Q-1: What Is Python, What Are The Benefits Of Using It, And What Do You Understand Of PEP 8?

Python is one of the most successful interpreted languages. When you write a Python script, it doesn't need to get compiled before execution. Few other interpreted languages are PHP and Javascript.

Benefits Of Python Programming

- Python is a dynamic-typed language. It means that you don't need to mention the data type of variables during their declaration. It allows to set variables like var1=101 and var2 =" You are an engineer." without any error.
- Python supports object orientated programming as you can define classes along with the composition and inheritance. It doesn't use access specifiers like public or private).
- Functions in Python are like first-class objects. It suggests you can assign them to variables, return from other methods and pass as arguments.
- Developing using Python is quick but running it is often slower than compiled languages. Luckily, Python enables to include the "C" language extensions so you can optimize your scripts.
- Python has several usages like web-based applications, test automation, data modeling, big data analytics and much more. Alternatively, you can utilize it as a "glue" layer to work with other languages.

PEP 8.

PEP 8 is the latest Python coding standard, a set of coding recommendations. It guides to deliver more readable Python code.

Q-2: What Is The Output Of The Following Python Code Fragment? Justify Your Answer.

```
def extendList(val, list=[]):
    list.append(val)
    return list

list1 = extendList(10)
list2 = extendList(123,[])
list3 = extendList('a')

print "list1 = %s" % list1
print "list2 = %s" % list2
print "list3 = %s" % list3
```

The result of the above Python code snippet is:

```
list1 = [10, 'a']
list2 = [123]
list3 = [10, 'a']
```

You may erroneously expect list1 to be equal to [10] and list3 to match with ['a'], thinking that the list argument will initialize to its default value of [] every time there is a call to the extendList.

However, the flow is like that a new list gets created once after the function is defined. And the same get used whenever someone calls the extendList method without a list argument. It works like this because the calculation of expressions (in default arguments) occurs at the time of function definition, not during its invocation.

The list1 and list3 are hence operating on the same default list, whereas list2 is running on a separate object that it has created on its own (by passing an empty list as the value of the list parameter).

The definition of the extendList function can get changed in the following manner.

```
def extendList(val, list=None):
 if list is None:
  list = []
 list.append(val)
 return list
```

With this revised implementation, the output would be:

```
list1 = [10]
list2 = [123]
list3 = ['a']
```

Q-3: What Is The Statement That Can Be Used In Python If The Program Requires No Action But Requires It Syntactically?

The pass statement is a null operation. Nothing happens when it executes. You should use "pass" keyword in lowercase. If you write "Pass," you'll face an error like "NameError: name Pass is not defined." Python statements are case sensitive.

```
letter = "hai sethuraman"
for i in letter:
   if i == "a":
      pass
      print("pass statement is execute .....")
   else:
      print(i)
```

Q-4: What's The Process To Get The Home Directory Using '~' In Python?

You need to import the os module, and then just a single line would do the rest.

```
import os
print (os.path.expanduser('~'))
```

Output:

```
/home/runner
```

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions

Q-5: What Are The Built-In Types Available In Python?

Here is the list of most commonly used built-in types that Python supports:

- **Immutable built-in datatypes of Python**
  - Numbers
  - Strings
  - Tuples
- **Mutable built-in datatypes of Python**
  - List
  - Dictionaries
  - Sets

Q-6: How To Find Bugs Or Perform Static Analysis In A Python Application?

- You can use PyChecker, which is a static analyzer. It identifies the bugs in Python project and also reveals the style and complexity related bugs.
- Another tool is Pylint, which checks whether the Python module satisfies the coding standard.

Q-7: When Is The Python Decorator Used?

Python decorator is a relative change that you do in Python syntax to adjust the functions quickly.

Q-8: What Is The Principal Difference Between A List And The Tuple?

List Vs. Tuple.

The principal difference between a list and the tuple is that the former is mutable while the tuple is not.

A tuple is allowed to be hashed, for example, using it as a key for dictionaries.

Q-9: How Does Python Handle Memory Management?

- Python uses private heaps to maintain its memory. So the heap holds all the Python objects and the data structures. This area is only accessible to the Python interpreter; programmers can't use it.
- And it's the Python memory manager that handles the Private heap. It does the required allocation of the memory for Python objects.
- Python employs a built-in garbage collector, which salvages all the unused memory and offloads it to the heap space.

