

1. Is indentation required in python?

Yes, it is required in python, it is a four space characters. It is used to avoid error.

All codes within loops, classes, functions are specified in an indented block.

2. What is the difference between Arrays & Lists in python.

* Arrays and Lists are used for: Storing data.

* Arrays can hold only single type of data but list can hold any type of type data.

* Arrays are more Speed than List.

* In arrays, additions & Subtractions and other mathematical operations done easily.

But in lists only additions & Subtractions done easily.

* Array has dimension

Whereas List has No dimensions

3. What is the difference between range & Arange.

Range is an inbuilt function in python.

It is used for generate list of numbers (integers)

Arange is an Numpy function i.e np.arange()

It is used to generate array of numbers.

4. What is it a lambda function?

Lambda is an Anonymous function.

It don't have any keywords like def, print, written...etc return...etc

A lambda function can take any number of arguments, but can only have one expression.

Example :

~~a = lambda x,y : x+y~~

~~print(a(5,6))~~ a(5,6)

5. How does break, continue, pass Work?

Break : Condition

When Specific condition is met, break allows to terminate the loop and control transferred to the next statement.

Example:

for letters in PYTHON:

if letters == "H":

break

print(f' current letters {letters}')

Output:

Current letters 'P'

Current letters 'Y'

Current letters 'T'

Continue :

When Specific Condition is met, Continuous Continue allows Skipping Somepart of a loop and Control transferred to the beginning of the loop.

Example:

```
for letters in "PYTHON":
```

```
    if letters == "H":
```

```
        continue
```

```
        print(f'current letters : {letters}')
```

Output :

```
Current letters : P
```

```
Current letters : Y
```

```
Current letters : T
```

```
Current letters : O
```

```
Current letters : N
```

Pass :

It represents a null operation in python. It is an empty block of code. The purpose is execution of code without getting any errors used with function.

Example :

Output :

```
for letters in "PYTHON":
```

```
    Current letters : P
```

```
    if letters == "H":
```

```
        Current letters : Y
```

```
        pass
```

```
        Current letters : T
```

```
        print("This is pass block")
```

```
This is pass block
```

```
        print(f'current letters : {letters}')
```

```
        Current letters : H
```

```
        print("Good bye")
```

```
        Current letters : O
```

```
        Current letters : N. Good  
        Good bye. bye
```

6. Difference between
- (i) String & List
 - (ii) List & tuple
 - (iii) Sets & Dictionary
 - (iv) List & Series & arrays
 - (v) Tuple & sets

(i)	String	List
-----	--------	------

(a) String is immutable	List is mutable
-------------------------	-----------------

(b) For string, We use either a Single quotation ' or double quotation "	For List, We use a Square brackets
--	---------------------------------------

(c) String doesn't exist any methods like pop(), remove(), update(), append() etc.	List have a methods like pop(), remove(), update() etc
---	--

(ii)	List	Tuple
------	------	-------

(a) Lists are mutable	Tuples are immutable
-----------------------	----------------------

(b) List consumes less memory and more time complexity	Tuples consumes less memory and less time complexity
---	---

(c) For list, We use a Square brackets [].	For tuple ; We use paranthesis () .
--	--

(d) List provides many inbuilt methods.	Whereas tuples has less inbuilt methods.
--	---

(iii)

Sets

- (a) For empty set parenthesis () is used.
- (b) Don't have key, values.
- (c) No duplicates allowed.

Dictionary

For empty list only bracket {} is used having key & values.
Duplicates are allowed.

(iv)

Series

- (a) Indexing can be given to the elements manually.
- (b) Elements are arranged vertically.
- (c) Indexes are shown along with the elements.

Arrays

Indexing is by default from zero.
Elements are arranged horizontally.
Indexes are not visible in arrays.

(v)

Tuple

- (a) Empty } ()
- Non-empty }

Empty = Set ()
Non-empty = { }

- (b) Tuple is immutable

Set is mutable

- (c) Duplicates are allowed

No duplicates allowed

- (d) Order of elements

Inorder of elements

7. Python dictionary v/s list , Which is faster?

Dictionary is faster than list .

