**1. What are the key features of Python?**

Python is one of the most popular programming languages used by data scientists and AIML professionals. This popularity is due to the following key features of Python:

* Python is easy to learn due to its clear syntax and readability
* Python is easy to interpret, making debugging easy
* Python is free and Open-source
* It can be used across different languages
* It is an object-oriented language which supports concepts of classes
* It can be easily integrated with other languages like C++, Java and more

**2. What are Keywords in Python?**

* Keywords in Python are reserved words which are used as identifiers, function name or variable name. They help define the structure and syntax of the language.
* There are a total of 33 keywords in Python 3.7 which can change in the next version, i.e., Python 3.8. A list of all the keywords is provided below:
* **Keywords in Python**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | class | finally | is | return |
| None | continue | for | lambda | try |
| True | def | from | nonlocal | while |
| and | del | global | not | with |
| as | elif | if | or | yield |
| assert | else | import | pass |  |
| break | except |  |  |  |

**3. What are Literals in Python and explain about different Literals?**

Literals in Python refer to the data that is given in a variable or constant. Python has various kinds of literals including:

1. String Literals: It is a sequence of characters enclosed in codes. There can be single, double and triple strings based on the number of quotes used. Character literals are single characters surrounded by single or double-quotes.
2. Numeric Literals: These are unchangeable kind and belong to three different types – integer, float and complex.
3. Boolean Literals: They can have either of the two values- True or False which represents ‘1’ and ‘0’ respectively.
4. Special Literals: Special literals are sued to classify fields that are not created. It is represented by the value ‘none’.

**4. How can you concatenate two tuples?**

**Solution ->**

Let’s say we have two tuples like this ->

tup1 = (1,”a”,True)

tup2 = (4,5,6)

Concatenation of tuples means that we are adding the elements of one tuple at the end of another tuple.

Now, let’s go ahead and concatenate tuple2 with tuple1:

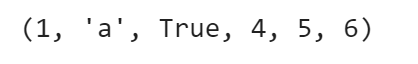
Code

tup1=(1,"a",True)

tup2=(4,5,6)

tup1+tup2

Output



All you have to do is, use the ‘+’ operator between the two tuples and you’ll get the concatenated result.

Similarly, let’s concatenate tuple1 with tuple2:

Code

tup1=(1,"a",True)

tup2=(4,5,6)

tup2+tup1

Output

https://lh5.googleusercontent.com/HF6v52iiVXCMwteW8WoUv1s-ru3rIa1jDAkjqXPKmmTGSC5HQ2WG4gnJkXoW2ok98Y7ExoV9ttsx_BaARohwZk3AGQhP3DNfMfX1iHI13DQjmRApacKqj63QIThlqjHkBisT7LU9

**5. What are functions in Python?**

Ans: Functions in Python refer to blocks that have organised, and reusable codes to perform single, and related events. Functions are important to create better modularity for applications which reuse high degree of coding. Python has a number of built-in functions like print(). However, it also allows you to create user-defined functions.

**6. How to Install Python?**

To Install Python, first go to Anaconda.org and click on “Download Anaconda”. Here, you can download the latest version of Python. After Python is installed, it is a pretty straightforward process. The next step is to power up an IDE and start coding in Python. If you wish to learn more about the process, check out this [Python Tutorial.](https://www.mygreatlearning.com/blog/python-tutorial-for-beginners-a-complete-guide/)

**7. What is Python Used For?**

Python is one of the most popular programming languages in the world today. Whether you’re browsing through Google, scrolling through Instagram, watching videos on YouTube, or listening to music on Spotify, all of these applications make use of Python for their key programming requirements. Python is used across various platforms, applications, and services such as web development.

**8. How can you initialize a 5\*5 numpy array with only zeroes?**

***Solution ->***

We will be using the .zeros() method

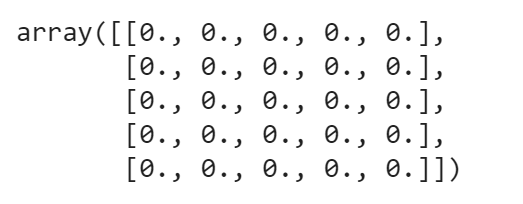
import numpy as np

n1=np.zeros((5,5))

n1

Use np.zeros() and pass in the dimensions inside it. Since, we want a 5\*5 matrix, we will pass (5,5) inside the .zeros() method.

This will be the output:



**9. What is Pandas?**

Pandas is an open source python library which has a very rich set of data structures for data based operations. Pandas with it’s cool features fits in every role of data operation, whether it be academics or solving complex business problems. Pandas can deal with a large variety of files and is one of the most important tools to have a grip on.

**10. What are dataframes?**

A pandas dataframe is a data structure in pandas which is mutable. Pandas has support for heterogeneous data which is arranged across two axes.( rows and columns).

Reading files into pandas:-

|  |  |
| --- | --- |
| 1  2 | Import pandas as pd  df=p.read\_csv(“mydata.csv”) |

Here df is a pandas data frame. read\_csv() is used to read a comma delimited file as a dataframe in pandas.

**11. What is a Pandas Series?**

Series is a one dimensional pandas data structure which can data of almost any type. It resembles an excel column. It supports multiple operations and is used for single dimensional data operations.

