

PYTHON BASICS

* General Information :-

Whitespace matters! Indent where needed.

Import modules "import modulename"

This is a comment.

print("Hello world!") # Prints to screen.

* Conditional Statements :-

if issunny:

 Print('It's sunny!')

elif go <= temp < 100 and bath > 80 :

 print('Bath is hot and full!')

elif not ((job == 'qa' or user == 'adm')):

 print('Match if not qa or adm')

else:

 print('No match. job is ' + job)

ATUL KUMAR (LINKEDIN).

NOTES GALLERY (TELEGRAM).

* Strings :-

title = 'Us and them'

most list operations

title[0]

len(title)

title.split(' ')

':'.join(['A','B','C'])

nine = str(9)

title.replace('them', 'us')

works on strings

output: 'U'

11

#[‘us’,‘and’,‘them’]

#[‘A’:‘B’:‘C’]

convert int into string

us and us.

* Dictionaries :-

```
votes = {'red': 3, 'blue': 5}.  
votes.keys() # Output: ['blue', 'red']  
votes['gold'] = 4 # add a key / val  
del votes['gold'] # deletes key  
votes['blue'] = 6 # change value  
len(votes) # 2  
votes.values() # [6, 3]  
'green' in votes # False  
votes.has_key('red') # True
```

* Numbers :-

```
total = 3 * 3 # Output: 9  
total = 5 + 2 * 3 # Output: 11  
cost = 1.50 + 3.75 # Output: 5.25  
total = int("9") + 1 # Output: 10
```

ATUL KUMAR (LINKEDIN).

NOTES GALLERY (TELEGRAM).

* Tuples :-

Like lists, except they cannot be changed
tuple1 = (1, 2, 3, "a", "z") # creates tuple
tuple1[3] # 'a'

* Lists :-

```
scores = ['A', 'C', 90, 75, 'C']  
scores[0] # Output: 'A'  
scores[1:3] # 'C', 90  
scores.count('C') # 2  
scores.append(100) # Adds 100 to list  
scores.pop() # removes third item.
```

ATUL KUMAR (LINKEDIN).

Python OOPS Concepts

Object Oriented Programming.

Python is a multiparadigm programming language. It supports different programming approaches.

One of the most popular approaches to solve programming problem by creating objects. This is known as Object Oriented programming. (OOP).

OOP has two characteristics

- 1). Attributes
- 2). Behavior

Example:- A parrot is an object, as it has following properties.

- name, age, color as attributes.
- Singing, dancing as behavior.

The concepts of OOPS in Python focuses on creating reusable code. This concept is also known as DRY (Don't Repeat Yourself).

ATUL KUMAR (LINKEDIN)
TELEGRAM - NOTES GALLERY.

Class.

A class is a blueprint for the object. We can think of class as a sketch of parrot with labels. It contains all details about the name, colors, size etc.

Ex.

Class Parrot:

Pass.

Here class keyword define an empty class parrot from class we construct instances (-) an instance is a specific object created from a particular class.

ATUL KUMAR (LINKEDIN).

TELEGRAM-NOTES GALLERY

Object :-

An object (instance) is an instantiation of a class. When class is defined only the description for the class object is defined. Therefore, no memory or storage is allocated.

ex:-

```
Obj = Parrot()
```

Here object is an object of class parrot.

Suppose we have details of parrots. Now we are going to show how to build the class and object of parrots. We can access the class attribute using - class - species .

Inheritance :-

Inheritance is a way of creating new class for using details of an existing class without modifying it. The newly formed class is derived class, similarly, the existing class is a base class.

Ex:- Use of Inheritance in python.

```
Class Bird:
```

```
    def __init__(self):  
        print("Bird is ready")  
    def swim(self):  
        print("swim faster")
```

```
Class Penguin(Bird):
```

```
    def __init__(self):  
        super().__init__()  
    def run(self):
```

```
        print("Run fast")  
P = Penguin()  
P.swim()  
P.run()
```

ATUL KUMAR (LINKEDIN).