The average time complexity of a dictionary key lookup is $O(1)$ i.e in dictionary controller look for directly key and return its value whereas the average time complexity of lookup in a list is $O(n)$, i.e in list look all values and then return/print required value.

8. What is pickling and Unpickling ?

Pickling :

Any python object convert into a string representation then it dump into a file using dump function. This process is called Pickling.

Example :

```
import pickle
```

```
a = [1, 2, 3, 4]
```

```
pickle.dump(a, 'wb'(or) 'db')
```

Unpickling :

The process of retrieving original python objects from the string representation is called unpickling.

9. What does this mean `*args`, `**kwargs`? And Why Would We use it?

`*args`:

When We don't know, how many arguments are going to pass into a function is called `*args`.
`*args` are generally used with :list, tuples, strings.

Example :

Class A :

`def __init__(self, *args)`

`Self.Name = *args[0]`

`Self.Roll.No = *args[1]`

`Self.Age = *args[2]`

`**kwargs`:

When We don't know, how many keyword arguments are going to pass into a function is called `**kwargs`.

`**kwargs` are generally used in dictionary.

Example :

Class A :

`def __init__(self, **kwargs)`

`Self.Name = **kwargs["N"]`

`Self.Roll.No = **kwargs["R"]`

`Self.Fee = **kwargs["F"]`

Q10. What is monkey patching in python?

During the runtime, dynamic modification of a class (or) module is called as the monkey patching.

Example:

class A:

 def x(self): → function-1
 print("He is good")

 def y(self): → function-2
 print("He is bad")

monk.A.x = y

Obj = monk.A

Obj.x

Output:

"He is bad"

1. python ?

modification
called as the

→ function - 1

→ function - 2

11. What is a dictionary in python ?

- * Mapping of one to one relationship between keys and values is called dictionary in python.
- * Dictionaries contain pair of keys and their corresponding values.
- * In dictionary keys are the indexes.
- * Dictionary is built-in datatypes in python.

Example :

```
A = dict(name = "Vannika", age = 36, country = "India")
print(A)
```

Output :

```
{'name': 'Vannika', 'age': 36, 'country': 'India'}
```

12. Explain split(), sub(), subin() methods of 're' module in python ?

* re : regular expression in python.

* re-module : is used for modification of strings.
It provides 3 methods.

re.split() : It uses a regular expression pattern to split a given string into a list.

Example :

```
a = "She is a good girl"
```

```
re.split(a)
```

Output :

```
["she", "is", "a", "good", "girl"]
```

re.sub(): It differentiates between it finds all substrings where the regular expression pattern matches and then replace with different strings.

Example:

```
b = "She is *intelligent*"  
pattern = *good girl*  
re.sub(b, pattern)
```

Output:

```
b = ["She", "is", "good girl"]
```

re.subn():

It is similar to sub(). And also return the newstring along with the number of replacements.

Example:

```
b = "She is *intelligent*"  
pattern = *good girl*  
re.subn(b, pattern)
```

Output:

```
b = ["she", "is", "good girl", 1]
```

13. What are docstrings in python?

Docstrings are assigned to any within triple quotes. This generally used in modules.

It explains, who we can call them.

14. What are Indexes?

* Index represents

* They are represented by

1. Positive

2. Negative

Positive Index:

In the first position

Ex: A = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Negative Index:

-ve index

First position

Ex: A = [-7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Q. 13. What are docstrings (documentation strings) in python?

Docstrings are not actually comments and not assigned to any variable and it represents within triple quotes " " " ". This generally used inside the functions, classes and modules.

It explains, what operations happened/occurred. We can call them output using `doc`-function.

Q. 14. What are Indexes and why are they used?

along with

* Index represents the position of an elements

* They are represented in two ways.

1. Positive Index (+ve)

2. Negative Index (-ve)

Positive Index:

In +ve index, start from left.

First position is zero, Second position is 1...etc

Ex: $A = [0 \ 1 \ 2 \ 3 \ 4 \ 5]$

Negative Index:

-ve index starts from right.

