

* What is Python?

→ Python is high-level, interpreted general purpose programming language. It can be used to build almost any type of application with tools/ libraries provided. It supports objects, modules, threads, exception-handling which help in modeling real-world problems.

* What are the benefits of using Python?

→ Python is general-purpose programming language that has simple, easy to learn syntax that emphasizes readability & therefore reduces cost of maintenance. It supports high level data structures along with dynamic typing.

* What are lists and tuples?

→ Lists, tuples, sets are basic data structures in python language.

one of the differing point among them is mutability (which is the ability to change an object after its creation)

Lists are mutable

i.e. They can be modified

Lists are slower

than tuple

Syntax:

`li = [10, 'ch', 20]`

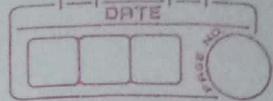
Tuples are preferred when user does not want data to be modified. Tuples use less memory.

`tup = (10, 'ch', 20)`

Faster

Tuples are immutable
(can't be modified)

Negative indexing is present in python only.



- * what is difference between Del , Remove and Pop on python lists
- The del keyword delete any variable , list of values from list

Syntax : `del list-name [index]` `del a[0]`
`del list-name` `del a`

The remove() method removes first matching value from list.

`a = [1, 2, 2, 3, 3]` `remove(2)` ⇒ `a = [1, 2, 3, 3]`
The pop() deletes value at particular index.

It returns deleted value

`rem = a.pop(3)`

- * What is decorator? Explain with example?
- A decorator is just a function that takes another function as an argument, add some kind of functionality f then returns another function. All of this without altering source code of original function that you passed in.

eg: `def decorator_func(func):`
 `def wrapper_func():`
 `print("wrapper func worked")`
 `return func()`
 `print("decorator_func worked")`
 `return wrapper_func`.

`def show():`
 `print("show worked")`.

`decorator_show = decorator_func(show)`
`decorator_show()`

* Difference b/w list and dict comprehension

→ List comprehension

[expression for item in iterable if condition]

Dict comprehension

{key : value for (key, value) in iterable if conditional}

* How memory is managed in python?

→ Mem. manag. in python involves a private heap containing all python objects & data structures. Interpreter takes care of python heap & programmer has no access to it.

The allocation of heap space for python objects is done by python memory manager. Python also has a build-in garbage collector which recycles all the unused memory.

The gc module defines functions to enable/disable garbage collector

* Difference b/w Generators and Iterators

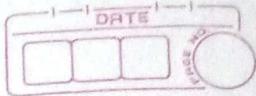
→ Generators are

iterators which can execute only once

An iterator is an object which contains countable no. of values & it is used to iterate over iterable objects like list / tuple sets etc.

Generator uses yield keyword

Iterator uses iter() and next() functions



Generator :

```
def sqr(n):  
    for i in range(1, n+1):  
        yield i*i
```

```
a = sqr(3)
```

```
print(next(a))
```

```
print(next(a))
```

```
print(next(a))
```

Output : 1

4

9

In gen., we cannot
use same value
used using next()

Iterator

```
iter_list = iter(['A', 'B', 'C'])  
print(next(iter_list))  
print(next(iter_list))  
print(next(iter_list))
```

Output : A

B

C

* What is init keyword in python

→ There is __init__.py file and __init__() function.
The __init__.py file is in package, file lets the
python interpreter know that a directory contains
code for python module.

The __init__() is similar to constructors in C++, Java
constructors are used to initialize object's state
class Person:

```
def __init__(self, name):  
    self.name = name
```

* DIFF b/w Modules and packages in python

→ module;

It is a simple python file that contains collections of functions & global variables and having a .py extension file.
import numpy.

Package:

The package is a simple directory having collections of modules. It contains python modules & -init-.py file by which interpreter interprets it as package.

* What are in-built datatypes in python or explain mutable & immutable datatypes

→ A first fundamental distinction that Python makes on data is whether or not the value of an object changes.

- If value can change, then object is mutable
- If value cannot change, object is immutable.