Q-10: What Are The Principal Differences Between The Lambda And Def?

Lambda Vs. Def.

- Def can hold multiple expressions while lambda is a uni-expression function.
- Def generates a function and designates a name to call it later. Lambda forms a function object and returns it.
- Def can have a return statement. Lambda can't have return statements.
- Lambda supports to get used inside a list and dictionary.

Also Check.
Python Programming Quiz for Beginners

Q-11: Write A Reg Expression That Confirms An Email Id Using The Python Reg Expression Module "Re"?

Python has a regular expression module "re."

Check out the **"re"** expression that can check the email id for .com and .co.in subdomain.

```
import re
print(re.search(r"[0-9a-zA-Z.]+@[a-zA-Z]+\.(com|co\.in)$","micheal.pages@mp.com"))
```

Q-12: What Do You Think Is The Output Of The Following Code Fragment? Is There Any Error In The Code?

```
list = ['a', 'b', 'c', 'd', 'e']
print (list[10:])
```

The result of the above lines of code is []. There won't be any error like an IndexError.

You should know that trying to fetch a member from the list using an index that exceeds the member count (for example, attempting to access list[10] as given in the question) would yield an IndexError. By the way, retrieving only a slice at the starting index that surpasses the no. of items in the list won't result in an IndexError. It will just return an empty list.

Q-13: Is There A Switch Or Case Statement In Python? If Not Then What Is The Reason For The Same?

No, Python does not have a Switch statement, but you can write a Switch function and then use it.

Q-14: What Is A Built-In Function That Python Uses To Iterate Over A Number Sequence?

Range() generates a list of numbers, which is used to iterate over for loops.

```
for i in range(5):
print(i)
```

The range() function accompanies two sets of parameters.

- **range(stop)**
- stop: It is the no. of integers to generate and starts from zero. eg. range(3) == [0, 1, 2].
- **range([start], stop[, step])**
- Start: It is the starting no. of the sequence.
- Stop: It specifies the upper limit of the sequence.
- Step: It is the incrementing factor for generating the sequence.
- **Points to note:**
- Only integer arguments are allowed.
- Parameters can be positive or negative.
- The **range()** function in Python starts from the zeroth index.

Q-15: What Are The Optional Statements Possible Inside A Try-Except Block In Python?

There are two optional clauses you can use in the **try-except** block.

- The "**else**" clause
- It is useful if you want to run a piece of code when the try block doesn't create an exception.
- The **"finally"** clause
- It is useful when you want to execute some steps which run, irrespective of whether there occurs an exception or not.

Q-16: What Is A String In Python?

A string in Python is a sequence of alpha-numeric characters. They are immutable objects. It means that they don't allow modification once they get assigned a value. Python provides several methods, such as join(), replace(), or split() to alter strings. But none of these change the original object.

Q-17: What Is Slicing In Python?

Slicing is a string operation for extracting a part of the string, or some part of a list. In Python, a string (say text) begins at index 0, and the nth character stores at position text[n-1]. Python can also perform reverse indexing, i.e., in the backward direction, with the help of negative numbers. In Python, the slice() is also a constructor function which generates a slice object. The result is a set of indices mentioned by range(start, stop, step). The slice() method allows three parameters. 1. start – starting number for the slicing to begin. 2. stop – the number which indicates the end of slicing. 3. step – the value to increment after each index (default = 1).

Q-18: What Is %S In Python?

Python has support for formatting any value into a string. It may contain quite complex expressions.

One of the common usages is to push values into a string with the %s format specifier. The formatting operation in Python has the comparable syntax as the C function printf() has.

Q-19: Is A String Immutable Or Mutable In Python?

Python strings are indeed immutable.

Let's take an example. We have an "str" variable holding a string value. We can't mutate the container, i.e., the string, but can modify what it contains that means the value of the variable.

Q-20: What Is The Index In Python?

An index is an integer data type which denotes a position within an ordered list or a string.

In Python, strings are also lists of characters. We can access them using the index which begins from zero and goes to the length minus one.

For example, in the string "Program," the indexing happens like this:

```
Program 0 1 2 3 4 5
```

Q-21: What Is Docstring In Python?

A docstring is a unique text that happens to be the first statement in the following Python constructs:

Module, Function, Class, or Method definition.

A docstring gets added to the __doc__ attribute of the string object.

Now, read some of the Python interview questions on functions.

Q-22: What Is A Function In Python Programming?