First position is -1, Second position is -2....etc

Ex: $A = [-7 \ -6 \ -5 \ -4 \ -3 \ -2 \ -1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5]$

15. What are the built-in types of python?

Built-in types in python are as follows

1. Integers
2. Floating-point
3. Complex Numbers
4. Strings
5. Boolean
6. Built-in functions

16. What is the usage of help() and dir() function in python?

Help() function:

Used to display the documentation of modules, classes, functions, keywords etc

dir() function:

* It aims to produce the most relevant data rather than complete information.

* It returns valid list of attributes and methods of object.

17. How to remove va

Pop() method:

* If we give index the deleted position

Pop(Index)

* If we don't function by def the last element

Ex: fruits = ['apple',
fruits.pop()

Output:

banana

clear() method:

clear(), n

list

Ex: a = [7, 8, 9]

a.clear()

print(a)

Output: []

Remove() metho

The remove
occurrence of th

Example:

a = [1, 2, 1, 1]

a.remove(1)

print(a)

of python?

as follows

and dir()

mentation of
words etc

list - relevant data

attributes

classmate

Date _____
Page _____

classmate

Date _____
Page _____

17. How to remove values to a python array?

Pop() method:

* If we give index, it deleted and return the deleted position value.

Pop(Index)

Ex: * If we don't give index in pop() function by default it deleted and return the last element of that list

Ex: fruits = ['apple', 'banana', 'cherry']

fruits.pop(1)

Output:

banana

Clear() method:

clear(), remove all the elements in the list.

Ex: a = [7, 8, 9]

a.clear()

print(a)

Output: []

Remove() method:

The remove() method removes the first occurrence of the element with the specified value.

Example:

a = [1, 2, 1, 1, 4, 5]

Output: [2, 1, 1, 4, 5]

a = list a.remove(1)

[1, 2, 1]

print(a)

18. What is the purpose of 'is', 'not' and 'in' operators?

is:

Returns, When two operands are true.

Ex: $a = 10$

$b = 20$

$a \text{ is } b$

Output : False

Not:

Returns the 'inverse' of the boolean value.

Ex: $a = 10$

$b = 20$

$\text{not}(a == b)$

Output : True

in:

Checks if some element is present in some sequence.

Example:

Word = "Hello"

if 'e' in Word:

print ("The letter 'e' is present in the word")

Output:

The letter 'e' is present in the word.

19. What is the Immutable & Mutable :

It can

Example:

List, Set,

Immutable :

It can

Example:

String, tuple

20. How is Except

Try:

The block for Error.

Except:

Block

Finally:

Executed

Whether error or not

Else:

not and in:

and are true.

the boolean

's present

present in the Word")

the Word

19. What is the difference between mutable and immutable datatypes?

Mutable:

It can be edited

Example:

List, Set, Dict

Immutable:

It can not be edited

Example:

String, tuples, Integers, float, boolean

20. How is Exception handling done in Python?

Try: The block of code which is monitored for Error.

Except:

Block gets Executed When an error occurred

Finally:

Executed block of code, irrespective of whether error occurred or not

Else:

Ques 21. What is difference between $/$ and $//$ and $\%$?

$/$ (Forward slash) : $= \text{float}$

Represents division, returns the Quotient.

Ex: $7/2 = 3.5$

$//$: $= \text{integer}$

Represents floor division, returns the floor value (integer) of quotient.

Ex: $7//2 = 3$

$\%$: $= \text{remainder}$

Represents division, returns remainder value.

Ex: $7 \% 2 = 1$

Ques 22. What is PIP?

Ans: PIP: Python Installer Package

It is an acronym.

Which provides a Seamless Interface to install Various python modules.

Ques 23. Is python a language?

Python is b
Which means
first compiled

Ques 24. What are used?

Python Nan

(a) Local Na

* A l
class , a fu

* Objects declar
outside the

Example:

a='global a

y='global y

def test_nu

a='enc

def inner

a='l

print

print

inner na

print(a)

23. Is python a Compiled language or an Interpreted language?

Python is both Compiled and Interpreted language, which means when we run a python code, it is first compiled and then interpreted line by line.

24. What are Python namespaces? Why are they used?

Python NameSpaces are 3 types:

(a) Local Name Space: * Declares within the function.

* A local namespace is defined for a class, a function, a loop or any block of code.