Output

Swim faster
run fast.

We can use the `super()` function inside the `_init_()` method. This allows us to run `_init_()` method.

Encapsulation :-

Using OOP in python, we can restrict access to methods and variables. This prevents data from direct modification which is called encapsulation. In python we denote private attribute using `_` as the prefix. i.e single `_` or double `__`.

Class Computer:

```

def __init__(self):
    self.__maxprice = 900
def setmaxprice(self, price):
    self.__maxprice = price.
C = computer()
C.__maxprice = 1000
C.setmaxprice(100)

```

We used `_init_()` method to store the maximum selling price of computer.

`C.__maxprice = 1000`

Method	Variable.
--------	-----------

Class Member access specifier	Access from own Class	Accessible from derived	Accessible from Object.
Private	Yes	No	No
Protected	Yes	Yes	No
Public	Yes	Yes	No

Polymorphism :-

Polymorphism is an ability to use a common interface for multiple forms (data types) Polymorphism in python defines methods in the child class that have the same name as the method in the parent class. It is possible to modify a method in a child class that it has inherited from child class .. Parent class. .

Class parrot :

```
def fly(self):
    print("parrot can fly")
def swim(self)
    print("parrot can't swim")
```

Class Penguin :

```
def fly(self):
    print("Penguin can't fly")
def swim(self):
    print("Penguin can swim")
```

```
def flying-test(bird):
    bird.fly()
```

```
blue = Parrot()
```

```
Peg = Penguin()
```

```
flying-test(blue)
```

```
flying-test(Peg).
```

Output

Parrot can fly.

Penguin can't fly.

ATULKUMAR (LINKEDIN).

TELEGRAM - NOTES GALLERY.

PYTHON FUNCTIONS

What are Functions?

A Function is a block of code only runs when is called.
In Python a function is defined using the def keyword.

```
def my_function():
    print("Hello")
```

Arguments in a Function

Arguments are specified after the function name,
inside the parentheses.

You can add as many arguments as you want
separate them with a comma.

```
def my_function(fname)
    print(fname + "Refsnes")
```

```
my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

ATUL KUMAR (LINKEDIN)
TELEGRAM - NOTES GALLERY.

Arbitrary Arguments

If you do not know how many arguments that will be
passed into your function, add a * before the parameter name
in the function definition.

```
def my_function(*kids):
    print("The youngest child is " + kids[2])
my_function("Soul", "god1", "txpark")
```

Keyword Arguments (Kwargs)

You can also send arguments with the key = value syntax.

```
def my_function(child3, child2, child1):  
    print("The youngest child is " + child3)  
my_function(child1 = "mcb", child2 = "bca", child3 = "btc")
```

ATUL KUMAR (LINKEDIN).

Arbitrary Keyword Arguments **kwargs.

If you do not know how many keyword arguments that will be passed into your function, add two asterisks: ** before the parameter name in the function definition.