A function is an object which represents a block of code and is a reusable entity. It brings modularity to a program and a higher degree of code reusability.

Python has given us many built-in functions such as print() and provides the ability to create user-defined functions.

Q-23: How Many Basic Types Of Functions Are Available In Python?

Python gives us two basic types of functions.

1. Built-in, and

2. User-defined.

The built-in functions happen to be part of the Python language. Some of these are print(), dir(), len(), and abs() etc.

Q-24: How Do We Write A Function In Python?

We can create a Python function in the following manner.

Step-1: to begin the function, start writing with the keyword def and then mention the function name.

Step-2: We can now pass the arguments and enclose them using the parentheses. A colon, in the end, marks the end of the function header.

Step-3: After pressing an enter, we can add the desired Python statements for execution.

Q-25: What Is A Function Call Or A Callable Object In Python?

A function in Python gets treated as a callable object. It can allow some arguments and also return a value or multiple values in the form of a tuple. Apart from the function, Python has other constructs, such as classes or the class instances which fits in the same category.

Q-26: What Is The Return Keyword Used For In Python?

The purpose of a function is to receive the inputs and return some output.

The return is a Python statement which we can use in a function for sending a value back to its caller.

Q-27: What Is “Call By Value” In Python?

In call-by-value, the argument whether an expression or a value gets bound to the respective variable in the function.

Python will treat that variable as local in the function-level scope. Any changes made to that variable will remain local and will not reflect outside the function.

Q-28: What Is “Call By Reference” In Python?

We use both “call-by-reference” and “pass-by-reference” interchangeably. When we pass an argument by reference, then it is available as an implicit reference to the function, rather than a simple copy. In such a case, any modification to the argument will also be visible to the caller.

This scheme also has the advantage of bringing more time and space efficiency because it leaves the need for creating local copies.

On the contrary, the disadvantage could be that a variable can get changed accidentally during a function call. Hence, the programmers need to handle in the code to avoid such uncertainty.

Q-29: What Is The Return Value Of The Trunc() Function?

The Python trunc() function performs a mathematical operation to remove the decimal values from a particular expression and provides an integer value as its output.

Q-30: Is It Mandatory For A Python Function To Return A Value?

It is not at all necessary for a function to return any value. However, if needed, we can use None as a return value.

Q-31: What Does The Continue Do In Python?

The continue is a jump statement in Python which moves the control to execute the next iteration in a loop leaving all the remaining instructions in the block unexecuted.

The continue statement is applicable for both the “while” and “for” loops.

Q-32: What Is The Purpose Of Id() Function In Python?

The id() is one of the built-in functions in Python.

```
Signature: id(object)
```

It accepts one parameter and returns a unique identifier associated with the input object.

Q-33: What Does The *Args Do In Python?

We use *args as a parameter in the function header. It gives us the ability to pass N (variable) number of arguments.

Please note that this type of argument syntax doesn't allow passing a named argument to the function.

Example of using the *args:

```
# Python code to demonstrate
# *args for dynamic arguments
def fn(*argList):
for argx in argList:
print (argx)

fn('I', 'am', 'Learning', 'Python')
```

The output:

```
I
am
Learning
Python
```

Q-34: What Does The **Kwargs Do In Python?

We can also use the **kwargs syntax in a Python function declaration. It let us pass N (variable) number of arguments which can be named or keyworded.

Example of using the **kwargs:

```
# Python code to demonstrate
# **kwargs for dynamic + named arguments
def fn(**kwargs):
for emp, age in kwargs.items():
print ("%s's age is %s." %(emp, age))

fn(John=25, Kalley=22, Tom=32)
```

The output:

```
John's age is 25.
```

```
Kalley's age is 22.
Tom's age is 32.
```

Q-35: Does Python Have A Main() Method?

The main() is the entry point function which happens to be called first in most programming languages.

Since Python is interpreter-based, so it sequentially executes the lines of the code one-by-one.

Python also does have a Main() method. But it gets executed whenever we run our Python script either by directly clicking it or starts it from the command line.

We can also override the Python default main() function using the Python if statement. Please see the below code.

```
print("Welcome")
print("__name__ contains: ", __name__)
def main():
print("Testing the main function")
if __name__ == '__main__':
main()
```

The output:

```
Welcome
__name__ contains:  __main__
Testing the main function
```

Q-36: What Does The __ Name __ Do In Python?

