1) What do you mean by Class Variables and Instance Variables in Python OOP?

Reference: https://www.edureka.co/blog/python-class/

- **Class variable** is a variable that is shared by all the different objects/instances of a class.
- **Instance variables** are variables which are unique to each instance. It is defined inside a method and belongs only to the current instance of a class.

2) What is self in Python OOP? Write the syntax of self.

Reference: https://intellipaat.com/blog/interview-question/python-interview-questions/

Self is an object or an instance of a class. This is explicitly included as the first parameter in Python. On the other hand, in Java it is optional. It helps differentiate between the methods and attributes of a class with local variables.

The self variable in the init method refers to the newly created object, while in other methods, it refers to the object whose method was called.

Syntax:

```
Class A:
def func(self):
print(“Hi”)
```

3) Discuss re.findall() in the Python Regular Expressions with a suitable example.

Reference: https://www.w3schools.com/python/python_regex.asp

```
# Example
import re
#Return a list containing every occurrence of "ai":
txt = "The rain in Spain"
x = re.findall("ai", txt)
print(x)
```

```
['ai', 'ai']
```

4) How to create a new database in SQLite by using Python?

Reference: https://realpython.com/data-engineer-interview-questions-python/#a-sqlite-example

A SQLite Example

Now that you've answered what relational databases are, it's time to dig into some Python! SQLite is a convenient database that you can use on your local machine. The database is a single file, which makes it ideal for prototyping purposes. First, import the required Python library and create a new database:

Python

```
import sqlite3

db = sqlite3.connect(':memory:')  # Using an in-memory database
cur = db.cursor()
```

5) **How do you create a 3D array in NumPy? How can you identify the datatype of a given NumPy array? Discuss with the example.**

Reference: https://www.upgrad.com/blog/numpy-interview-questions-and-answers/

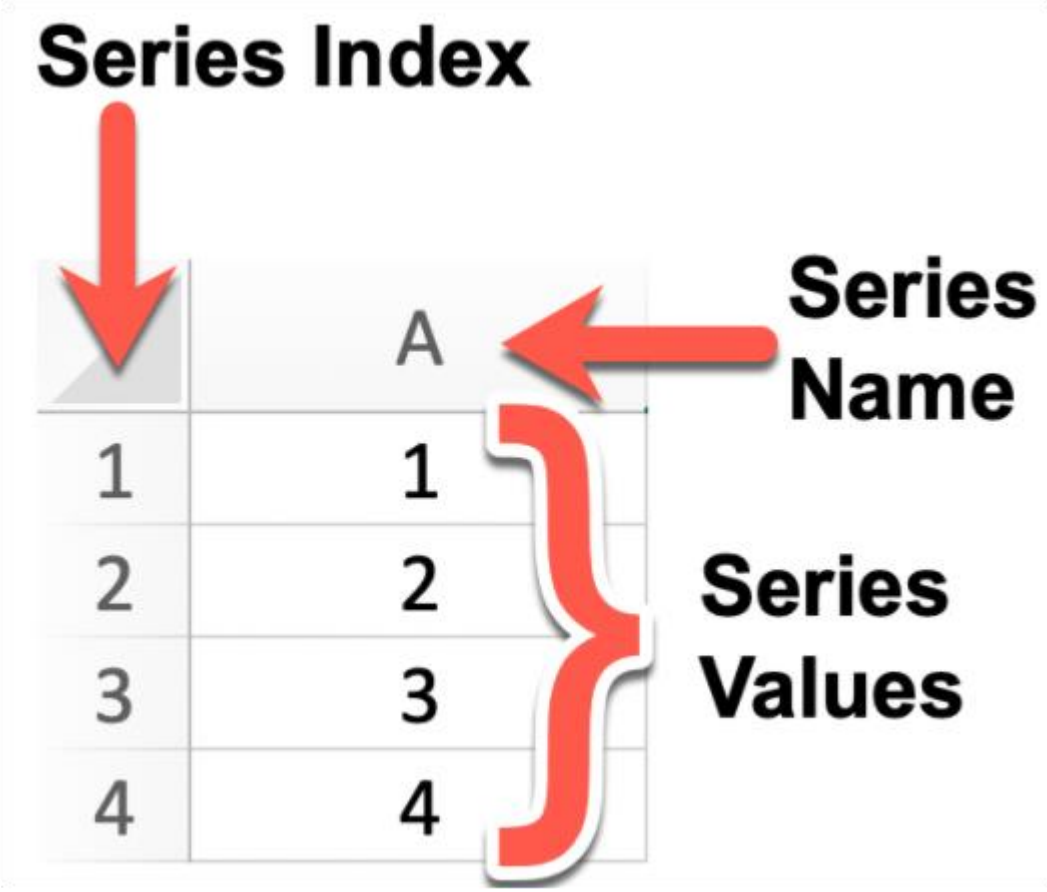
```
#Example
import numpy as np
num3=[[[1,2,3],[4,5,6],[7,8,9]]]
num3 = np.array(num3)
print(num3)
print(num3.dtype)
```

```
[[[1 2 3]
  [4 5 6]
  [7 8 9]]]
int64
```