```
def my_function(**kid):  
    print("His last name is " + kid["lname"])  
my_function(fname = "Tobias", lname = "Refsnes")
```

ATUL KUMAR (LINKEDIN).

PYTHON OPERATOR CHEATSHEET

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Arithmetic operators:

An arithmetic operator performs mathematical operations like addition, subtraction, division and multiplication.

Operator	Meaning	Example
+	Add two operands	>>> 5 + 4 9
-	Subtract two operands	>>> 6 - 4 2
*	Multiply two operands	>>> 2 * 10 20
/	divide the left operand by right operand	>>> 5 / 2 2.5
%	which will give us remainder part	>>> 5 % 3 2
//	which will give integer part	>>> 5 // 3 1
**	raised to the power	>>> 5 ** 3 125

Comparisons Operator

Operator	Description	Example
$==$	If two operands values are equal, then the condition becomes true.	$>>> 5 == 3$ False
$!=$	values of two operands are not equal, then condition become true.	$>>> 5 != 3$ True
$>$	IF the value of left operand is greater than the value of right operand, then condition becomes true.	$>>> 4 > 3$ True
$<$	If the value of left operand is less than the value of right operand, then condition becomes true.	$>>> 6 < 4$ False
\geq	greater than equal: If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	$>>> 5 \geq 6$ False
\leq	less than e	$>>> 4 \leq 6$ True

ATULKUMAR(LINKEDIN).
NOTES GALLERY(TELEGRAM)

Logical Operators

- Logical operators is Boolean expressions such as and , or, not.
- It is just a conditional test that a result is either true or false.

Continue →



Operator	Description	Example
And	if two operands are true it become true.	>>> True and True True >>> False and True False
Or	if two operands are non-zero then condition becomes true.	>>> True or false True >>> False or False False
Not	It is used to reverse the logical state of its operand.	>>> not True False >>> not not not True False

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Assignment operator:

Operator	Description	Example
=	Assigns values from right side operands to left side operand.	>>> x = 20
+=	It adds right operand to the left operand and assign the result to left operand.	>>> x += 20
-=	It subtracts right operand from the left operand and assign the result to left operand.	>>> x -= 20
*=	It multiplies right operand with the left operand and assign the result to left operand.	>>> x *= 50
/=	It divides left operand with the right operand and assign the result to left operand.	>>> x /= 50
%=	It takes modulus using two operands and assign the result to left operand.	>>> x %= 50
**=	Performs exponential (power) calculation on operators and assign value to the left operand.	>>> x **= 50
//=	It performs floor division on operators and assign value to the left operand.	>>> x //= 50

Membership Operator

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	>>> x = "Python" >>> 'P' in x True.
