>
----> 1 next(a)

StopIteration
```

Q31. What is JSON? How can Python objects be represented in JSON format?

- JSON means Javascript object notation, Python library which has json modul e converts
 python object to JSON format. It has loads( ) and dump( ) to serialize and de -serialize Python objects.

```
import json
#dumps() converts Python object to JSON
data={'name':'Manali', 'subjects':['phy', 'che', 'maths'], 'marks':[70,90,60]}
jso=json.dumps(data)
#loads() converts JSON string back to Python object
Pydata=json.loads(jso)
Pydata
Output: {'name': 'Manali', 'subjects': ['phy', 'che', 'maths'], 'marks': [70, 90, 60]}
```

- We store JSON string to a file using load() and retrieve Python dictionary ob ject from it and that file should have write and read permission.

```
data={'name': 'Manali', 'subjects':['phy', 'che', 'maths'], 'marks':[ 70,90,60]}

file=open('jsonfile.txt','w')

json.dump(data, file)

file.close()

Output: {'name': 'Manali', 'subjects':['phy', 'che', 'maths'], 'marks':[ 70,90,60]}
```

- We use load( ) to load the json string from file to Python dictionary:

```
file=open('jsonfile.txt','r')

data=json.load(file)

data

Output: {'name': 'Manali', 'subjects':['phy', 'che', 'maths'], 'marks':[ 70,90,60]}

file.close()
```

Q32. What is the difference between set and frozenset?

- Set is an unordered collection of unique elements, set doesn’t allow duplicate elements that is it removes duplicate elements automatically.

You can initialize set in these ways:

```
cars={“BMW”,”Maruti”,”Honda”,”Toyota”, “BMW”}

type(cars)

Output: < class ‘set’ >

cars

Output: {“BMW”,”Maruti”,”Honda”,”Toyota”}
```

The second way to initialize a set is:

```
a=set( )

a.add(1)

a.add(2)

Output: {1,2}
```

- As set is an unordered collection, we can’t do indexing on set.
- Frozen Set: Unordered collection and it doesn’t allow you to change the cont ent of frozen set that is frozen sets are immutable (add/remove operation is n ot possible) this is the basic difference between a set and frozen set.

```
f=frozenset([10,20,30])

f

Output: frozenset({10, 20, 30})

f.add(40)

Traceback (most recent call last):

 File "<pyshell#24>", line 1, in <module>

   f.add(40)

AttributeError: 'frozenset' object has no attribute 'add'
```

Q31. Does Python support polymorphism? Explain with a suitable example.

- Polymorphism is an ability in object oriented programming to use common i nterface for multiple form (data types).
- Polymorphism means same function name (but different signatures) being us es for different types.
- Suppose, we need to color a shape, there are multiple shape option(rectangle, square, triangle, circle).
- However we could use same method to color any shape. This concept is calle d polymorphism.

The example of polymorphism would be:

Code:

```
class Parrot:
    def fly(self):
        print("Parrot can fly")
    def swim(self):
        print("Parrot can't swim")
class Penguin:
    def fly(self):
        print("Penguin can't fly")
    def swim(self):
        print("Penguin can swim")
def fly_test(bird):
    bird.fly()
pa=Parrot()
pe=Penguin()
fly_test(pa)
fly_test(pe)
```

Output:
```
Parrot can fly
Penguin can't fly
```

Q33. What is the difference between bytes( ) and bytearray( ) in python?

- bytes( ): bytes( ) is group of byte values. If you are using images, audio, vide o files then only we use bytes( )

Here are the steps to use bytes( ) :
L=[10,20,30,40]

```
b=bytes(L)
print(type(b))
```

```
Output: bytes
```

```
for x in L:
    print(x)
```

```
Output:  10
          20
          30
          40
```

Rules:

1. Order is important in bytes( ).
2. The bytes( ) datatype is valid between range of 0 to 255.
3. byte( ) is immutable datatype, values can't be changed.

bytearray( ): The difference between bytes( ) and bytearray( ) is that, bytes( ) is i mmutable and bytearray( ) is mutable.

Here are the steps to use bytes( ) :