The __name__ is a unique variable. Since Python doesn't expose the main() function, so when its interpreter gets to run the script, it first executes the code which is at level 0 indentation.

To see whether the main() gets called, we can use the __name__ variable in an if clause compares with the value "__main__."

Q-37: What Is The Purpose Of "End" In Python?

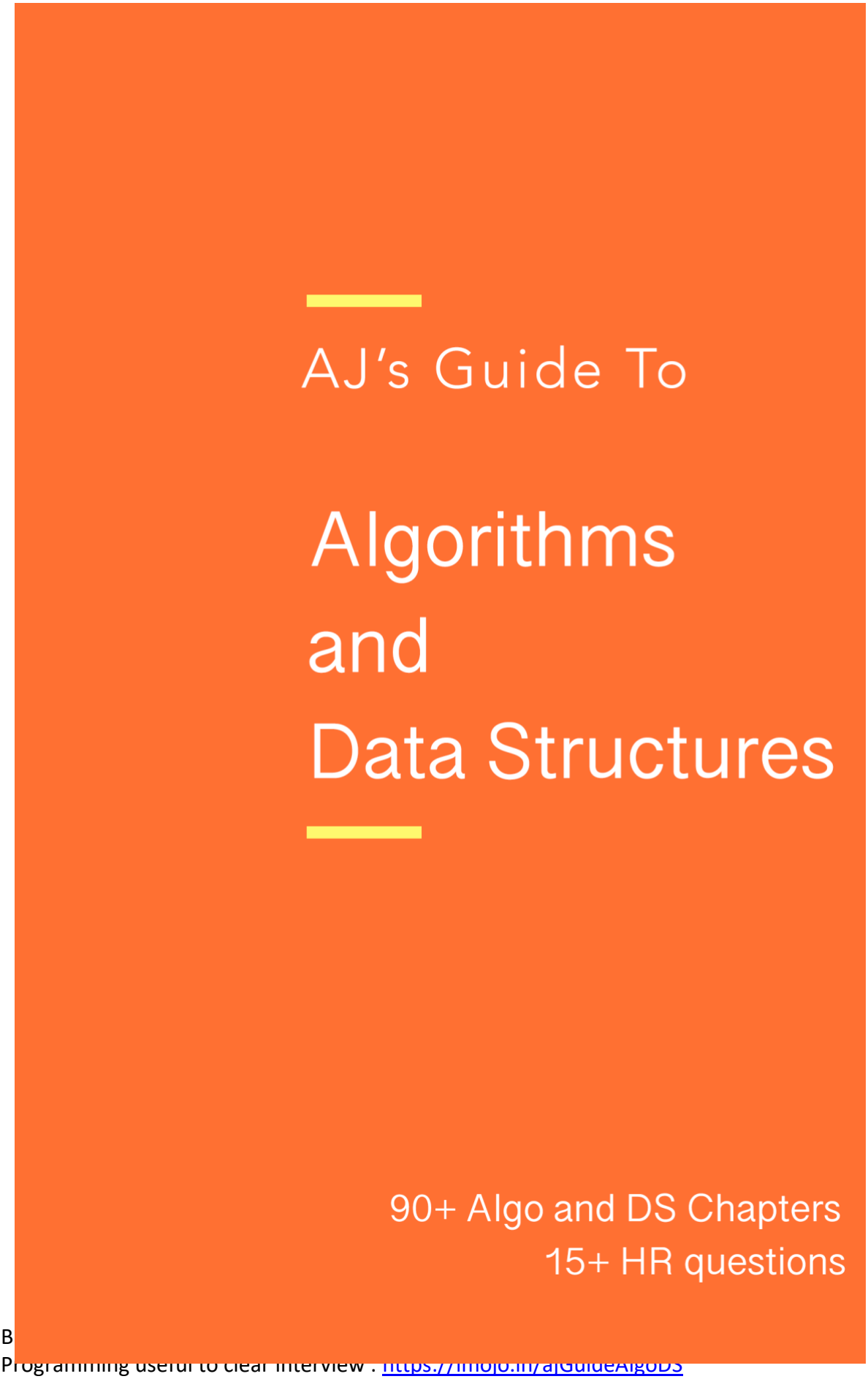
Python's print() function always prints a newline in the end. The print() function accepts an optional parameter known as the 'end.' Its value is '\n' by default. We can change the end character in a print statement with the value of our choice using this parameter.

```
# Example: Print a  instead of the new line in the end.
print("Let's learn" , end = ' ')
print("Python")

# Printing a dot in the end.
print("Learn to code from techbeamers" , end = '.')
print("com", end = ' ')
```

The output is:

```
Let's learn Python
Learn to code from techbeamers.com
```



Q-38: When Should You Use The “Break” In Python?