not in	Evaluates to true if it does not find a variable in the specified sequence and false otherwise.	>>> x = "python" >>> 'z' in x False.

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Bitwise Operators.

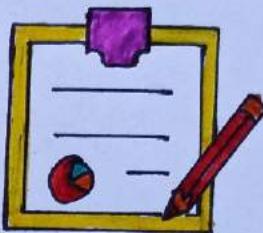
Operator	Description	Example
&	Operator copies a bit to the result if it exists in both operands	>>> 10 & 20 0
	It copies a bit if it exists in either operand.	>>> 10 20 30
^	It copies the bit if it is set in one operand but not both.	>>> 10 ^ 20 30
~	It is unary and has the effect of 'flipping' bits.	>>> ~ 20 -21
<<	The left operand's value is moved left by the number of bits specified by the right operand.	>>> 10 << 2 40
>>	The left operand's value is moved right by the number of bits specified by the right operand.	>>> 10 >> 2 2

Identity Operators

Operator	Description	Example.
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	>>> x = "hello" >>> y = "hello" >>> x is y True.
is not	Evaluates to false if the variables on either side of the operator point to the same object and True otherwise.	>>> x = "hello" >>> y = "hello" >>> x is not y False.

PYTHON NOTES

VARIABLES & DATATYPE



Python Variables :

Variable is a name that is used to refer to memory location. Python is also known as an identifier and used to hold value.

In python, we don't need to specify the type of variable. Python is a infer and smart enough to get variable type.

example : `x = 10`

`name = "Atul Kumar"`

`value = 12.2`

ATULKUMAR (LINKEDIN).

NOTES GALLERY (TELEGRAM).

identifier:

Variable are the example of identifier. An identifier is used to identify the literals used in the program.

Rules to name an identifier —

- The first character of the variable must be alphabet or underscore (_).
- All character except first character may be an alphabet (A-Z), (a-z) or digit (0-9).

Continue →

- Identifier name must not contain any white space or special character (!, @, #, %)
- Identifier name not similar to any keyword defined in language.

Valid vs invalid identifier

Valid

- abc123
- abc_de
- _abc
- ABC
- abc

Invalid

- 123 abc
- abc@
- 123
- For

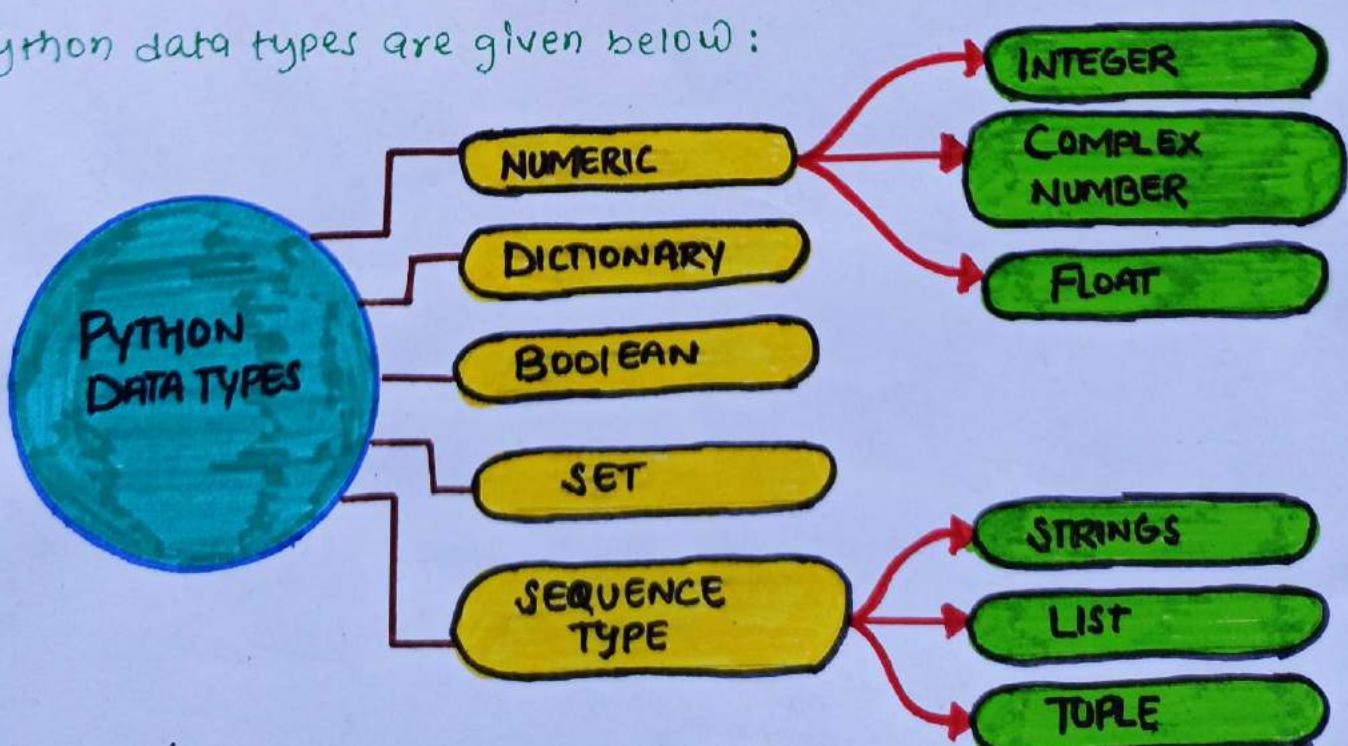
ATUL KUMAR (LINKEDIN).
NOTE GALLERY (TELEGRAM).

Python Datatype

A variable can hold different type of value. for example name is stored in string where as id is stored in integer.

Variable can store data of different types, and different types can do different thing.

Python data types are given below:



Operators in Python

The operator can be defined as a symbol which is responsible for particular operation between two operand.

Operator are used to perform operations on variable and value.

Python divides operator in following Groups:

- Arithmetic operator
- Comparison operator
- Identity operator
- Membership operator.
- Assignment operator.
- Logical operator
- Bitwise Operator

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Arithmetic Operator