6) What are the different types of Data Structures in Pandas?

Reference: https://www.upgrad.com/blog/pandas-interview-questions-answers-for-freshers-experienced/

Panda library supports two major types of data structures, DataFrames and Series. Both these data structures are built on top of NumPy. Series is a one-dimensional and simplest data structure, while DataFrame is two dimensional. Another axis label known as the “Panel” is a 3-dimensional data structure and includes items such as major_axis and minor_axis.



7) Define: Graph / Chart
Why do you use Matplotlib?

- **Definition: Graph / Chart**

A graph or chart is simply a visual representation of numeric data.

Matplotlib: Visualization with Python

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

8) What are the features of the Matplotlib?

- Matplotlib makes easy things easy and hard things possible.

Create	Customize	Extend
• Develop publication quality plots with just a few lines of code • Use interactive figures that can zoom, pan, update...	• Take full control of line styles, font properties, axes properties... • Export and embed to a number of file formats and interactive environments	• Explore tailored functionality provided by third party packages • Learn more about Matplotlib through the many external learning resources

9) What is Web Scraping?

Reference: https://www.edureka.co/blog/web-scraping-with-python/#whywebscraping

What is Web Scraping?

Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code. In this article, we'll see how to implement web scraping with python.

10) What is the framework or options are available to develop a GUI in Python?

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

- Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- wxPython − This is an open-source Python interface for wxWindows http://wxpython.org.
- JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

11) What is Python Class Inheritance? What are the types of Inheritance?

Reference: https://www.edureka.co/blog/python-class/

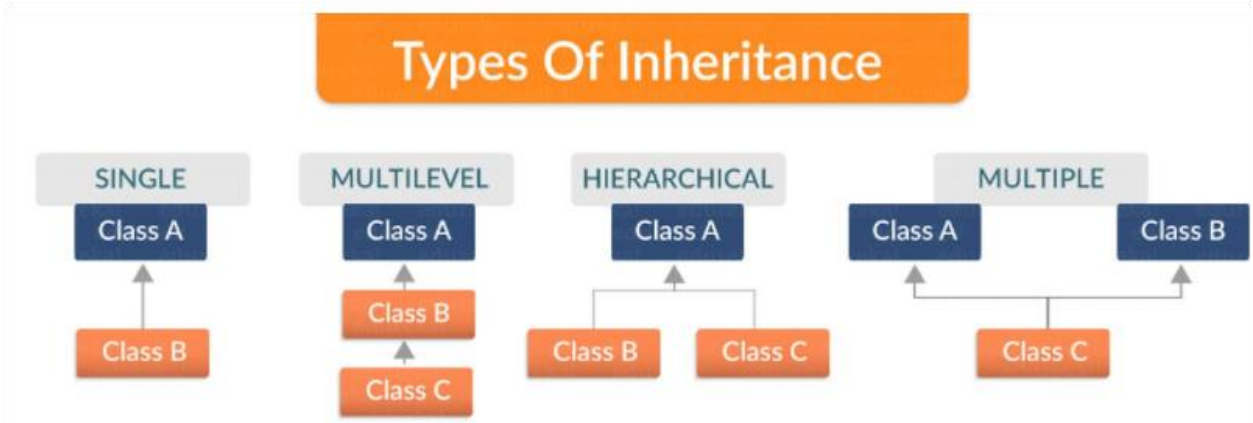
Inheritance allows us to inherit attributes and methods from the base/parent class. This is useful as we can create sub-classes and get all of the functionality from our parent class. Then we can overwrite and add new functionalities without affecting the parent class. Let's understand the concept of parent class and child class with an example.