Python provides a break statement to exit from a loop. Whenever the break hits in the code, the control of the program immediately exits from the body of the loop.

The break statement in a nested loop causes the control to exit from the inner iterative block.

Q-39: What Is The Difference Between Pass And Continue In Python?

The continue statement makes the loop to resume from the next iteration.

On the contrary, the pass statement instructs to do nothing, and the remainder of the code executes as usual.

Q-40: What Does The Len() Function Do In Python?

In Python, the len() is a primary string function. It determines the length of an input string.

```
>>> some_string = 'techbeamers'
>>> len(some_string)
11
```

Q-41: What Does The Chr() Function Do In Python?

The chr() function got re-added in Python 3.2. In version 3.0, it got removed.

It returns the string denoting a character whose Unicode code point is an integer.

For example, the chr(122) returns the string ‘z’ whereas the chr(1212) returns the string ‘Ҽ’.

Q-42: What Does The Ord() Function Do In Python?

The ord(char) in Python takes a string of size one and returns an integer denoting the Unicode code format of the character in case of a Unicode type object, or the value of the byte if the argument is of 8-bit string type.

```
>>> ord("z")
122
```

Q-43: What Is Rstrip() In Python?

Python provides the rstrip() method which duplicates the string but leaves out the whitespace characters from the end.

The rstrip() escapes the characters from the right end based on the argument value, i.e., a string mentioning the group of characters to get excluded.

The signature of the rstrip() is:

```
str.rstrip([char sequence/pre>
#Example
test_str = 'Programming    '
# The trailing whitespaces are excluded
print(test_str.rstrip())
```

Q-44: What Is Whitespace In Python?

Whitespace represents the characters that we use for spacing and separation.

They possess an “empty” representation. In Python, it could be a tab or space.

Q-45: What Is Isalpha() In Python?

Python provides this built-in isalpha() function for the string handling purpose.

It returns True if all characters in the string are of alphabet type, else it returns False.

Q-46: How Do You Use The Split() Function In Python?

Python’s split() function works on strings to cut a large piece into smaller chunks, or sub-strings. We can specify a separator to start splitting, or it uses the space as one by default.

```
#Example
str = 'pdf csv json'
print(str.split(" "))
print(str.split())
```

The output:

```
['pdf', 'csv', 'json']
['pdf', 'csv', 'json']
```

Q-47: What Does The Join Method Do In Python?

Python provides the join() method which works on strings, lists, and tuples. It combines them and returns a united value.

Q-48: What Does The Title() Method Do In Python?

Python provides the title() method to convert the first letter in each word to capital format while the rest turns to Lowercase.

```
#Example
str = 'lEaRn pYtHoN'
print(str.title())
```

The output:

```
Learn Python
```

Now, check out some general purpose Python interview questions.

Q-49: What Makes The CPython Different From Python?

CPython has its core developed in C. The prefix 'C' represents this fact. It runs an interpreter loop used for translating the Python-ish code to C language.

Q-50: Which Package Is The Fastest Form Of Python?

PyPy provides maximum compatibility while utilizing CPython implementation for improving its performance.

The tests confirmed that PyPy is nearly five times faster than the CPython. It currently supports Python 2.7.

Q-51: What Is GIL In Python Language?

Python supports GIL (the global interpreter lock) which is a mutex used to secure access to Python objects, synchronizing multiple threads from running the Python bytecodes at the same time.

Q-52: How Is Python Thread Safe?

Python ensures safe access to threads. It uses the GIL mutex to set synchronization. If a thread loses the GIL lock at any time, then you have to make the code thread-safe.

For example, many of the Python operations execute as atomic such as calling the sort() method on a list.

Q-53: How Does Python Manage The Memory?

Python implements a heap manager internally which holds all of its objects and data structures.

This heap manager does the allocation/de-allocation of heap space for objects.

Q-54: What Is A Tuple In Python?

A tuple is a collection type data structure in Python which is immutable.

They are similar to sequences, just like the lists. However, There are some differences between a tuple and list; the former doesn't allow modifications whereas the list does.