Arithmetic Operators are used to perform common mathematical operation.

operator	name	example
+	addition	$x+y$.
-	Subtraction	$x-y$
*	Multiplication	$x*y$
/	Division	x/y
%	Modul	$x \% y$
**	exponentiation	$x^{**}y$
//	Floor division	$x//y$

Assignment Operator

Assignment operator are used to assign value to variable.

Operator	example	same as
$=$	$x = 5$	$x = 5$
$+=$	$x += 1$	$x = x + 1$
$-=$	$x -= 1$	$x = x - 1$
$*=$	$x *= 1$	$x = x * 1$
$/=$	$x /= 1$	$x = x / 1$
$\cdot=$	$x \cdot= 1$	$x = x \cdot 1$
$//=$	$x //= 1$	$x = x // 1$
$**=$	$x **= 1$	$x = x ** 1$
$\&=$	$x \&= 1$	$x = x \& 1$
$ =$	$x = 1$	$x = x 1$

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Comparison Operator

Operator	Description	example
$==$	equal	$x == y$
$!=$	not equal	$x != y$
$>$	greater than	$x > y$
$<$	less than	$x < y$
\geq	Greater/ equal	$x \geq y$
\leq	less/ equal	$x \leq y$

Logical Operator

Operator	Description	example
and	Returns true if both statement are true	$x < 5 \text{ and } x < 10$
or	Returns true if one of statement is true	$x < 5 \text{ or } x < 4$
not	Reverse the result, Returns false if the result is true.	$\text{not}(x > 5 \text{ and } x > 10)$

Python identity operator

Identity Operator are used to compare the object, not if they are equal. but if they are actually same object with the same memory location.

operator	Description	example
is	Return True if both variables are the same object.	x is y
is not	Returns true if both are not same object.	x is not y.

Python membership Operator.

Membership operator are used to test if a sequence is pretended is an object.

operator	Description	example
in	Return True if a sequence with the specified value is present.	x in y
not in	Return True if a sequence with the specified value is not present.	x not in y.

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Python Bitwise Operator

Bitwise operator perform bit by bit operation on the values of two operand. Consider the following table.

Continue →

→

Operator	name	Description
&	AND	Set each bit to 1 if both sides are 1.
	OR	Set each bit to 1 if one of two bits is 1.
^	XOR	Set each bit to 1 if only one of two bits is 1.
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros from the right and let the leftmost bits fall off.
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left and let the rightmost bits fall off.

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

Input in Python

python input() function

Python input function is used to get input from user. It prompts for the user input and read a line. After reading data. it converts it into a string and returns that. It throws an error E of error is EOF is read.

Parameter

Prompt : It is a string message with prompt for user input.

Return

It returns user input after converting into string.

Let see an example of input() function to understand its functionality.

Continue →

→ example:

Input