```
L=[10,20,30,40]
b=bytearray(L)
print(type(b))
```

```
Output: bytearray
```

```
for x in L:
    print(x)
```

```
Output:  10
          20
          30
          40
```

Rules:

1. The bytes( ) datatype is valid between range of 0 to 255.
2. bytearray( ) is mutable, value of list can be accessed and can be changed or r eplaced.

Q34. What is metaclass in Python?

- Class is like a constructor which we use in python to create object.
- We have classes which we are using to create objcets
- Object which is also known as instance of a class is basically used to access t he attributes in a class.
- To access those attributes we have to create the object or create instance of cl ass and also when we create these instances the method of creating an instan ce for class is known as instantiation.
- Everything in python is treated as object and even a class for that matter is tr eated as as an object in Python so it's basically all about instances.
- Now since a class is also a object which means that class was instantiated fro m a different class so when you are creating class.
- There is another class which is being made or which is ready there from whic h we are creating another object so this class is called the Meta class.
- Meta class has defined the behavior of any class that we are declaring in pyth on because it's an object. Type is basically the meta class that we have in pyt hon and the instances of type are the class that we use in python so basically the object of the meta class that we have in python is the type class so whate ver object that we have in python is basically an instance of the type class an d type can also be used to create classes dynamically.

Q35. What is shebang in Python?

- Shebang stands for special character symbol which stands for #!
- The name shebang for the distinctive two characters comes from an inexact c ontraction of SHArp bang or haSH bang, referring to the two typical Unix na mes for them.
- It is also called sha-bang, hashbang, pound-bang, or hash-pling.It has a speci al meaning for UNIX operating system.
- #!/usr/bin/python3 is a shebang line

- A shebang line defines where the interpreter is located. In this case, the pytho n3 interpreter is located in /usr/bin/python3 .
- Shebang line is not actually needed by script, it is for the system to know ho w to find interpreter.
- Need of Shebang: The shebang is used if you run the script directly as an exe cutable file (for example with the command ./script.sh).

Q36. What is assert in Python? Is it a function, class or keyword? Where an d how is it used?

- Python has built in keyword called assert , assert will help you to perform many types of assertion on your requirement.
- assert will simply taken one condition and if this condition evaluates to true it will continue with next set of statement buy if it fails it will throw asser tion error.
- Or if the condition becomes false then you can also give an optional messa ge.

Code: assert True
```
print("DataMonk")
```
Output: Hey
```
assert False
print("DataMonk")
```
```
Output: AssertionError  Traceback (most recent call last)
<ipython-input-2-72fad4a4f591> in <module>
----> 1 assert False
      2 print("DataMonk")

AssertionError:
```

```
assert True,"Condition failed"
print("DataMonk")
```
Output: DataMonk

```
assert False,"Condition failed"
print("DataMonk")
```

```
Output: AssertionError Traceback (most recent call last)
<ipython-input-4-da6c8900508c> in <module>
----> 1 assert False,"Condition failed"
      2 print("DataMonk")

AssertionError: Condition failed
```
This gives assertion error with condition failed message.

Code:
```
assert "Selenium" in "Selenium is for web application","Validation passed"
print("Validation1 passed")
str1="Python"
str2="Python"
assert str1==str2, "String matching failed"
print("Validation 2 passed")
```
```
Output: Validation1 passed
        Validation 2 passed
```

Code:
```
assert "Selenium" in ["Zelenium","Python","Java"],"Selenium is not present"
print("Validation 3 passed")
```
```
Output: AssertionError Traceback (most recent call last)
<ipython-input-7-9732ac2a05b0> in <module>
----> 1 assert "Selenium" in ["Zelenium","Python","Java"],"Selenium is not present
"
      2 print("Validation 3 passed")

AssertionError: Selenium is not present
```

Q37. Python Program to check if given array is Monotonic or not.
An array is monotonic **if and only if it is** monotone increasing**, or** monotone dec reasing**.**
we only need to check adjacent elements to determine if the array is monotone inc reasing or decreasing.
To check whether an array A is monotone increasing**:**

we’ll check A[i] <= A[i+1] for all i indexing from 0 to len(A)-2.
we can check for monotone decreasing where A[i] >= A[i+1] for all i indexing fro
m 0 to len(A)-2.

Code:
```
# Python3 program to find sum in nth group