Also, the tuples use parentheses for enclosing, but the lists have square brackets in their syntax.

Q-55: What Is A Dictionary In Python Programming?

A dictionary is a data structure known as an associative array in Python which stores a collection of objects.

The collection is a set of keys having a single associated value. We can call it a hash, a map, or a hashmap as it gets called in other programming languages.

Q-56: What Is The Set Object In Python?

Sets are unordered collection objects in Python. They store unique and immutable objects. Python has its implementation derived from mathematics.

Q-57: What Is The Use Of The Dictionary In Python?

A dictionary has a group of objects (the keys) map to another group of objects (the values). A Python dictionary represents a mapping of unique Keys to Values.

They are mutable and hence will not change. The values associated with the keys can be of any Python types.

Q-58: Is Python List A Linked List?

A Python list is a variable-length array which is different from C-style linked lists.

Internally, it has a contiguous array for referencing to other objects and stores a pointer to the array variable and its length in the list head structure.

Here are some Python interview questions on classes and objects.

Q-59: What Is Class In Python?

Python supports object-oriented programming and provides almost all OOP features to use in programs.

A Python class is a blueprint for creating the objects. It defines member variables and gets their behavior associated with them.

We can make it by using the keyword “class.” An object gets created from the constructor. This object represents the instance of the class.

In Python, we generate classes and instances in the following way.

```
>>>class Human:  # Create the class
...    pass
>>>man = Human()  # Create the instance
>>>print(man)
<__main__.Human object at 0x0000000003559E10>
```

Q-60: What Are Attributes And Methods In A Python Class?

A class is useless if it has not defined any functionality. We can do so by adding attributes. They work as containers for data and functions. We can add an attribute directly specifying inside the class body.

```
>>> class Human:
...    profession = "programmer" # specify the attribute 'profession' of the class
>>> man = Human()
>>> print(man.profession)
programmer
```

After we added the attributes, we can go on to define the functions. Generally, we call them methods. In the method signature, we always have to provide the first argument with a self-keyword.

```
>>> class Human:
profession = "programmer"
def set_profession(self, new_profession):
self.profession = new_profession
>>> man = Human()
>>> man.set_profession("Manager")
>>> print(man.profession)
Manager
```

Q-61: How To Assign Values For The Class Attributes At Runtime?

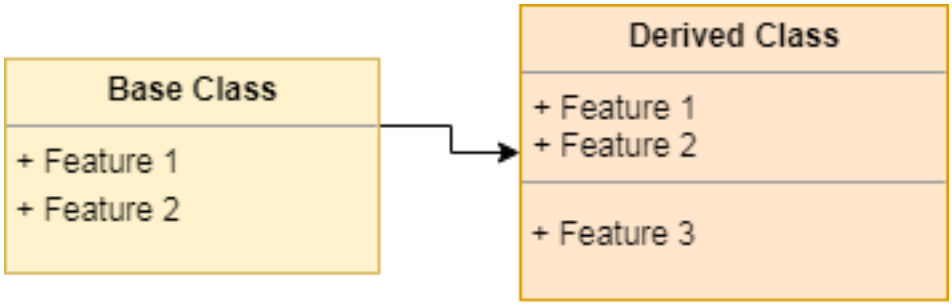
We can specify the values for the attributes at runtime. We need to add an init method and pass input to object constructor. See the following example demonstrating this.

```
>>> class Human:
    def __init__(self, profession):
        self.profession = profession
    def set_profession(self, new_profession):
        self.profession = new_profession

>>> man = Human("Manager")
>>> print(man.profession)
Manager
```

Q-62: What Is Inheritance In Python Programming?

Inheritance is an OOP mechanism which allows an object to access its parent class features. It carries forward the base class functionality to the child.



Inheritance In Python

We do it intentionally to abstract away the similar code in different classes.