* Objects declared in this namespace cannot be used outside the function.

Example:

```
a='global a'  
y='global y'  
def test_namespaces():  
    a='enclosing a'
```

```
def inner_namespace():
```

```
    a='local a'
```

```
    print(a)
```

```
    print(y)
```

```
inner_name_space()  
print(a)
```

→ Local

(b) Global Name Space :
Global Variables are declared outside the functions.

Example:

```
a = 'global a'  
y = 'global y'
```

→ Global

```
def test_namespace():  
    a = 'enclosing a'  
    def inner_namespace():  
        a = 'local a'  
        print(a)  
        print(y)
```

→ local

(c) Built-in Name Space :

Contains the names of built-in functions and objects of python.

Example:

Int, boolean, float, def, string,

len(), etc

25. What is List Comprehension?

List Comprehension:

It is a Single Line of loop code.

It is a Shorter Syntax When We Want to Create a new list based on the Values of an existing list.

Example:

```
newlist = [x for x in range(10)]  
print(newlist)
```

Output:

0,1,2,3,4,5,6,7,8,9

26. What is PEPs and Why is it important?

PEP = Python Enhancement Proposal

PEP 8 is a documentation with a set of guidelines about python language about the designing of python; more readable and usable for anyone

27. Show me three different ways of fetching every third item in the list?

- Yield function
- List Comprehension
- For loop

28. What is polymorphism in python?

Polymorphism means the ability to take multiple forms.

In programming it refers to methods/functions/operators with the same name that can be executed on many objects or classes.

Example:

```
class Person():
    def __init__(self, name, idnumber):
        self.name = name
        self.idnumber = idnumber
    def display(self):
        print(self.name)
        print(self.idnumber)
    def details(self):
        print("My name is", self.name)
        print("IdNumber:", self.idnumber)

class Employee(Person):
    def __init__(self, name, idnumber, salary, post):
        self.salary = salary
        self.post = post
    def __init__(self, name, idnumber):
        Person.__init__(self, name, idnumber)
    def details(self):
        print("My name is", self.name)
        print("IdNumber:", self.idnumber)
        print("Post:", self.post)
a = Employee('Vannika', 886012, 200000, "Intern")
a.display()
a.details()
```

Output : -

Vannika

886012

My name is: Vannika

IdNumber: 886012

Post : Intern

29. Define Encapsulation in python?

Encapsulation means binding the code and the data together.

Example:

Class Employee:

```
def __init__(self, name, Salary, project):
    self.name = name
    self.Salary = Salary
    self.project = project
    } → Data

    def Show(self):
        print('Name : ', self.name, 'Salary : ', self.Salary)
Methods of
    def Work(self):
        print(self.name, 'is working on', self.project)
Code

emp = Employee('Jessa', 8000, 'NLP')
```

emp.Show()

emp.Work()

Output:

Name : Jessa Salary : 8000

Jessa is working on NLP

30. What are the Generators in Python?

* Generators are functions that returns an iterator.

* It is a live normal function, except return keyword replaced by yield.

* If return-statement terminated entirely but yield-statement pause the function, continuous successive.

Example:

Output:

```
def Simple():
    for i in range(10):
        if (i % 2 == 0):
            yield i
for i in Simple():
    print(i)
```

0

2

4

6

8

31. What are Decorators?

* Extension of a existing function without any modification to the original function source code is called Decorators.

* Decorators allows us to modify or alter the function, methods and classes.

Example:

```
def uppercase_decorator(func):
    def wrapper():
        result = func()
        return result.upper()
    return wrapper
```

```
def say_hello():
    return "Hello, World!"
print(say_hello())
```

Output:

HELLO, WORLD!

32. What is a python module? How is it different from libraries?

Module

1. It is collection of codes, function, def, variables called modules

2. Module has a .py extension file.

Ex: DATETIME, RESEX, Random

Libraries

1. It is collection of reusable functionality of codes to perform variety of tasks.

2. Libraries hasn't any extension file.

Ex: Numpy, pandas, Matplot, Seaborn

33. Explain difference between Methods and functions

Method

- * Method always defines inside a class
- * Methods are dependent on the class
- * Method has "Self" argument
- * Methods called with an object directly, we can't call by name.