Datatech

Boekan (Boo)

Anterior

Floar

String

tuple

Frozensekt

111 +

set

dict

Mutable / Immutable

Immutable

Immutable

Immutability

Immutable

Immutability

Immutability

Mutable

Mutabile

Mutalle

- * Ternary operator in python
- The syntax for python ternary statement is as follows:
[if_true] if [expression] else [if_false]

$$\underline{\text{age}} = 25$$

discount = 5 if age < 65 else 10

print (discount)

local

1. It is declared inside function

2 If it is not
initialized,
garbage value is
stored.

3. When the value of local var. is modified in one fun. changes are not visible in other functions

If it is not initialized
zero is stored as
default

Value of global var. is modified in one fun. than changes are visible in rest of program

* Explain break, continue and pass statement.

break: When it is used inside the loop, it will terminate the loop and exit.

continues : It will stop current execution when used inside loop, and control will go back to start of loop

pass: It is a null statement, when interpreter comes across pass statement, it does nothing

- * what is self keyword in python
→ The self parameter is a reference to the current instance of class, and is used to access variables that belongs to class
- * What is pickling & unpickling
→ pickling : pickle module accepts any python object, transforms it into string representation and dumps it into file by using dump function. function used is pickle.dump()
unpickling : reverse of pickling.
- * what does * args and **kwargs means
→ When we are not clear how many arguments we need to pass to a particular fun. then we use * args and **kwargs.
The *args keyword represents varied no. of arguments. It is used to add together values of multiple arguments.
The **kwargs represents an arbitrary no. of arguments that are passed to functions.
**kwargs keywords are stored in dictionary. You can access each item by referring to keyword you associated with an argument.

```
def sum(*args):  
    total = 0  
    for a in args:  
        total = total + a  
    print(total)
```

```
sum(1,2,3,4,5)
```

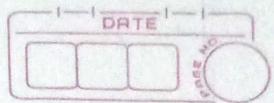
```
def show(**kwargs):  
    print(kwargs)
```

```
show(A=1, B=2, C=3)
```

- * "open" and "with" in python
 - both are used in file handling
 - with automatically closes the opened file
- * How exception is handled ?
- try : This block will test the exceptional error to occur .
- except : Here you can handle the error
- Else : If there is no exception then this block will be executed
- Finally : Finally block always gets executed either exception is occurred or not .

```
try :  
    # some code  
except:  
    # optional block  
    # handling of exception  
else :  
    # some code  
    # execute if no exception  
finally :  
    # some code (always executed)
```

- * What is pip
- Python pip is package manager for python packages
- We can use pip to install packages that do not come with python



* How to use f string and format or replacement operator

→ name = 'Nitin'

role = 'Developer'

print(f"Hello, my name is {name} and I'm {role}")

Format operator ⇒

print("Hello, my name is {} and I'm {}").format(name, role)

* Is multiple Inheritance possible in python

→ Yes

class A :

```
def abc(self):  
    print("a")
```

class B(A) :

```
def abc(self):  
    print("b")
```

class C(A) :

```
def abc(self):  
    print("c")
```

class D(B,C) :

pass

d = D()

d.abc

Output ⇒ b

- * Diff b/w .py file and .pyc file
- .py files contain source code of program
- .pyc files contain bytecodes of program
- python compiles .py files and saves it as .pyc files
- The .pyc contains compiled bytecodes of python source file. This code is then executed by python's virtual machine

- * Can you concatenate two tuples. If yes, How is it possible since it is immutable?
- Yes, we can concatenate two tuples using + sign

$$t_1 = (1, 2, 3) \quad t_2 = (7, 9, 10)$$

$$t_1 = t_1 + t_2$$

$$\text{print}(t_1) \Rightarrow (1, 2, 3, 7, 9, 10)$$

Tuples are immutable but here we didn't apply any changes to t_1 , we are referring t_1 as another reference for list after concatenation.

* Diff b/w list and array

List

1. List can store value of diff. types
2. List cannot handle direct arithmetic operations
3. The list are inbuilt D.S so don't need to import it
4. consumes large memory

Array

- It only consists values of same type
- It can handle arithmetic operations
- need to import array
- compact in memory

* What is `-a`, `--a`, `--a--` in Python?

→ `-a`

Python doesn't have real private methods so one underline in beginning of variable/method means its private for internal use only. also called as weak private

`--a`

It tells python interpreter to rewrite name in order to avoid conflict in subclass.

Interpreter changes variable name with class extension & that feature known as mangling

`--a--`

It is special method in Python

Python provides this methods to use it as operator overloading depending on user.

Python provides this convention to differentiate user defined function with module's function

* Multithreading and multiprocessing

→ Multithreading is a technique where multiple threads are spawned by process to diff. tasks at about same time, just one after another

In python, Global Interpreter Lock (GIL) prevents threads from running simultaneously

Multiprocessing is technique where parallelism in its truest form is achieved

* JSON

→ JSON = Javascript object Notation .
JSON is a data exchange format similar to XML . JSON is lightweight compared to XML
It is a data exchange format

* -- name -- = "main"

→ It is to know in which file we are , we can say it is an entry point.
If we directly in a file print -- name .
It will give main as output , but if we import any file into another and in second file we used a function which was defined in first file with -- name -- in it . It will give -- name -- as first file's name .