As we can see in the image, a child inherits the properties from the father. Similarly, in python, there are two classes:

1. Parent class (Super or Base class)
2. Child class (Subclass or Derived class)

A class that inherits the properties is known as Child Class whereas a class whose properties are inherited is known as Parent class.

Inheritance refers to the ability to create Sub-classes that contain specializations of their parents. It is further divided into four types namely single, multilevel, hierarchical, and multiple inheritances. Refer to the below image to get a better understanding.


Types Of Inheritance
SINGLE
Class A
Class B
MULTILEVEL
Class A
Class B
Class C
HIERARCHICAL
Class A
Class B
Class C
MULTIPLE
Class A
Class B
Class C

12) What is __init__ in Python? Discuss it with the example.

Reference: https://intellipaat.com/blog/interview-question/python-interview-questions/

Equivalent to constructors in OOP terminology, __init__ is a reserved method in Python classes. The __init__ method is called automatically whenever a new object is initiated. This method allocates memory to the new object as soon as it is created. This method can also be used to initialize variables.

```
#Example
class Human:
# init method or constructor
  def __init__(self, age):
    self.age = age
# Sample Method
  def say(self):
    print('Hello, my age is', self.age)
h= Human(22)
h.say()
```

```
Hello, my age is 22
```

13)What are Python Attributes in the Python class? What are the types of Attributes? Discuss Public, Protected, and Private Attributes.

Reference: https://www.edureka.co/blog/python-class/

Attributes in Python defines a property of an object, element, or file. There are two types of attributes:

- **Built-in Class Attributes:** There are various built-in attributes present inside Python classes. For example _dict_, _doc_, _name _, etc. Let me take the same example where I want to view all the key-value pairs of employee1. For that, you can simply write the below statement which contains the class namespace:

  ```
  print(emp_1.__dict__)
  ```

  After executing it, you will get output such as: {'fname': 'aayushi', 'lname': 'johari', 'sal': 350000, 'email': 'aayushi.johari@company.com'}

- **Attributes defined by Users**: Attributes are created inside the class definition. We can dynamically create new attributes for existing instances of a class. Attributes can be bound to class names as well.

Next, we have public, protected, and private attributes. Let's understand them in detail:

Naming	Type	Meaning
Name	Public	These attributes can be freely used inside or outside of a class definition
_name	Protected	Protected attributes should not be used outside of the class definition, unless inside of a subclass definition
__name	Private	This kind of attribute is inaccessible and invisible. It's neither possible to read nor to write those attributes, except inside of the class definition itself

14) Discuss and list six Metacharacters with the example of the Python Regular Expressions.

Reference: https://www.w3schools.com/python/python_regex.asp

Metacharacters

Metacharacters are characters with a special meaning:

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"world$"
*	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
{}	Exactly the specified number of occurrences	"al{2}"
|	Either or	"falls|stays"
()	Capture and group	

15)List six Special Sequences with the example of the Python Regular Expressions.

Reference: https://www.w3schools.com/python/python_regex.asp

Special Sequences

A special sequence is a \ followed by one of the characters in the list below, and has a special meaning:

Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"
\b	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\bain" r"ain\b"
\B	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"\Bain" r"ain\B"
\d	Returns a match where the string contains digits (numbers from 0-9)	"\d"
\D	Returns a match where the string DOES NOT contain digits	"\D"
\s	Returns a match where the string contains a white space character	"\s"
\S	Returns a match where the string DOES NOT contain a white space character	"\S"
\w	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore _ character)	"\w"
\W	Returns a match where the string DOES NOT contain any word characters	"\W"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

16)What are sets in the Python Regular Expressions? Discuss at least six sets.

Reference: https://www.w3schools.com/python/python_regex.asp

Sets

A set is a set of characters inside a pair of square brackets `[]` with a special meaning:

Set	Description
[arn]	Returns a match where one of the specified characters (`a` , `r` , or `n`) are present
[a-n]	Returns a match for any lower case character, alphabetically between `a` and `n`
[^arn]	Returns a match for any character EXCEPT `a` , `r` , and `n`
[0123]	Returns a match where any of the specified digits (`0` , `1` , `2` , or `3`) are present
[0-9]	Returns a match for any digit between `0` and `9`
[0-5][0-9]	Returns a match for any two-digit numbers from `00` and `59`
[a-zA-Z]	Returns a match for any character alphabetically between `a` and `z` , lower case OR upper case
[+]	In sets, `+` , `*` , `.` , `|` , `()` , `$` , `{}` has no special meaning, so `[+]` means: return a match for any `+` character in the string