Creating a series from data:

**Code**

import pandas as pd

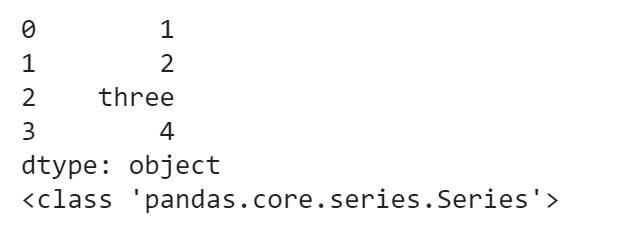
data=["1",2,"three",4.0]

series=pd.Series(data)

print(series)

print(type(series))

**Output**



**12. What is pandas groupby?**

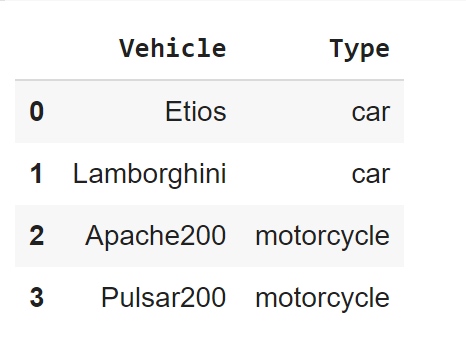
A pandas groupby is a feature supported by pandas which is used to split and group an object.  Like the sql/mysql/oracle groupby it used to group data by classes, entities which can be further used for aggregation. A dataframe can be grouped by one or more columns.

**Code**

df = pd.DataFrame({'Vehicle':['Etios','Lamborghini','Apache200','Pulsar200'], 'Type':["car","car","motorcycle","motorcycle"]})

df

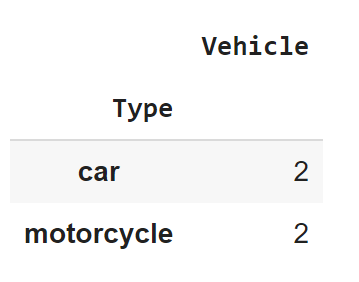
**Output**



To perform groupby type the following **code:**

df.groupby('Type').count()

**Output**



**13. How to create a dataframe from lists?**

To create a dataframe from lists ,

1)create an empty dataframe

2)add lists as individuals columns to the list

**Code**

df=pd.DataFrame()

bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]

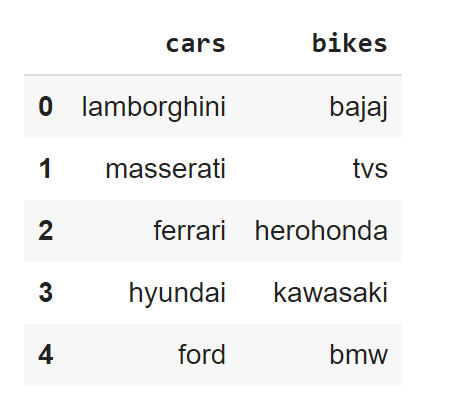
cars=["lamborghini","masserati","ferrari","hyundai","ford"]

df["cars"]=cars

df["bikes"]=bikes

df

**Output**



**14. How to create dataframe from a dictionary?**

A dictionary can be directly passed as an argument to the DataFrame() function to create the data frame.

Code

import pandas as pd

bikes=["bajaj","tvs","herohonda","kawasaki","bmw"]

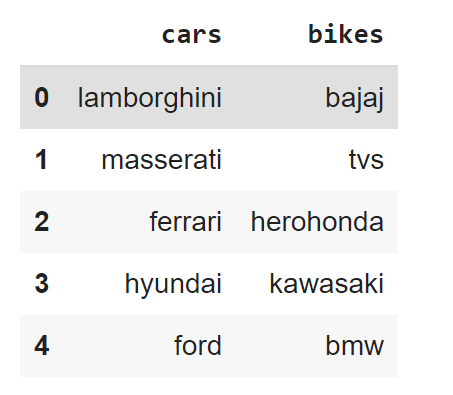
cars=["lamborghini","masserati","ferrari","hyundai","ford"]

d={"cars":cars,"bikes":bikes}

df=pd.DataFrame(d)

df

Output



**15. How to combine dataframes in pandas?**

Two different data frames can be stacked either horizontally or vertically by the concat(), append() and join() functions in pandas.

Concat works best when the dataframes have the same columns and can be used for concatenation of data having similar fields and is basically vertical stacking of dataframes into a single dataframe.

Append() is used for horizontal stacking of dataframes. If two tables(dataframes) are to be merged together then this is the best concatenation function.

Join is used when we need to extract data from different dataframes which are having one or more common columns. The stacking is horizontal in this case.

Before going through the questions, here’s a quick video to help you refresh your memory on Python.

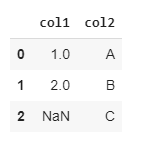
**16. What kind of joins does pandas offer?**

Pandas has a left join, inner join, right join and an outer join.

**17. How to merge dataframes in pandas?**

Merging depends on the type and fields of different dataframes being merged. If data is having similar fields data is merged along axis 0 else they are merged along axis 1.

**18. Give the below dataframe drop all rows having Nan.**

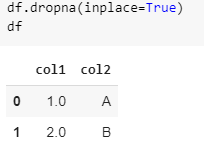


The dropna function can be used to do that.

df.dropna(inplace=True)

df

Output



**19. How to access the first five entries of a dataframe?**

By using the head(5) function we can get the top five entries of a dataframe. By default df.head() returns the top 5 rows. To get the top n rows df.head(n) will be used.

**20. How to access the last five entries of a dataframe?**

By using tail(5) function we can get the top five entries of a dataframe. By default df.tail() returns the top 5 rows. To get the last n rows df.tail(n) will be used.