*

→ Exceptions are occurred errors detected at time of program execution

Try :

```
div=int(x) / int(y) → critical statement
except Exception as e:
    print("Exception occurred", e)
    div=None
print("division is:", div)
```

* Self
→ self means class itself

* Lambda
→ lambda is an anonymous function in Python
they can accept any no. of arguments but can only
have single expression

mul = lambda a, b : a*b

* Join, split
→ ' '.join(string-list)

string1 = string0.split(' ')

* Built-in datatypes -
Numbers

List

Tuple

String

Set

Dictionary

Boolean

- ~~on system~~
Pandas is python library used for working on data sets
It has functions for analyzing, exploring & manipulating data
Pandas → Panel data / Python data analysis

Pandas

- Pandas stand for Panel data & is core library for data manipulation & data analysis
- It consists of single & multiple dimensional data structures for data manipulation

Pandas series object :

```
import pandas as pd  
s1 = pd.Series([10, 20, 30, 40, 50])
```

s1

output ⇒

0	10
1	20
2	30
3	40
4	50

To change index :

```
s1 = pd.Series([10, 20, 30, 40, 50], index=['a', 'b', 'c', 'd', 'e'])
```

Series using dictionary

```
d1 = {"a": 10, "b": 20, "c": 30}  
pd.Series(d1)
```

output ⇒

a	10
b	20
c	30

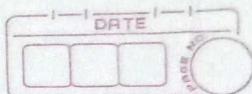
Extracting individual element

```
s1 = pd.Series([1, 2, 3, 4, 5, 6, 7])  
s1[3]
```

s1[-3:] ⇒

1	4	5
2	5	6
3	6	7

Seru → single dimensional arrays



Adding scalar values

$s1 + 5$

Output ⇒

0	6
1	7
2	8
3	9
4	10
5	11

Data frame → 2-dimensional tabular data structure

import pandas as pd.

```
pd.DataFrame({ "Name": ["Bob", "Sam", "Anne"],  
               "Marks": [76, 25, 92] })
```

Output ⇒

	Name	Marks
0	Bob	76
1	Sam	25
2	Anne	92

df.iloc[0:3, 0:2]
↓ ↓
rows columns

exclusive

~~it's drop~~ To drop columns ⇒
df.drop('column_name', axis=1)

↑
denotes it is column to be dropped

To drop rows ⇒

df.drop([1, 2, 3], axis=0)

↑
denotes they are rows to be dropped

Reading CSV files :

```
import pandas as pd  
df = pd.read_csv('data.csv')  
print(df)
```

To set maximum no of rows after reading csv file :

```
pd.options.display.max_rows = 9999  
df = pd.read_csv('data.csv')  
print(df)
```

Reading JSON files :

```
df = pd.read_json('data.json')
```

To remove empty cells :

Best way to do this is to drop rows that contain empty cells.

```
new_df = df.dropna()  
df.dropna(inplace = True)
```



Will not return new DF, replaces orig. DF.

Replace empty values :

```
df.fillna(130, inplace = True)
```

Replacing for specified column :

```
df['calories'].fillna(130, inplace = True)
```

Setting a value for particular row & column :

```
df.loc[7, 'duration'] = 45
```

To discover duplicates ⇒

```
print(df.duplicated())
```

To remove duplicates

df.drop_duplicates(inplace = True)

To create CSV files

df.to_csv('name.csv', index = False)

* Tkinter →

Currently 15 widgets

GUI libraries, namely, PyQt5, Tkinter

* Steps to use Tkinter

1) Import Tkinter module
from tkinter import *

2) Create object / main window

main_window = Tk()

3) Run an infinite mainloop

main_window.mainloop()

* Widgets are controls in GUI app:

1) Label

Button

Entry

Checkbutton

Canvas

Frame

* Geometry Manager classes

1) pack() - Place widget on top

2) grid() - widget in table like structure

3) place() - widget at specific position

1. Label

→ `l1 = Label(window, text = "Sample")
l1.pack() or l1.place(x = 50, y = 100)`

To show photo →

`i1 = PhotoImage(file = "path/photo.png")
l1 = Label(window, image = i1)
window.mainloop()`

2. Button

→ `b1 = Button(window, text = "sample")
b1.pack()`

Entry : To take input from user.

`e1 = Entry(window, width = 20, font = ('calibri', 20))`

3. `cb = Checkbutton(window, text = "Male")
cb.pack()`

4. Frame

→ can hold diff widgets

5. Listbox

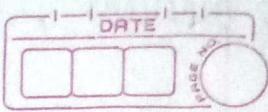
→ give a user list of options
`lb = Listbox(window, width = 20)`

`lb.pack`

`l1 = ['Tony', 'Edwin', 'Goddarji']`

For i in l1:

`lb.insert(END, i)`



Voice Recognition Chatbot -

Audio enabled voice chatbot based on Python and for database manag. used. SQLite studio.