17)Distinguish between pandas.DataFrame.loc and pandas.DataFrame.iloc

Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html
Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html
Reference:https://towardsdatascience.com/how-to-use-loc-and-iloc-for-selecting-data-in-pandas-bd09cb4c3d79
Reference: https://www.analyticsvidhya.com/blog/2020/02/loc-iloc-pandas/

18) Write a program to perform element wise addition and subtraction between two 5x4 matrices by using NumPy.

```
# Elementwise Addition
import numpy as np
A = np.matrix(np.arange(0,20)).reshape(5,4)
B=np.matrix(np.arange(20,40)).reshape(5,4)
print(A)
print(' ')
print(B)
print(' ')
print(np.add(A,B))
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]]

[[20 21 22 23]
 [24 25 26 27]
 [28 29 30 31]
 [32 33 34 35]
 [36 37 38 39]]

[[20 22 24 26]
 [28 30 32 34]
 [36 38 40 42]
 [44 46 48 50]
 [52 54 56 58]]
```

```python
# Elementwise Substraction
import numpy as np
A = np.matrix(np.arange(0,20)).reshape(5,4)
B=np.matrix(np.arange(20,40)).reshape(5,4)
print(A)
print(' ')
print(B)
print(' ')
np.subtract(A,B)
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]]

[[20 21 22 23]
 [24 25 26 27]
 [28 29 30 31]
 [32 33 34 35]
 [36 37 38 39]]

matrix([[-20, -20, -20, -20],
        [-20, -20, -20, -20],
        [-20, -20, -20, -20],
        [-20, -20, -20, -20],
        [-20, -20, -20, -20]])
```

19) Discuss pandas.DataFrame.dropna and pandas.DataFrame.fillna with its parameters and with the suitable example.

Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.dropna.html
Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html

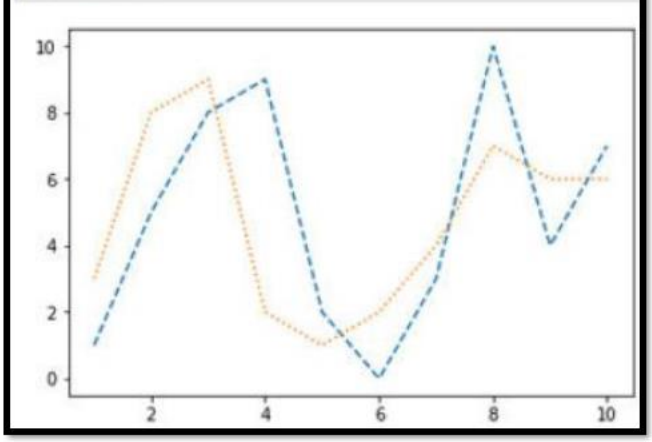
20) List out at least 5 Line Appearance and Line Colors characters.
Write a program to demonstrate Line Appearance and Line Colors.

- ## Matplotlib Line Style

Character	Line Style
'-'	Solid line
'--'	Dashed line
'-.'	Dash-dot line
':'	Dotted line

```
import matplotlib.pyplot as plt
%matplotlib inline

values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
plt.plot(range(1,11), values, '--')
plt.plot(range(1,11), values2, ':')
plt.show()
```



- ## Matplotlib Colours

Character	Color
'b'	Blue
'g'	Green
'r'	Red
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

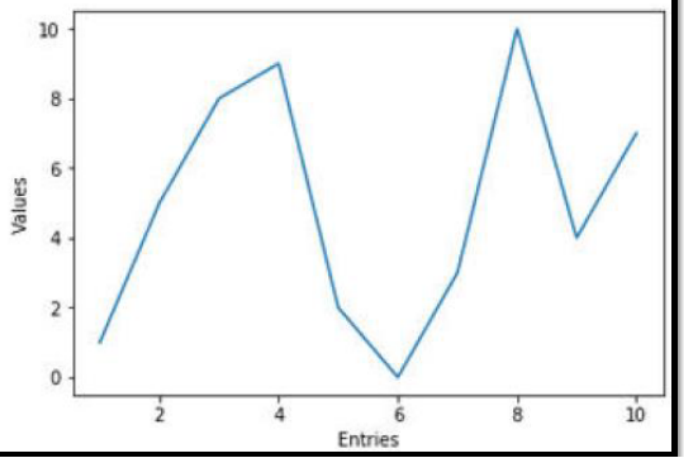
```
import matplotlib.pyplot as plt
%matplotlib inline

values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
plt.plot(range(1,11), values, 'r')
plt.plot(range(1,11), values2, 'm')
plt.show()
```

