0. Introduce yourself.

1. Tell me about Python?

2. Why Python is so popular now a days?

3. Features of Python

4. Advantages and Disadvantages of Python

5. Interpreted vs Compiled time programming languages. Explain in detail

6. .py vs .pyc files

7. How compilation will happen internally. Explain in detail

8. Why Python is Dynamically typed programming Language. Explain

9. Python is Platform independent.Explain

10.Different ways to write python program.

Interactive, IDLE, CommandPrompt, IDE

Advantages, Disadvantages

11.sourcecode vs bytecode

12.Register instruction set

13.High Level vs Low level programming Language

14.Python architecture

15.Explain Garbage Collection mechanism in detail.

Answer Introduction. Myself Akanksha Verma. I am a B.TECH graduate and completed my Engineering in CSE stream from Integral University, Lucknow,UP. After graduation,I joined ICICI Bank Pvt Ltd as Deputy Manager at Delhi Branch and worked there for almost 2 years.I used to resolve customer queries, assist my suboordinates, used to validate checks and used to complete my targets.So there I gained lot of corporate experiences. During that time I was also preparing for govt exams like UpSc. However, I couln't clear the prelims. Later I joined A coaching academy named Royal Academy and guided student and taught Maths there upto 10th level. Meanwhile, I post graduated that is completed MBA course in HR. Later, I got interested towards my core branch ie CSE and decided to move ahead in It sector. I completed my one year Internship in ITVedant Pvt Ltd and gained hands on experience in latest technologies like HTML5, CSS3, Javascript. It was mainly based on Python for web development. Learnt creating projects using Python web frameworks like Django,Flask,got understanding of ORM libraries. I had created a web app in Angular using components and services which is e-commerce web application. Also, I have created an Online School Mangament website using PHP, MySQL, Html,Css and Bootstrap.Thus,I gained various insights there regarding latest technologies.

I am an optmistic and punctual person. I dedicated towards my work and put my 100% effort in it. I am good as a team worker and can do well as a solo contributor too.

Ans 1. Tell me about Python?

Python is a high level programming language and highly demanding language. It is very easy to learn,use, understand and debug.There are Very few lines of code in

python, it uses simples English words which makes it easier to understand and work with. It can run on any os that is,

it is platform independent. It is interpreted language and integrates with other programming languages too.

It is open source, freely available and

It has large community support.

It has lots of libraries, packages and in built functions that needs to be just imported in our code, We do not need to create everything from scratch and

One can use its web frameworks like Django and Flask. Beacuse of these faetures,

It is widely used by programmers these days.

Ans 2.Why Python is so popular now a days?

a)**Dynamically Typed language** - we need not mention variable type before assigning values to a variable.

b)**Easier** - Easy to understand,use,learn and maintain- simple English words in code

c)**Integrated Language** - can integrate with other programming languages.

d)**Open source** - any one can download it and it is freely avaialble (no license is needed)

e**)Large community support** - On various websites, programmars solve and upload the code solutions so, whenever programmar faces problems, they can take help from such websites.

f)Very few lines of code as compared to other high level programming languages like JAVA, c

g)**Libraries and packages , in-built functions** available -We just need to import them in our code without building everything from scratch - good for web and app developments.

h)Error identification soon, - We can identify, and rectify errors soon.

i)Used by many renowned companies like Google, Facebook, Instagram

j)Used in - data sciece, ML, automation, GUI

k)Python is a web programming languages that is used to create massive websites or web applications like Google, Netflix, instagram.

1) Creating of game applications

2)Big data analytics

3)Useful in machine learning and AI

- ML

- data science

- GuI applications like (Tkinter,Kivy, PyQt)

- Image processing (like OpenCV, Pillow)

-Web scraping (like Scrapy, BeautifulSoup, Selenium)

-Test framworks

-multimedia

-Scientific Computing

-Text Processing

o**)Platform Independent** – It can on any OS, We do not need to change the code. Directly we can have .py files and execute it.