```
# Python input() function.  
val = input("enter a value :")  
print ("you entered : ", val)
```

Output

```
enter a value : 20  
you entered : 20
```

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

PYTHON INTERVIEW QUES.

1. Does Python have OOP's concept?

- Python is an Object Oriented Programming language. This means that any problem can be solved in python by creating a model. However, python can be treated as procedural as well as structural language.

2. How to add values to Python array?

- Elements can be added to an array using the append(), extend() and insert(i, x) function.

- append()
- extend()
- insert()

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

3. How do you do data abstraction in Python?

- Data abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in python by using interface and abstract class.

4. How will you capitalize the first letter of string?

- In python, the capitalize() method capitalizes the first letter of string. If the string already consists of a capital letter at the beginning, then it returns the original string.

5. What are generators in Python?

- Functions that return an iterable set of items are called generators.

6. What are docstrings in Python?

- Docstrings are not actually comments, but they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times serve the purpose of comments as well.

example:-

Using docstring as a comment

```
x = 8  
y = 4  
z = x/y  
print(z)
```

7. What are built-in type of Python?

- • Integers
• Floating-point
• Complex numbers
• String
• Boolean
• Built-in function.

ATUL KUMAR (LINKEDIN).
NOTES-GALLERY (TELEGRAM).

8. Define encapsulation in Python?

- Encapsulation means binding the code and data together. A Python class is an example of encapsulation.

9. What does an object() do?

- It return a featureless object that has a base class for all classes. Also it does not take any parameter.

10. How to create empty class in Python?

- The empty class is a class that does not have any code defined within its block. It can be created using the pass keyword. However, you can create object of this class outside the class itself. In python the pass command does nothing when its executed its a null statement.

example:-

```
class a:  
    pass  
Obj = a()  
Objname = "xyz"  
print ("Name = Object.name")
```

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

11. Does Python support multiple Inheritance?

- Multiple inheritance means that a class can be derived from more than one parent classes. Python does not support multiple inheritance unlike java.

12. What is Polymorphism in Python?

- Polymorphism means the ability to take multiple forms.. so far instance. If the parent class has a method named ABC then the child class also can have method with same name ABC having its own parameter and variable python allows polymorphism.

13. What is split used for?

- The split() method is used to separate a given string in Python.

example:-

```
a = "edureka python"  
print (a.split())
```

14. How to import modules in Python?

- Modules can be imported using the `import` keyword.
You can import modules in three ways.

example :-

1* `import array`
2* `import array as arr`
3* `from array import *`.

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

15. What is Pickling and Unpickling?

- `Pickle` module accepts any Python object and convert it into string representation and dump it using `dump` function, this process is called pickling.
While the process of retrieving original python objects from the stored string representation is called unpickling.

ATUL KUMAR (LINKEDIN).
NOTES GALLERY (TELEGRAM).

PYTHON INTERVIEW QUESTIONS PART-2

1]. What is pickling and unpickling?

- Pickle module accepts any python object and convert it into string representation and dump it using dump function, this process is called pickling. While the process of retrieving original python objects from the stored string representation is called Unpickling.

ATUL KUMAR (LINKEDIN)

TELEGRAM-NOTES GALLERY.

2]. What are generators in python?

- Functions that return an iterable set of items are called generators.

3]. How you will capitalize the first letter of string?

- In python, the capitalize() method capitalizes the first letter of string. If the string already consists of a capital letter at the beginning, then, it returns the original string.

4]. What are docstring in python?

- Docstrings are not actually comments, but they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

example:-

```
"""
using docstring as a comment
"""
```

```
X=8
Y=4
Z=X/Y
Print(Z)
```

5]. What are built-in types of Python?

- > • Integers
- Floating - Point
- Complex number
- String
- Boolean
- Built in function .

6]. Define encapsulation in Python?

- > Encapsulation means binding the code and data together. A Python class is an example of encapsulation.

ATUL KUMAR (LINKEDIN).

TELEGRAM-NOTES GALLERY.

7]. What does an object() do?

- > It returns a featureless object that is a base class for all classes. Also it does not take any parameter.

8]. How to create empty class in python?

- > The empty class is a class that does not have any code defined within its block it can be created using the pass keyword. However, you can create object of this class outside the class itself. In python the pass command does nothing when its executed. its a null statement.

example :-

```
class a:  
    pass  
obj = a()  
objname = "XYZ"  
print("Name = " + obj.name)
```

9]. Does Python support multiple inheritance?

- > Multiple inheritance means that a class can be derived from more than one parent classes. Python does support multiple inheritance. unlike . java.

10]. What is Polymorphism in python?

- Polymorphism means the ability to take multiple forms so for instance. If the parent class has a method named ABC then the child class also can have method with same name ABC having its own parameter and variable Python allows polymorphism.

11]. How do you do data abstraction in Python?

- Data abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in python by using interface and abstract class.

12]. How to import modules in Python?

- Modules can be imported using the import keyword. You can import modules in three ways.

example:-

- ①. import array
- ②. import array as arr
- ③. from array import *

13]. What is split used for?

- The split() method is used to separate a given string in python.

example:-