21)What is Label, Annotation, and Legends in the Matplotlib? Write a program to demonstrate Label, Annotation, and Legends in the Matplotlib.

```
import matplotlib.pyplot as plt
%matplotlib inline

values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
plt.xlabel('Entries')
plt.ylabel('Values')
plt.plot(range(1,11), values)
plt.show()
```

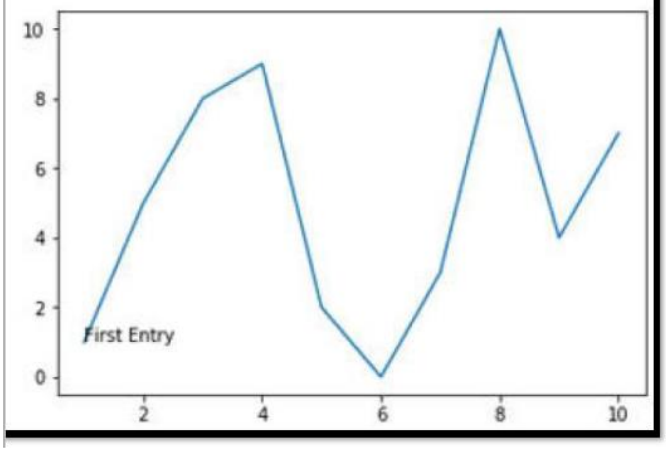


- Label

Label: Provides positive identification of a particular data element or grouping. The purpose is to make it easy for the viewer to know the name or kind of data illustrated.

```
import matplotlib.pyplot as plt
%matplotlib inline

values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
plt.annotate(xy=[1,1], s='First Entry')
plt.plot(range(1,11), values)
plt.show()
```

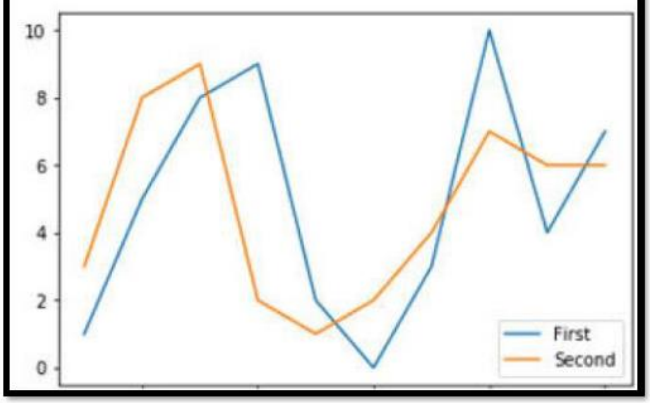


- Annotation

Annotation: Augments the information the viewer can immediately see about the data with notes, sources, or other useful information. In contrast to a label, the purpose of annotation is to help extend the viewer's knowledge of the data rather than simply identify it.

```
import matplotlib.pyplot as plt
%matplotlib inline

values = [1, 5, 8, 9, 2, 0, 3, 10, 4, 7]
values2 = [3, 8, 9, 2, 1, 2, 4, 7, 6, 6]
line1 = plt.plot(range(1,11), values)
line2 = plt.plot(range(1,11), values2)
plt.legend(['First', 'Second'], loc=4)
plt.show()
```



- Legends

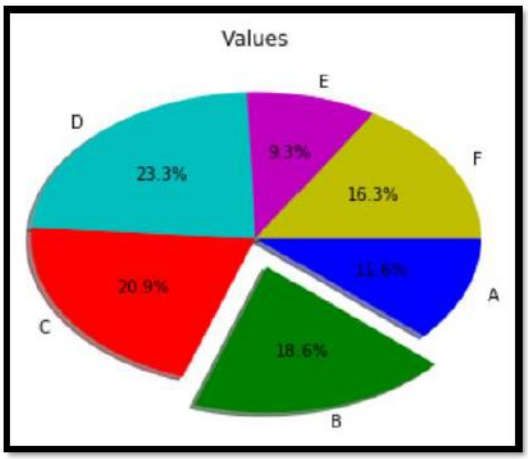
Legend: Presents a listing of the data groups within the graph and often provides cues (such as line type or color) to make identification of the data group easier. For example, all the red points may belong to group A, while all the blue points may belong to group B.