# Check if given array is Monotonic
def isMonotonic(A):

    return (all(A[i] <= A[i + 1] for i in range(len(A) - 1)) or
            all(A[i] >= A[i + 1] for i in range(len(A) - 1)))

A = [6, 5, 4, 4]

# Print result
print(isMonotonic(A))
```

Output: True

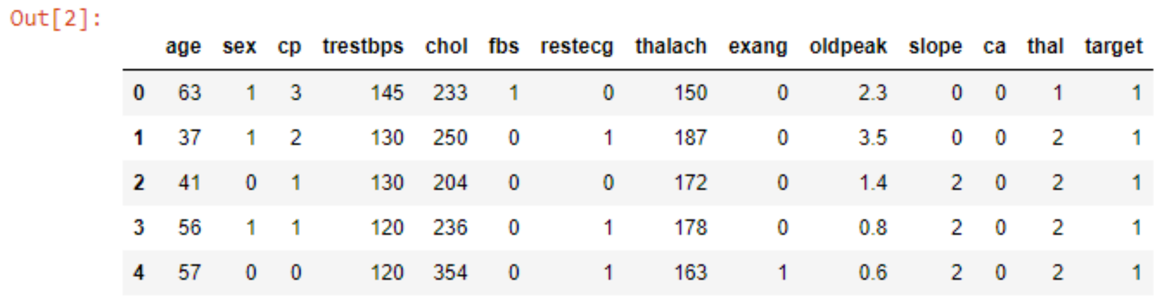
Q38. What is groupby in pandas explain with an example.

- groupby() is a function in python which is used to group the data based on the given criteria given by the programmer.
- Suppose you have different vehicles of different type, we can group them according to the type of vehicle.
- I am having a csv file which has all the functions from which we can predict whether a person is having a heart disease or not.

Code:
```
import pandas as pd
import numpy as np
df = pd.read_csv(r'C:\Users\Manali\Desktop\data monk\heart.csv')
df.head()
```

# The **head()** function is used to get the first n rows. This function returns the first n rows #for the object based on position.

Output:



Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

These are the top 5 rows of the heart.csv file

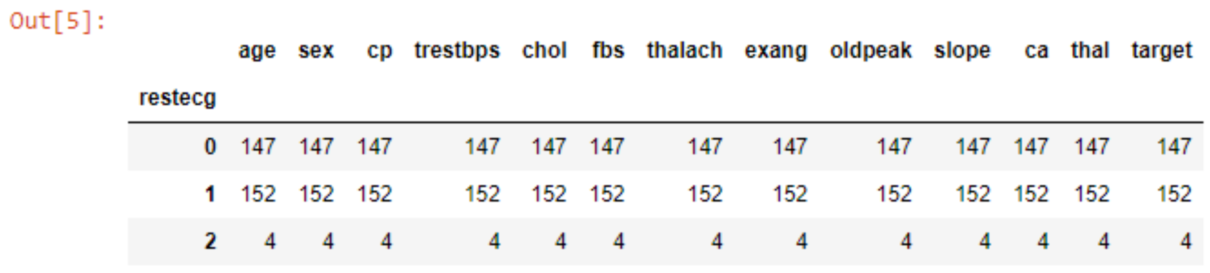
df.groupby('age')

Output:

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000000599 4B3AC0>

df.groupby('restecg').count()

# The **count()** function is used to **count** non-NA cells for each column or ro w. The values None, #NaN, NaT, and optionally numpy



Out[5]:

	age	sex	cp	trestbps	chol	fbs	thalach	exang	oldpeak	slope	ca	thal	target
restecg													
0	147	147	147	147	147	147	147	147	147	147	147	147	147
1	152	152	152	152	152	152	152	152	152	152	152	152	152
2	4	4	4	4	4	4	4	4	4	4	4	4	4

Summary