```
a = "edureka Python"  
print(a.split())
```

14]. Does Python have OOPS concept?

➤ Python is an Object Oriented Programming language. This means that any program can be solved in python by creating a model. However, python can be treated as procedural as well as structural lang.

ATULKUMAR (LINKEDIN).

15]. How to add Values to python array?

TELEGRAM-NOTES GALLERY

➤ Elements can be added to an array using the `append()`, `extend()` and `insert(i, x)` function.

- ① `append()`
- ② `extend()`
- ③ `insert()`.

PYTHON INTERVIEW

QUESTIONS

Part-1

1]. What type of language is Python? Programming or scripting?

- Python is capable of scripting, but in general purpose programming language.

2]. How is Python an interpreted language?

- An interpreted language is any programming language which is not in machine level code before runtime. Therefore, Python is an interpreted language.

ATUL KUMAR (LINKEDIN)
TELEGRAM-NOTES GALLERY.

3]. What is PEP 8?

- A name PEP stands for python Enhancement proposal. It is a set of rules that specify how to format python code for maximum readability.

4]. How is memory managed in Python?

- • Memory management in python is managed by python private heap space. All python objects and data structures are located in private heap. The programmer does not have access to this private heap. The python interpreter takes care of instead.
- The allocation of heap space for python objects is done by python's memory manager. The core API gives access to some tools for the programmer to code.

- Python also has an inbuilt garbage collection which recycles all the unused memory and so for that it can be made available to the heap space.

5]. **What is namespace in Python?**

- A namespace is naming system used to make sure that names are unique to avoid naming conflicts.

ATUL KUMAR (LINKEDIN).

TELEGRAM-NOTES GALLERY

6]. **What are Python modules? Name some commonly used built-in modules in Python?**

- Python modules are files containing python code. This code can be functions classes or variables. A python module is a .Py file containing executable code.

Some of the commonly used built-in modules are:

- os
- sys
- datetime
- json

7]. **What are local and global variable in Python?**

➤ **Global Variable :-**

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:-

Any variable declared inside a function is known as a local variable. This variable is present in the local and not in the global space.

8]. **Is Python case sensitive?**

- Yes, Python is a case sensitive language.

9]. What is type conversion in Python?

- > Type conversion refers to the conversion of one data type into another.
For ex. `int()` , `float()` , `set()` , `list()`.

10]. Is Indentation required in Python?

- > Indentation is necessary for python. It specifies a block of code. All code within loops classes function etc within an indented blocks. It is usually done using four space character. If your code is not indented necessarily. It will not execute accurately and will throw errors as well.

11]. 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.

```
def Newfun();
    print("Iti ")
```

12]. What is `_init_` ?

- > `_init_` is a method or constructor in Python. This method is automatically called to allocate memory when a new object of a class is created. All classes have the `_init_` method. for eg:-

```
Class Emp:
    def __init__(self, name):
        self.name = name
E1 = Emp("xyz")
print(E1.name)
```

13]. What is lambda Function?

- > An anonymous function is known as a lambda function. This function can have any number of parameters but have just one statement.

example:-

```
a = lambda x, y: x + y
print(a(5, 6))
```

14]. What is self in Python?

- > Self is an instance or an object of a class. In python, this is explicitly included as the first parameter. This is not the case in java where it's optional. It helps to 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.

ATUL KUMAR (LINKEDIN).

TELEGRAM-NOTES GALLERY.

15]. What are Python Iterators?

- > Iterators are objects which can be traversed through or iterated upon.