Ans 3.Features of Python

a)Easy to use, learn,understand and maintain

b)**Dynamically Typed language**- We need not specify data types before assigning value to varibles. Data type is needed at run time only

c)**Interpreted Language** - It uses interpreter for translation and has line by line execution

d**)Open source**- it is freely available on internet and any one download and use it.

e**)Large community support** - This is the reason programmers use Python a lot these days as problems are posted in community and solutions are put by other programmers. so, one can take help from there.

f**)Integrated language**- It can well integrate with other languages too.

g)**Libraries and packages** - It has various libraries and packages that we just need to import in our code and it makes coding easier. These can be used in web integration, data processing, n/w communication and h/w interaction. Various built – in objects are there like – lists, dictionaries.

h)Error Identification is soon and results are obtained faster

i)**Platform Independent** - It is flexible that is platform independent and can run on any OS.

j)Very few lines are needed as compared to other programming languages.

k)Simple English words are used.

n)**Front and backend** - With py script , we can write python codes in HTML using <py-script> and <py-env>. For backend we can use frameworks like Django and Flask.

o)**Allocating memory dynamically** - Variable data type need not to be provided. The memory is automatically allocated to a variable at run time when it is given a value.Developers do not need to write int y = 18 if the integer value 15 is set to y. You may just type y=18.

p)**Object-Oriented Language**- supports object oriented concepts like encapsulation, Inheritance

q)**GUI programming** - Graphical User interfaces can be made using a module such as PyQt5,Tkinter in Python. PyQt5 is the most popular option for creating graphical apps with Python.

r)**High level Language** - We do not need to manage memory and no need to remember system architecture.

Ans 4. Advantages and Disadvantages of Python

Advantages oF Python -

a)Dynamically typed language - we do not need to mention the data types of variable before assigning values to the variable.

b)Easy to learn, use, maintain and debug

c)Very few lines of code are there in python programming

d)Simple English words are used in code which makes it easier to program.

e)Integrated language - It can integrate well with other programming languages like c, c++

f)open source - It is freely available on internet and one can easily download it. No need of license for that.

g)Large community support - Many people are posting in community regarding debugging errors and problems that occur in day to day coding. It is helpful a lot for programmers and they can take help from various sites.

h)Libraries and packages - Availability of various packages, modules makes it a good language for web applications, games app development. It has web frameworks like Django and Flask which leads to easier, faster and maintainable code. We donot need to build things from scratch.

i)Error identification is soon - debugging is simple in Python. If source code has any error, it halts execution and throws error. If it doesn't has any error,it is converted into byte code.

j)OOPs concept – Python is an OOPs language with support of inheritance, encapsulation, classes and methods

k)General Purpose – Python is a general purpose language. And it is not domain specific. For Example –SQL is used to connect to database only.

Disadvantages -

a)Slow processing - Python is an interpreted language, that is, it processes each line of code in a sequence. Efficiency of the code affects the overall speed and performance. As the code becomes more complex, it's natural to overlook inefficient code, causing slower processing and decreased performance.

b)Low memory Efficient – Python code are tested are run time. So, any changes can be made that time also which makes the application slower, as issues in the application can slow down the whole application. Thus, python has become slower, and less efficient programming language. Python is not optimized to reduce memory. It can use the RAM ten times more compared to other languages. It is not optimized to reduce memory. So it is more memory intensive. Garbage collector also cannot gather all discarded resources immediately.

c) Not good for mobile app development - because of slow processing and not being memory efficient.

Python cannot be used for mobile web application development because of its speed issues. Since mobile phones has limited memory and processing capability, it needs to be designed in a framework which is faster and ensures smooth functionality.

d)Interpreted Language - It takes lot of time in execution of instructions as it executes line by line. Any changes can be made even at run time at run time

e)Low database connectivity – Python lacks powerful , easy to use interface like java database connectivity. It is more difficult to work with databases in python than in some other programming languages.

f)run time errors- . Python is not compiled until runtime, so errors appear only. As we can change the code dynamically so it is more prone to run time errors.

g)Not Very fast – It is slow in comparison to other programming languages like JAVA or C. Python is interpreted language and dynamically typed language, so the run time compiler has a lot to do. It has to validate the data types of each variable (everytime) which makes it slower.