22)What is a Pie Chart? What are the Pie Wedges? Write a program to demonstrate Explode Parameters, Explode Values, autopct, counterclock, shadow of the pie chart.

- Pie charts focus on showing parts of a whole.
- The entire pie would be 100 percent.
- The question is how much of that percentage each value occupies.

- The colours parameter lets you choose custom colours for each pie wedge.
- You use the labels parameter to identify each wedge.
- In many cases you need to make one wedge stand out from the others, so you add the explode parameter with list of explode values.
- A value of 0 keeps the wedge in place — any other value moves the wedge out from the centre of the pie.
- Each pie wedge can show various kinds of information.

```
import matplotlib.pyplot as plt
%matplotlib inline
values = [5, 8, 9, 10, 4, 7]
colors = ['b', 'g', 'r', 'c', 'm', 'y']
labels = ['A', 'B', 'C', 'D', 'E', 'F']
explode = (0, 0.2, 0, 0, 0, 0)
plt.pie(values, colors=colors, labels=labels,
explode=explode, autopct='%1.1f%%',
counterclock=False, shadow=True)
plt.title('Values')
plt.show()
```



- This example shows the percentage occupied by each wedge with the autopct parameter.
- You must provide a format string to format the percentages.
- Use the counterclock parameter to determine the direction of the wedges.
- The shadow parameter determines whether the pie appears with a shadow beneath it (for a 3-D effect).
- You can find other parameters at https://matplotlib.org/api/pyplot_api.html .

23) How do you Scrape Data from A Website?

Reference: https://www.edureka.co/blog/web-scraping-with-python/#whywebscraping

When you run the code for web scraping, a request is sent to the URL that you have mentioned. As a response to the request, the server sends the data and allows you to read the HTML or XML page. The code then, parses the HTML or XML page, finds the data and extracts it.

To extract data using web scraping with python, you need to follow these basic steps:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

24) Why is Python good for Web Scraping?

Reference: https://www.edureka.co/blog/web-scraping-with-python/#whywebscraping

Here is the list of features of Python which makes it more suitable for web scraping.

- Ease of Use: Python is simple to code. You do not have to add semi-colons ";" or curly-braces "{}" anywhere. This makes it less messy and easy to use.
- Large Collection of Libraries: Python has a huge collection of libraries such as Numpy, Matlplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.
- Dynamically typed: In Python, you don't have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.
- Easily Understandable Syntax: Python syntax is easily understandable mainly because reading Python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in Python also helps the user to differentiate between different scope/blocks in the code.
- Small code, large task: Web scraping is used to save time. But what's the use if you spend more time writing the code? Well, you don't have to. In Python, you can write small codes to do large tasks. Hence, you save time even while writing the code.
- Community: What if you get stuck while writing the code? You don't have to worry. Python community has one of the biggest and most active communities, where you can seek help from.

25) Briefly explain any 6 Operator & Description of the Tkinter Widgets.

Reference: https://www.tutorialspoint.com/python/python_gui_programming.htm

26) Briefly explain any 3 standard attributes of the Tkinter.

Reference: https://www.tutorialspoint.com/python/python_gui_programming.htm

27) What is the difference between a class and an object?

Reference: https://www.edureka.co/blog/interview-questions/oops-interview-questions/

Object	Class
A real-world entity which is an instance of a class	A class is basically a template or a blueprint within which objects can be created
An object acts like a variable of the class	Binds methods and data together into a single unit
An object is a physical entity	A class is a logical entity
Objects take memory space when they are created	A class does not take memory space when created
Objects can be declared as and when required	Classes are declared just once

28) Why do you need Regular Expression Functions in Python? Discuss at least 4 Python Regular Functions.

Reference: https://www.w3schools.com/python/python_regex.asp

RegEx Functions

The `re` module offers a set of functions that allows us to search a string for a match:

Function	Description
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string
split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

29) Discuss and list out at least five Pandas Series Attributes / Axes.

Reference: https://pandas.pydata.org/docs/reference/series.html

30) How to get the common items between two python NumPy arrays? Write a Python Program. How to get the positions where elements of two arrays match? Write a Python Program.