The common code rests with the base class, and the child class objects can access it via inheritance. Check out the below example.

```
class PC: # Base class
    processor = "Xeon" # Common attribute
    def set_processor(self, new_processor):
        processor = new_processor

class Desktop(PC): # Derived class
```

```
os = "Mac OS High Sierra" # Personalized attribute
ram = "32 GB"

class Laptop(PC): # Derived class
os = "Windows 10 Pro 64" # Personalized attribute
ram = "16 GB"

desk = Desktop()
print(desk.processor, desk.os, desk.ram)

lap = Laptop()
print(lap.processor, lap.os, lap.ram)
```

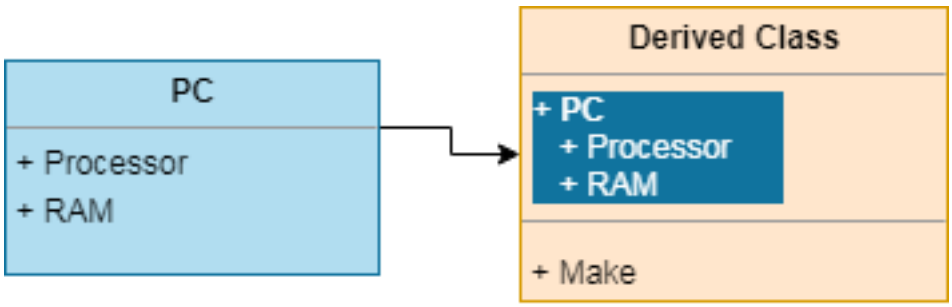
The output:

```
Xeon Mac OS High Sierra 32 GB
Xeon Windows 10 Pro 64 16 GB
```

Q-63: What Is Composition In Python?

The composition is also a type of inheritance in Python. It intends to inherit from the base class but a little differently, i.e., by using an instance variable of the base class acting as a member of the derived class.

See the below diagram.



Composition In Python

To demonstrate composition, we need to instantiate other objects in the class and then make use of those instances.

```
class PC: # Base class
processor = "Xeon" # Common attribute
def __init__(self, processor, ram):
self.processor = processor
```

```
        self.ram = ram

    def set_processor(self, new_processor):
        processor = new_processor

    def get_PC(self):
        return "%s cpu & %s ram" % (self.processor, self.ram)

class Tablet():
    make = "Intel"
    def __init__(self, processor, ram, make):
        self.PC = PC(processor, ram) # Composition
        self.make = make

    def get_Tablet(self):
        return "Tablet with %s CPU & %s ram by %s" % (self.PC.processor, self.PC.ram, self.make)

if __name__ == "__main__":
    tab = Tablet("i7", "16 GB", "Intel")
    print(tab.get_Tablet())
```

The output is:

```
Tablet with i7 CPU & 16 GB ram by Intel
```

Q-64: What Are Errors And Exceptions In Python Programs?

Errors are coding issues in a program which may cause it to exit abnormally.

On the contrary, exceptions happen due to the occurrence of an external event which interrupts the normal flow of the program.

Q-65: How Do You Handle Exceptions With Try/Except/Finally In Python?