* Problem Statement -

1. Business that struggle with customer engagement can often utilize digital transformation solutions to help discover more info about what market needs from them.
2. Customers sometimes don't know where to find the info. they need and requires customer care support.

* To solve this :

1. Chatbots can be one part of this soln.
2. Chatbots can be trained for frequently asked questions and can answer customers for seeking the info they need.
3. Chatbot makes conversation with Humans by identifying sentences & making decisions to respond to answer a question.

Chatbot can interact with users in two ways ↗

- 1) Text type chatting
- 2) Voice chat

1. The system is GUI based app.
2. System analyzes the question & then answers the q.

used waterfall testing model

Eg. of linear sequential model

Each phase must be completed before next phase begins

using model bcoz?

1. Requirement clear & fix & may not change.
2. There are no ambiguous req.
3. Risk is minimum.
4. Project is short.

Reanalysis → System design → Implementation → Testing → Deployment



Maintenance

* There were 2 modules

1) Train Bot :

Here only registered users can access this feature.
DB was used where all credentials & Q/A's were stored.

2) Use Bot :

Anyone can access & talk with Bot.

Technologies used :

- Tkinter
- SOLITE STUDIO
- chatterbot

chatterBot - It is python library that makes it easy to generate automated responses to user's input.

chatterBot uses a selection of machine learning algo's to produce

Input statement provided



Select a known statement that most closely matches input.
Return known response to selected match ↗

* Audio enabled feature -

Text to speech library from python

Pyttsx3 is text to speech conversion lib in python
Errors in changing voices, Rate & volume. and importing library

* Speech Recognition :

Pip install SpeechRecognition as sr

r = sr.Recognizer()

with sr.Microphone() as source :

audio = r.listen(source)

Limitations of project :

This soft. is not hosted anywhere .

Adv's of project :

usefulness, Reliability, Accuracy, Usability,

* Performance analysis :

Used Boundary value Testing, design test cases by equivalence partitioning

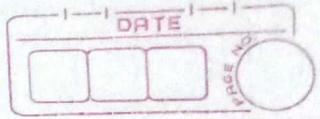
1. Diff b/w C & Python
2. Different types of datatypes in python
3. Diff b/w List & tuples
4. what is linked list
5. Types of linked list
6. What is constructor
7. what is data structure
8. what is deadlock
9. Diff types of searching algorithm / sorting algo's
10. Diff b/w insert & delete
11. Stack & Queue
12. what is SDLC

four major OOP concepts in C++, real life eg.

What is header file

What is function

- 1). what are features of language that you specify,
if diff b/w C++, Python
- 2) what is pointers
- 3) what are OOPS concepts



Several reasons -

- 1) more than 2.5 employees
- 2) greatest contributor to country's economy
- 3) 4th Indian comp. to cross 100 billion in market capitalization
- 4) From my linkedIn connections of seniors I came to know about learning programs of trainings your company provide, hackerrank you organize.

* Where do you see yourself in 2/5 years?

→

My first priority is to join this company and after four years, I want to see myself more stable more knowledgeable & more responsible with good career growth where I can give my 100% to your organisation and family, by doing this, I would be more motivated & experienced if I will encourage my juniors to give their best.

* Why should we hire you

→ As a fresher, I think I'll be a great fit for this role, since I am a quick learner & highly passionate about pursuing a career in this industry. I have participated in various co-curricular activities college events, here I get various opportunities to interact with people, present my idea, lead my society & basically throughout this journey I have nourished my communication skills very much.

I have technical knowledge, practically I have worked on this programming concepts of different projects. As I am currently interning at a I know am also implementing my practical knowledge. If I am given a chance I would definitely build & work towards my skills & benefit to the organisation.

* Why do you want to join our company?

→ The major reason for my desire to work for your company is that your organization has set a benchmark in the field of technology. More than 2.5 lakh employees are happily employed. It is our greatest contributor to country's economy.
~~What do you~~ From my LinkedIn connections & seniors, I came to know about learning programs, training that company provides. It will be a pleasure to work with such a MNC and as I am a fresher it would be a great chance for me to contribute towards the to your organization.

* What do you know about Infosys

→ Infosys is an Indian multinational information technology company that provides business consulting, information technology and outsourcing services. Founded on 2nd July 1981 by Mr. Narayana Murthy, Nandan Nilekani, current CEO is Sajil Parekh. and Headqu. is in Bangalore.

No. of employees are 2,59,615

Tagline is "Powered by intellect, driven by values."