Advantages and disadvantages of Python

|  |  |
| --- | --- |
| Beginner friendly. Easy to learn and use | Issues with support |
| Large community support | Slower than compiled languages |
| Flexible – can run on any os | Dynamically typed language- line by line- slow |
| Libraries and packages, built in functions | Run time Errors |
| Highly scalable | Slow processing |
| Open source- freely available | Low memory efficient |
| Integrated language – can integrate itself with other languages | Code security- not 100% secure. take steps for that – like QA testing |
| Debugging is easy | Not good for mobile application |
| Simple English words are there. So easy to understand | Database connectivity- problem with database connection. Python doesn’t have strong db interface like JDBC |
| Used in – data science, ML, Big data |  |
| General purpose language |  |
| OOPs concept is supported like Inheritance, polymorphism, classes, methods |  |

Ans5. Interpreted vs Compiled time programming languages. Explain in detail

a)Compiled programming languages are those which require compiler for translation whereas interpreted languages require interpreter for its translation

b)compiled languages can be directly executed by CPU, whereas interpreted languages cannot be directly executed by Cpu

c)compilation requires 2 steps while interpretation requires single step

d)compiled languages can be executed by target machine, but interpreted languages need translation of bytecode into machine code

e)compiled languages are faster as compared to interpreter

f)compiled languages has higher performance than interpreted language

g)compiler translates entire program into compiled file, whereas interpreter executes line by line.

h)Compiler debugs entire file at compilation time whereas all debugging is done at run time by interpreter

|  |  |
| --- | --- |
| compiler | Interpreter |
| Compilation is done before execution | Compilation and execution takes place simulatneously |
| Faster | Comparatively slower |
| Memory requirement is more due to creation of object code | Less memory is required |
| Displays all errors after compilation, all at same time | Displays errors of each line one by one |
| difficult | Easier comparatively |
| Takes entire program at one go | Takes a single line of code and executes it |
| Generates intermediate code | No intermediate code is generated |
| Best suited for production environment | Best suited for software development |
| More efficient | Less efficiency |
| Converts entire program into m/c code, when all syntax errors are removed, execution takes place | Each time program is executed every line is checked for syntax errors and converted into equivalent m/c code |
| Slow for debugging | Debugging is easy |
|  | Allows memory management automatically which reduces memory risks |
|  | More flexible than compiled langauages |
| Examples – C,C++, JAVA | Examples – PHP, Python, perl |

Ans 6. py vs .pyc files

a).py files contain the source code of a program. Whereas, .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

b)Before executing a python program python interpreter checks for the compiled files. If the file is present, the virtual machine executes it. If not found,

it checks for .py file. If found, compiles it to .pyc file and then python virtual machine executes it.

c)Having .pyc file saves the compilation time.

.py files are source code that is human readable form whereas .pyc files are bytecode and not human readable.Bytecode is intermediatory between source code and machine code

source code is changeable; but byte code cannot be changed directly

source code has .py extension whereas bytecode has .pyc extension

|  |  |
| --- | --- |
| .py files is written by programmers | .pyc files is not written by programmers |
| It is human readable and understandable | It can’t be understood by humans |
| It is called the source code | It is known as Byte code |
| It is changeable and can be edited | compiler version , no direct change can be done |
| It is the base code | It is the intermediate code |
| Uses simple English words | Consists of combination of letters and digits |
| Source requires compiler for translation | Byte code requires Interpreter for translation |
| Cannot be directly executed | Bytecode is converted to m/c code |
| It is written in Python programming languages | Compiled Bytecode files are generated by Python interpreter after passing .py files |
| Programmers write to create Python programs |  |
| These are plain text files | These are binary files |
| We can open them using a text editor to view and modify the code | We cannot directly read and understand their content using text editor |
| When we run a python program code the python Interpreter reads and executes the code directly from .py files | Python Interpreter doesnot directly execute the .py source code. Instead it compiles the code into a lower level representation known as Bytecode which is a platform independent |
|  | Bytecode is stored in .pyc to speed up execution of python programs. When we run .pyc files for future executions to avoid recompilation and unless .py file has been modified since the last compilation. |
| When we run a python program for the first time, interpreter compiles .py files to .pyc files. | .pyc files are compiled bytecode files used by Python Interpreter to speed up the program execution. |
| .pyc files are edited by programmers | .pyc files are automatically generated and used by Python to otimize the execution performance. |

Ans7. How compilation will happen internally. Explain in detail.