Functions

- We don't need class to define function.
- Function are independent entities of a program
- It doesn't have any "self" argument.
- Functions directly called by its name.

Example:

```
class person :
```

```
    -init-(self, Age, Name)
```

```
        self.Age = age
```

```
        self.Name = Name
```

```
    def Name_Age(self):
```

```
        print (self.Name)
```

```
        print (self.Age)
```

```
P = person(22, "Vannika")
```

```
P.Name_Age()
```

Example:

```
def even_odd(st):
```

```
    for i in st :
```

```
        if i % 2 == 0
```

```
            print ("even numbers", i)
```

```
        else
```

```
            print ("odd numbers", i)
```

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
even_odd(a)
```

34. What are Compound data types & data structures?

Compound data types:

Compound data type is built by combining primitive data types.

→ It can hold only values not the data.

→

Examples:

Strings, Integers, Floats, Booleans, Sequence.

Compound data structures:

Compound data structures are formed by combining one or more data types.

→ Allows us to store multiple observations.

→ It can hold different kinds of data within a single object.

Examples: Stack, Queue, Linked List, Binary Trees, Graph

35. Explain Zip() & Enumerate() function.

Zip()

* Takes multiple lists as input

* Zip returns the list of tuples

*

Enumerate()

* Take one list as input

* Returns the list of tuples

* It is generate index

Zip():

Example :

```
a = ["John", "Charles", "Mike"]
```

```
b = ["Jenny", "Christy", "Monica"]
```

```
x = zip(a,b)
```

```
print(tuple(x))
```

Output :

```
[('John', 'Jenny'), ('Charles', 'Christy'), ('Mike', 'Monica')]
```

38

Enumerate():

Example :

```
x = ['apple', 'banana', 'cherry']
```

```
y = enumerate(x)
```

```
print(list(y))
```

Output :

```
[(0, 'apple'), (1, 'banana'), (2, 'cherry')]
```

36. Explain the difference between print & return?

Print doesn't store any value, it simply prints. Whereas, return stores the value and then it prints.

37. Explain about class & object?

Class:

class is a blueprint of an object.

That defines the nature of future object

39

Object :

Object is an instance of class.
So, classes can construct instances of objects.
This is known as Instantiation.

38. Explain about apply() function?

It is used to apply for a particular condition on column of data frame like loop iteration.

Example:

```
import pandas as pd  
def cal_sum(x):  
    return x.sum()  
data = {'X': [50, 40, 30], 'Y': [300, 1112, 42]}  
df = pd.DataFrame(data)  
X = df.apply(cal_sum)  
print(X)
```

Output:

```
X 120  
Y 1454
```

39. Explain about Conditional flows & control flows?

Conditional flow:

It performs set of rules, if certain conditions met. The conditional flow is to control the execution of block of code, if the statements criteria match or not.

Examples:

if

elif

nested if

with else

Control flow:

Control flow is the order in which the code is executed, it regulates by conditional statements and loops

40. What are python iterators?

→ Iterators are objects in python, i.e used to iterate over iterable objects like lists, tuples, dicta dictionaries and sets

→ The iterator object is initialized using the iter() method. It uses the next() method for iteration.

Example:

```
mytuple = ("apple", "banana", "cherry")
```

```
myit = iter(mytuple)
```

```
print(next(myit))
```

```
print(next(myit))
```

```
print(next(myit))
```

Output:

apple

banana

cherry

41. What is `__init__`?

`__init__` is a method or constructor in python. This method is automatically called to allocate memory. When the new object (or) instance of a class is created. All classes have the `__init__` method.

Example:

```
class Person:
```

```
    def __init__(self, name, age):
```

```
        self.name = name
```

```
        self.age = age
```

```
p1 = Person("John", 36)
```

```
print(p1.name)
```

```
print(p1.age)
```

Output:

```
John
```

```
36
```

42. What is difference between deep copy & Shallow Copy?

Shallow Copy Deep Copy

- * In Shallow Copy, a copy of the original object is stored and only the reference address is finally copied.

- * Shallow copy is faster than Deep copy.

- * In deep Copy, the copy of the original object and the repetitive copies both are stored.

- * Deep copy is slower than shallow copy.

Shallow Copy

- * The changes made in the copied object also reflect the original object.

- * Syntax:

copy.copy(x)

- * It stores references of the object in the main memory.

Deep Copy

- * There is no reflection on the original object when the changes are made in the copied object.

- * Syntax:

deepcopy(x)

- * It stores copies of the object values.

43. What are functions in python?

A function is a block of code which is executed only when it is called.

To define a python function the "def" keyword is used along with print and return.

It avoids the repetition of code.

Example:

```
def my_function():
    print("Hello from a function")
my_function()
```

Output:

Hello from a function

44. What is the difference between Indexing & Slicing?

Indexing is Extracting / lookup : on a particular value in a data structure, Whereas Slicing retrieves a Sequence of elements.

45. How do map, reduce and filter function Works?

MAP FUNCTION:

Map function applies the given function to all the iterable and returns a new modified list. It applies the same function to each element of a Sequence.

Example :

numbers = [1, 2, 3, 4, 5]

Squared numbers = list(map(lambda x: x**2, numbers))
print(squared_numbers)

Output:

[1, 4, 9, 16, 25]

REDUCE FUNCTION:

Reduce function applies the same operation to items of a Sequence. It uses the result of operations as the first param of the next operation. It returns an item and not a list. This function is defined in "functools" module.

Example:

The `reduce()` function in python is a part of the `functools` module and doesn't return multiple values ; it just returns a single value.

Example:

```
from functools import reduce
```

```
nums = [1, 2, 3, 4]
```

```
ans = reduce(lambda x, y : x + y, nums)
```

Output with `print(ans)` here will give 10

Output:

10

: output 10

FILTER FUNCTION:

Filter Function filters an item out of a Sequence . It is used to filter the given iterable (list, sets, tuple) With the help of another function passed as an argument to test all the elements to be true or false . Its output is a filtered list .

Example:

```
Seq = [0, 1, 2, 3, 5, 8, 13]
```

```
result = filter(lambda x : x % 2 != 0, seq)
```

```
print(list(result))
```

Output:

[1, 3, 5, 13]

Q6. What is Regex? List some of the important Regex functions in python?

Regular Expression or RegEx is a Sequence of Characters that are used to Create Search patterns. In python, the following RegEx functions are mostly used:

Match(): It checks for a match only at the beginning of the String.

Search(): It locates a Substring matching the RegEx pattern anywhere in the String.

sub(): Searches for the pattern and replaces with a new value.

Split(): It is used to split the text by the given RegEx pattern.

findall(): It is used to find all the Sub-strings matching the RegEx pattern.

Q6. What is Regex? List some of the important Regex functions in python?

Regular Expression or RegEx is a sequence of characters that are used to create search patterns. In python, the following RegEx functions are mostly used:

Match(): It checks for a match only at the beginning of the string.

Search(): It locates a Substring matching the RegEx pattern anywhere in the string.

Sub(): Searches for the pattern and replaces with a new value.

Split(): It is used to split the text by the given RegEx pattern.

findall(): It is used to find all the sub-strings matching the RegEx pattern.

Q.7. What are the uses of Numpy?

Uses of Numpy:

- An alternative for lists.
- Maintain minimal memory.
 - * It uses function like view, copies
 - * Help us in Saving lot of memory.
- Multidimensional Array:
 - * Multiple rows and columns.
 - * Easily Create matrices using multi-dimensional arrays.
 - * That are easily understand by ML models & Algorithms.
- Mathematical Operations.
 - * Like linear algebra, Arithmetical Operations, String operations.
- Array applications.
 - * We can implement zero's (0), one's (1's) and different matrices.
- Numpy with pandas:
 - * For faster Computational operations & Scientific Computational operations.

→ Numpy with Matplotlib

* Matplotlib is module of Numpy.

* It is used for graphical representation.

→ Numpy with Scipy

→

48. What is the difference Series & Vectors?

Vectors:

It can only assign index positions
Values as 0, 1, 2, ..., (n-1)

Series:

It has only one column. It can assign Custom
index positions Values that are for each and
every data Series.

Examples: cust_ID, cust_name, sales

Series can be Created from the list, array, dict

Example:

import pandas as pd

a = [1, 7, 2]

myVar = pd.Series(a)

print(myVar)

Output:

0 1

1 7

2 2

49. What is the difference between data-frames and matrices?

Dataframes

- * Data frames are a collection of Series that share a common Index.

- * It can hold multiple Series, which are of different data types.

Matrices

- * A matrix in numpy is constructed with multiple "vectors".

- * It can hold only one data type in the entire two-dimensional structure.

Example :