Reference: https://www.machinelearningplus.com/python/101-numpy-exercises-python/

```
# Example
import numpy as np
a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
np.intersect1d(a,b)
```

```
array([2, 4])
```

```
# Example
import numpy as np
a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
np.where(a == b)
```

```
(array([1, 3, 5, 7]),)
```

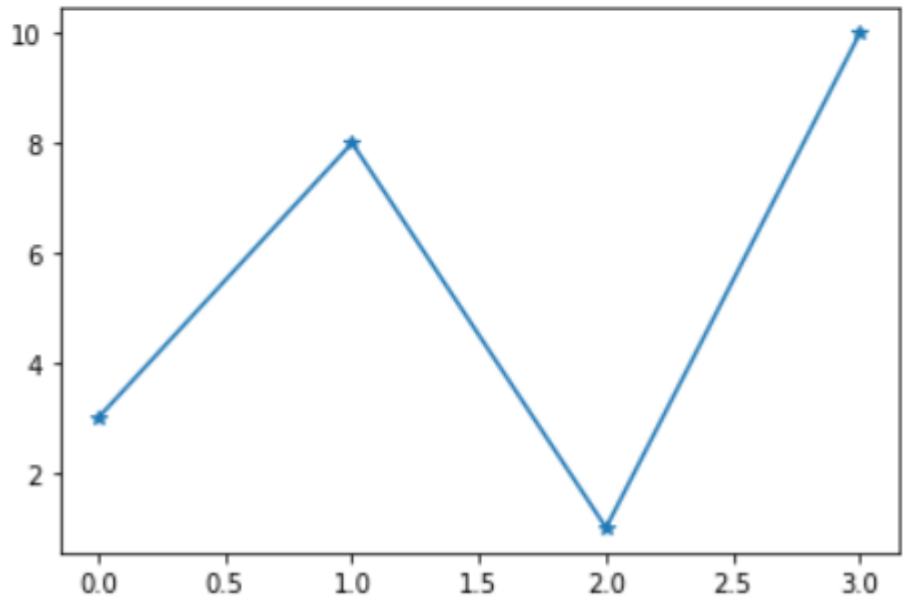
31) List out at least 10 markers in the Matplotlib? Explain any one marker with the program.

Marker	Description
'o'	Circle
'*'	Star
'.'	Point
','	Pixel
'x'	X
'X'	X (filled)
'+'	Plus
'P'	Plus (filled)
's'	Square
'D'	Diamond
'd'	Diamond (thin)
'p'	Pentagon
'H'	Hexagon
'h'	Hexagon
'v'	Triangle Down
'^'	Triangle Up
'<'	Triangle Left
'>'	Triangle Right
'1'	Tri Down
'2'	Tri Up
'3'	Tri Left
'4'	Tri Right
'|'	Vline
'_'	Hline

```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = '*')
plt.show()
```



32) Discuss at least 5 charts of the Data Visualization.

Write about the

1) Line Chart
2) Bar Chart
3) Pie Chart
4) Box Plots
5) Histogram
6) Scatter Plots etc.

Please refer to matplotlib references from the internet resources.

References: https://matplotlib.org/stable/gallery/index.html

33) Why is Web Scraping Used?

Reference: https://www.edureka.co/blog/web-scraping-with-python/#whywebscraping

Web scraping is used to collect large information from websites. But why does someone have to collect such large data from websites? To know about this, let's look at the applications of web scraping:

- Price Comparison: Services such as ParseHub use web scraping to collect data from online shopping websites and use it to compare the prices of products.
- Email address gathering: Many companies that use email as a medium for marketing, use web scraping to collect email ID and then send bulk emails.
- Social Media Scraping: Web scraping is used to collect data from Social Media websites such as Twitter to find out what's trending.
- Research and Development: Web scraping is used to collect a large set of data (Statistics, General Information, Temperature, etc.) from websites, which are analyzed and used to carry out Surveys or for R&D.
- Job listings: Details regarding job openings, interviews are collected from different websites and then listed in one place so that it is easily accessible to the user.

34) Discuss Geometry Management method in the Tkinter.

Reference: https://www.geeksforgeeks.org/python-geometry-method-in-tkinter/

https://www.tutorialspoint.com/python-geometry-method-in-tkinter

https://www.educba.com/tkinter-geometry/