Firstly the code is written which is checked for errors. If there is any error, execution is halted and it throws error.If there is no error, then the source code is converted internally into bytecode(internal compilation.This bytecode cannot be directly executed by CPU. so, Python virtual machine comes into picture and is the interpreter and converts bytecode into machine code

1)Source code - firstly we write the code which is human readable and .py as extension.if there is error, it halts execution. If thee is no error it is converted to .py file

2)Tokenizer - It converts large piece of code into smaller chunks and creates list of tokens

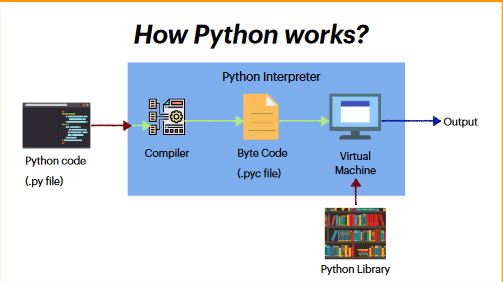
3)Parser - A Python Parser is a program that takes input data and converts it into machine-readable format.

4)Compilation - compiler converts source code(.py) files into byte code(.pyc)

5)Optimization - it is used to otimize the code like use less memory, few resources (i.e. CPU, Memory),extra spaces removed so that machine can work faster.

6)Python Virtual Machine - This is the Interpreter of Python. The .pyc code cannot be executed by CPU directly. So, Python virtual machine comes into picture that is the interpreter and converts bytecode into machine code.

7)After these steps, it is ready for execution.



Ans Why Python is Dynamically typed programming Language. Explain

Dynamically typed language: These are the languages that do not require to data type for any variable and it is interpreted at runtime by the machine itself. In these languages, interpreters assign the data type to a variable at runtime depending on its value.

Dynamically typed means there is no need to type data types before assigning values to variables.

Data type is needed at run time only. In Python while coding, we need not define data types, so it is dynamically typed programming language.

a = 10

print(typeof(a)) <class - 'int'>

Ans 9. Python is Platform independent. Explain

**Platform Independent definition**- It means same code can be used to run on any platform or OS. The same code can run practically on any platform. But With Python, we need to minimize compatibility issues, case sensitivity and avoiding specific modules.

Python is platform independent and can run on any OS. If there is code written for Windows, It will run on Mac or Linux without requiring the code to change. It is said to be portable because we do not need to change the code even we want to run the same piece of code on different machines with different OS. No changes has to be made in the code and we can directly execute that same code on other machine with any other OS like LINUX, MAC even if it has been written in machine with other OS like Windows.

The code can run on variety of OS.

Variety of platforms like – LINUX, SOLARIS, WINDOWS, MACINTOS

Ans 10. Different ways to write python program.

Interactive, IDLE, CommandPrompt, IDE

Advantages, Disadvantages

Interactive - When we type the code in one line and in another line we get the output.

Adv -

It is easier to code in interactive way

Faster – In one line we input and another line we get the output

Good for beginners – Those who are new to coding find it easy to work in interactive mode

Disadv –

not good for writing long code or larger programs. It is for one liner programs only

IDLE - It is scripting way of programming. It is integrated developement and learning environment.

Adv -

a)good for beginners - It is easier to learn and understand for them

b)Indentation support is provided in this Environment which makes our work a lot easier as Identation is very important in Python

c)Debugging is soon - Syntax errors are identified and rectified soon and easily

d)easier to code

Disadv

a)Debuging can be upgraded

b)Not good for advance level programs because it misses tools for that. It lacks various tools for complex programs

c)Does not have data science packages.

command Prompt - when we execute python code in cmd

adv -

a) good for beginners

b)faster – It is faster and higher performance is what we get while coding in command line interface

c)accuracy – It provides the most accurate results while performing arithmetic operations on variables.

disadv -

a)not good for complex programs - This involves many commands which are difficult to remember

b)need to remember commands

c)There is lot of copying and pasting in programming in cmd

IDE - Integrated Development Environment. It is a software application that has various tools for auto completion of code, building, editing, modifying the code

Adv -

a)good for advance coding – It provides tools for auto completion of code,

b)has various in-built packages and libraries

c)good for large programs – We can write programs, save them and t

d)advanced debugging is available is IDE environment like Visual Studio Code, Atom, Jupyter Notebook.