```
import pandas as pd
```

```
data = {
```

```
    "calories": [420, 380, 390],
```

```
    "duration": [50, 40, 45]
```

```
}
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

Output :

	calories	duration
0	420	50
1	380	40
2	390	45

	calories	duration
0	420	50
1	380	40
2	390	45

50. What is the difference between `.iloc` and `.loc`?

`.iloc`

- * We referred only to indexes
- * It only works for indexes

Example:

`Stores.iloc[0:9]`

Will return the rows with 0, 1, 2, 3, 4, 5, 6, 7, 8 as the indices.

→ It works in the same manner as python range function works; i.e. the last element is not included.

`.loc`

- * We referred either index or label
- * It only works for label

Example:

`Stores.loc[0:9]`

Will return the rows with 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 as the indices.

→ Here, the upper bound is included.

51. What is the difference between Merge, Join and Concatenate.

Merge

- * Default inner join

Join

- * Default outer join

Concatenate

- * Default outer join.

Merge: Merge is used to merge the data frames using the unique column identifier.

Syntax:

```
Pd.merge(df1, df2, 'outer', on = 'custId')
```

Example:

df1	df2
customer_id	customer_id
customer_name	customer_name
customer_email	customer_email
customer_address	customer_address
customer_dob	customer_dob
customer_phone	customer_phone
customer_id	customer_id
customer_name	customer_name
customer_email	customer_email
customer_address	customer_address
customer_dob	customer_dob
customer_phone	customer_phone

merging customer details of both the student into one row

customer_id int
customer_name str
customer_email str
customer_address str
customer_dob str
customer_phone str

Join :

Join is used to join the data frames using the unique index.

Syntax :

df1.join(df2)

Example :

Concatenate :

It joins the dataframes basically either by rows or columns.

Syntax :

pd.concat (df1, df2)

Example :

52. When do we use Crosstab & Pivot-table?

Crosstab: It is similar to pivot table.

* The Crosstab function can operate on Numpy arrays, Series or columns in a dataframe.

* Pandas does that work behind the scenes to count how many occurrences there are of each combination.

* Why even use a Crosstab function?

The short answer is that it provides a couple of handy functions to more easily format and summarize the data.

* CrossTab, Values won't work unless aggfunc is explicitly mentioned and vice-versa

* Gives Value Error:

1. Values cannot be used without an aggfunc.
2. aggfunc cannot be used without Values.

Syntax:

```
pd.Crosstab(index=df[], columns=df[], values=df[],  
aggfunc=)
```

- In a PIVOT TABLE: we add the results

- * If don't give explicitly values to pivot table then by default it takes all the numerical columns as values.
 - * Average is by default operation that pivot table works on.
 - * aggfunc automatically works without even mentioning the values in the values argument and can be used without explicitly passing values.

Syntax: a set of constraints which to obey

1. df.pivot_table(index = [], columns = [], values = [])

2. pd.pivot_table(df, index = [], columns = [], values = [])

53. How do you use groupby?

Groupby allows to group rows together based on a column and performs the aggregate function on those combined rows.

Example:

```
df.groupby('company').mean()
```

54. What are ways to reshape a pandas dataframe?

There are three ways of reshaping the dataframe:

Stack(): reshaping via Stack() Converts the data into Stacked form. that is the columns are stacked row wise

Unstack(): is reverse of Stacking. This function is used to unstack the row to columns.

melt(): The function is used to manage the data frame into a format where one or more columns are identifier variables.

55. How to find missing values/Null values in pandas?

In order to check missing values in pandas data frame.

We use a functions .isnull() and .notnull()

Both functions helps in checking the values is NaN or not.