Dis adv -

a)not good for beginners

b)complex coding - when coding gets complex, it is troubling to code

Ans11.sourcecode vs bytecode

source is the base code written by programmars and bytecode is intermediate between source and machine code.

source code is the code which is written by us and is in human readable format

Source code uses simple English words, byte code is compiled version of source code

source code can be readable by humans,bytecode cannot be read by humans

source code is changeable; but byte code cannot be changed directly

source code has .py extension whereas bytecode has .pyc extension

|  |  |
| --- | --- |
| Source code is written by programmers | Byte code is not written by programmers |
| It is human readable and understandable | It can’t be understood by humans |
| It has .py as its code extension | It has .pyc as its code extension |
| It is changeable and can be edited | compiler version , no direct change can be done |
| Source code is the base code | Byte code is the intermediate code |
| Uses simple English words | Consists of combination of letters and digits |
| Source requires compiler for translation | Byte code requires Interpreter for translation |
| Cannot be directly executed | Bytecode is converted to m/c code |

Ans12.Register instruction set

These are commands thar are used to set register for a specific value. If we want to copy data from memory to register or vice-versa or need to perform arithmetic and logic operations, these register instruction commands are needed

Registers are type of computer memory which are used to quickly accept, store and transfer data and instructions that are being used immediately by CPU. Registers are also called Processors Registers. They may hold instruction, and used for storage of address and data. Computers need these registers to manipulate data and registers for holding memory address. Registers hold memory location and is used to calculate address of next instruction after execution of current instruction is completed.

Memory Unit has 4096 words and each word contains 6 bits. There are various registers explained below:

1) Data Register – It holds memory operand

2) Memory address register (MAR) – It holds address for memory location for main memory from where information is to be fetched or information is to be stored.

3)Program Counter(PC) – It contains address of instruction to be executed next.

4)Accumulator Register(AR)- It is also called Processor Register. It holds intermediate results for computations. It is used for arithmetic and logical operations.

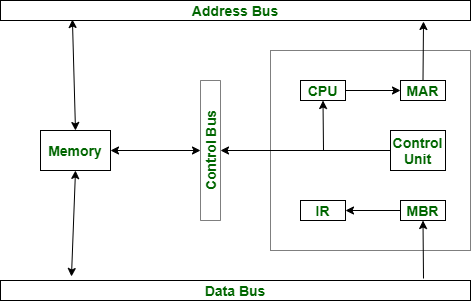
5)Instruction Register(IR) – Holds instruction code

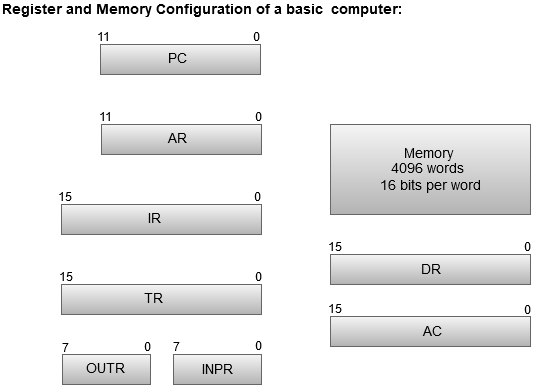
6)Temporary Register(TR) – Holds temporary data.

7)Input Register(IR) – It carries input data given by user

8)Output register(OR) – It contains output Data after processing.

9)Memory Buffer/Data Register(MBR ) – It consists of word of data to be written to memory or words mostly read.





Ans 13.High Level vs Low level programming Language

High level languages

a)programmar friendly - can be understood and used easily by programmars

b)portable - platform independent that is it can run on any OS

c)Easy to understand, learn and use

d)Debugging is easy - for such languages there are various editors which makes it easy to identify aand rectify errors

e)Needs compiler or translator for translation - It cannot be understood by machine so it has to be converted to machine readable form

f)Human readable - It is written by and for humans, so easily understood

g)widely used by programmars

h) Examples - C, C++, JAVA,PYTHON

Low level Languages

a)Machine friendly - It is understood by machines only

b)Machine dependent - It changes for different machines like MAC, LINUX, Windows will have different versions of it.

c)complex to understand, learn and not easy.

d)not human readable

e)Not widely used by programmers these days as simpler programming languages are there where we can code easily

g)debugging is complex - To understand and identify error and fixing them is difficult for low level languages.