Ex: df.isnull()

It Shows true when null value exists, otherwise shows false

Ex: df.notnull()

It shows false when null value exists
Otherwise shows true.

56. How to fill missing values in python?

→ Delete the record missing values for huge dataset.

→ Fill the null values with forward filling, backward filling, mean, median, mode.

→ Create a Separate model to handle missing values.

1. Forward filling :

Fill the Value with nearest above Value.

2. Backward filling :

Fill the Value with nearest below Value.

3. Mode :

For Categorical data, we use mode.

4. Mean :

If data have no outliers fill with mean.

5. Median :

If data have outliers fill with median.

57. List Some of the Categorical distribution plots.

Distribution Plots :

Displot : Figure-level interface for drawing distribution plots onto a face grid.

Histplot : Plots univariate and bivariate histograms to show distribution of datasets

Kaleplot : Plots univariate or bivariate distributions using kernel density estimation.

Categorical Plots :

- **Catplot** : Figure-level interface for drawing Categorical plots onto a `FacetGrid`.
- **Stripplot** : Draws a scatterplot where one variable is categorical.
- **Swarmplot** : Plots a categorical Scatter plot with non-overlapping points.
- **Boxplot** : Plots a box plot to show distributions with respect to categories.
- **Violinplot** : Plots a combination of boxplot and kernel density estimate.
- **Boxenplot** : Draws an enhanced box plot for larger datasets.
- **Pointplot** : Shows the point estimates and confidence intervals using scatterplot glyphs.
- **Barplot** : Shows the point estimates and confidence intervals as rectangular bars.
- **Countplot** : Shows the counts of observations in each categorical bin using bars.

58. What is a Scatterplot?

A Scatterplot is a two-dimensional data visualization that illustrates the relationship between observations of two different variables. One is plotted along the X-axis and the other is plotted against the Y-axis.

59. What is the difference between regplot(), lmplot() and residplot()?

regplot():

regplot() plots the data and a 'Linear regression model' fit. It will show the best fit line that can be drawn across, given the set of all observations.

lmplot():

lmplot() plots the data, and the regression model fit across a FacetGrid. It is more computationally intensive and it is intended as a convenient interface to fit regression models across conditional subsets of a dataset. lmplot() combines regplot() & Facetgrid.

residplot():

residplot() plots the errors or the residuals between X and Y, creating a linear regression equation for the same.

60. What is the difference between Stripplot () and Swarmplot () .

Swarmplot () → Stripplot () :

It plots a scatterplot where one variable is categorical. A Stripplot () can be drawn on its own, but it is also a good complement to a box plot in cases where one wants to show all observations and some representation of the underlying distribution.

Swarmplot () :

It is also used to plot a categorical scatter plot; however, it is with non-overlapping points here. Swarmplot () is similar to Stripplot (), but the points are adjusted (only along the categorical axis) not to overlap. It is a representation of the distribution of values; however, it does not scale well to large number of observations.

This style of the plot is sometimes called a 'bee swarm'.

61.

62.

63.

61. What is the purpose of density plot or kde plot? Where are these used?

A density plot visualizes the distribution of data over a continuous interval or time period.

A variation of the histogram density plot uses a kernel for smoothing the plot values. This allows smooth noise. The peaks of a density plot illustrate where the values are concentrated over the interval.

62. What is a Pairplot?

A pairplot shows the relationships for a pair of columns of the data. It creates a matrix of axes where the diagonals show the distribution of each variable with itself.

63. What is FacetGrid?

It is a multi-plot grid for plotting conditional relationships. It helps in visualizing the distribution of one variable and the relationship between multiple variables separately within subsets of your dataset using multiple panels. It maps the dataset into multiple axes arrayed in a grid of rows and columns that correspond to levels of variables in the dataset.

Ques 64. How is Violinplot() different from Boxplot()?

A box plot (or) Whisker plot shows the distribution of quantitative data that helps compare between variables or across levels of a categorical variable. A boxplot is the visual representation of the statistical five-number summary of a given dataset.

The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution except for points that are determined to be "outliers" using a method that is a function of the inter-quartile range.

A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one categorical variables such that those distributions can be compared.