h)needs assembler for translation -

i)Example - Machine readable code in form of 0's and 1's.

|  |  |
| --- | --- |
| Machine languages are understood by machines only | High Level languages can be understood by humans |
| These are machine dependent | These platform independent |
| Complex to understand, learn and use | Easy to learn, use and understand |
| Not readable by humans | Is easily read and can be changed |
| Debugging is difficult in low level languages | It is easier to debug, identify and rectify errors |
| Needs assembler for translation | Needs either compiler or Interpreter |
| Not widely used by programmers | Widely used by coders these days |
| Examples – Language consists of 0’s and 1’s | Examples – C,C++,JAVA, PYTHON |
| Direct Memory management | Interpreted |
| Much Faster performance | Poor performance |
| Codes are not simpler and not readable | Codes are concise |
| Hard to write and read by humans | Flexible syntax and easy to read |
| Few support and hard to learn | Object oriented and functional |
| Few support | Large community support |
| Prior Hardware knowledge required | Hardware knowledge is not required |
| Fast execution | Slow execution |
| Very little or no abstraction | Allow more abstraction |
| Support | Many facilities not provided at h/w level |
| Less memory required | More memory requirement |

Ans 14.Python architecture

1)Source code - firstly we write the code which is human readable and .py as extension.if there is error, it halts execution. If thee is no error it is converted to .py file

2)Tokenizer - It converts large piece of code into smaller chunks and creates list of tokens

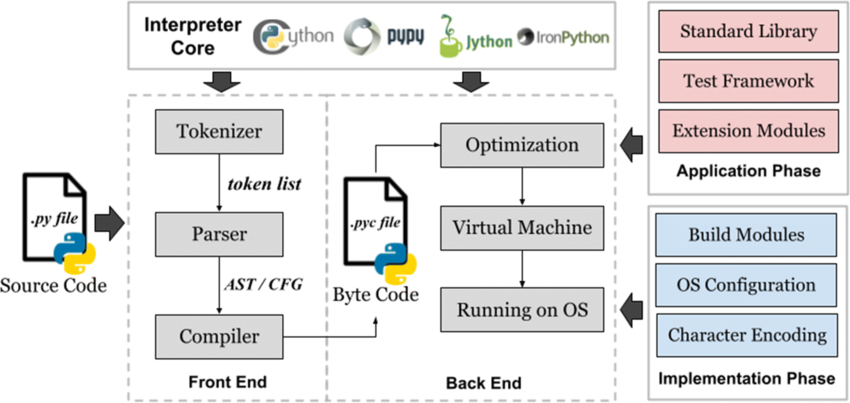
3)Parser - A Python Parser is a program that takes input data and converts it into machine-readable format.

4)Compilation - compiler converts source code(.py) files into byte code(.pyc)

5)Optimization - it is used to otimize the code like use less memory, few resources (i.e. CPU, Memory),extra spaces removed so that machine can work faster.

6)Python Virtual Machine - This is the Interpreter of Python. The .pyc code cannot be executed by CPU directly. So, Python virtual machine comes into picture that is the interpreter and converts bytecode into machine code.

7)After these steps, it is ready for execution.



The Great Python Architecture

1Ans 15.Explain Garbage Collection mechanism in detail.

**Garbgage collection** **Definition** –It is a process that attempts to reclaim garbage or memory occupied by objects that are no longer in use by programming. It was invented by McCarthy in 1959. And used in Lisp, Smalltalk, Python, Java. It involves algorithms like –

1)Reference Counting

2)Mark and sweep

3)Mark and compact

4)Copying Collector

5)Generational garbage collection

**Garbage collection is done automatically in Python**. It keeps track of allocated, deallocated memory. It keeps collection of unused objects. There are 2 ways of garbage collection in Python -

a)Reference Counter - When a variable is named or alias are used to refer an object, then its reference counter increases. When it's refernces are deleted, then it reference counter becomes zero and it is then garbae collected. Before that destructor del\_\_() method is invoked and all reference to object is deleted and object is garbage collected

b)Generational garbage collection - when an object points to itself, its reference counter never becomes zero. then generational garbag collector is used. GC is done in 3 phases:

a)Scan

b)Mark

c)sweep

In GC, It scans entire heap to mark reachable objects then sweep away unreachable objects.