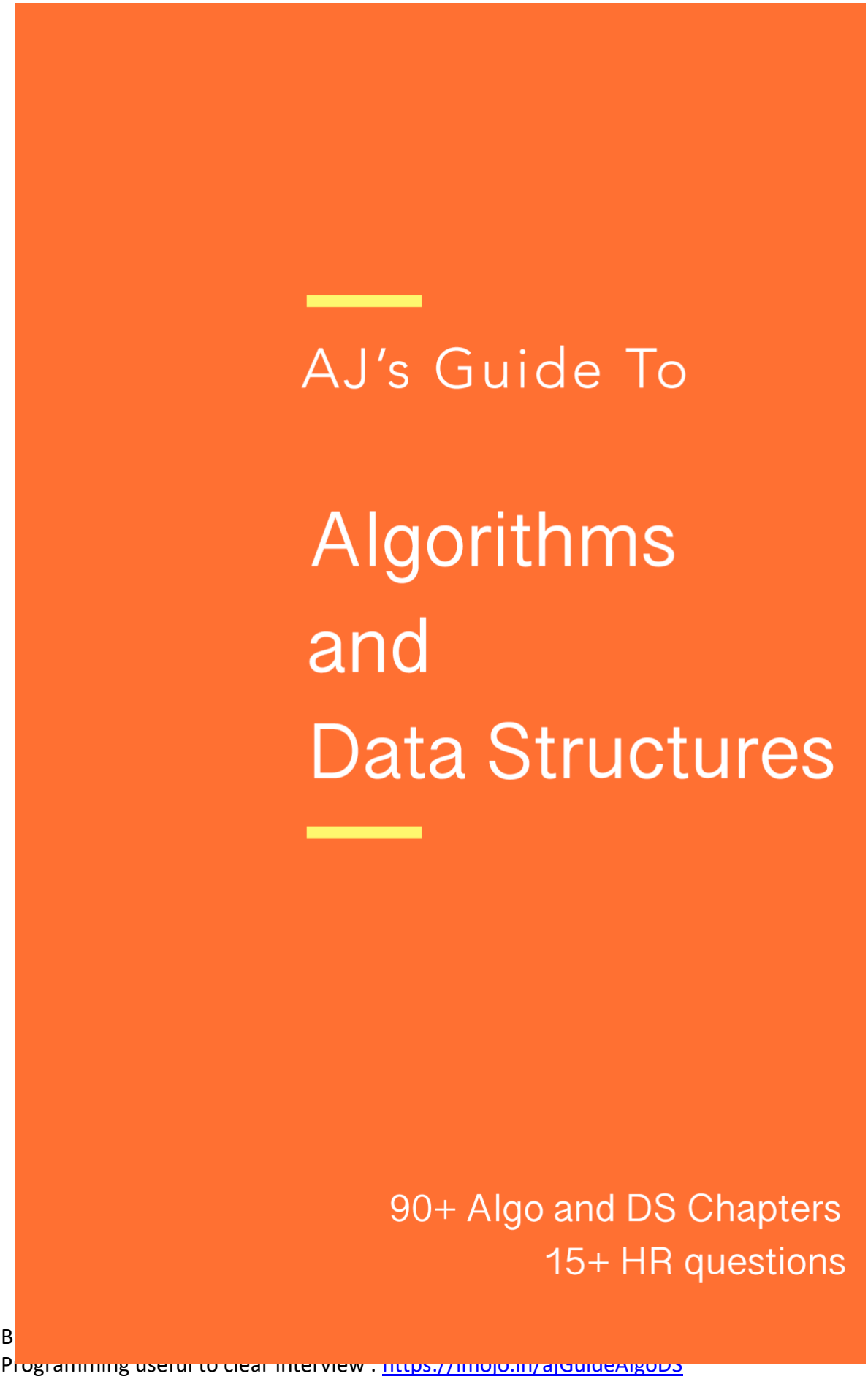
Python lay down Try, Except, Finally constructs to handle errors as well as Exceptions. We enclose the unsafe code indented under the try block. And we can keep our fall-back code inside the except block. Any instructions intended for execution last should come under the finally block.

```
try:
    print("Executing code in the try block")
    print(exception)
except:
    print("Entering in the except block")
```

```
finally:
print("Reached to the final block")
```

The output is:

```
Executing code in the try block
Entering in the except block
Reached to the final block
```



Q-66: How Do You Raise Exceptions For A Predefined Condition In Python?

We can raise an exception based on some condition.

For example, if we want the user to enter only odd numbers, else will raise an exception.

```
# Example - Raise an exception
while True:
try:
value = int(input("Enter an odd number- "))
if value%2 == 0:
raise ValueError("Exited due to invalid input!!!")
else:
print("Value entered is : %s" % value)
except ValueError as ex:
print(ex)
break
```

The output is:

```
Enter an odd number- 2
Exited due to invalid input!!!
Enter an odd number- 1
Value entered is : 1
Enter an odd number-
```

Q-67: What Are Python Iterators?

Iterators in Python are array-like objects which allow moving on the next element. We use them in traversing a loop, for example, in a “for” loop.

Python library has a no. of iterators. For example, a list is also an iterator and we can start a for loop over it.

Q-68: What Is The Difference Between An Iterator And Iterable?

The collection type like a list, tuple, dictionary, and set are all iterable objects whereas they are also iterable containers which return an iterator while traversing.

Here are some advanced-level Python interview questions.

Q-69: What Are Python Generators?

A Generator is a kind of function which lets us specify a function that acts like an iterator and hence can get used in a “for” loop.

In a generator function, the yield keyword substitutes the return statement.

```
# Simple Python function
def fn():
return "Simple Python function."

# Python Generator function
def generate():
yield "Python Generator function."

print(next(generate()))
```

The output is:

```
Python Generator function.
```

Q-70: What Are Closures In Python?

Python closures are function objects returned by another function. We use them to eliminate code redundancy.

In the example below, we’ve written a simple closure for multiplying numbers.

```
def multiply_number(num):
def product(number):
'product() here is a closure'
return num * number
return product

num_2 = multiply_number(2)
print(num_2(11))
print(num_2(24))

num_6 = multiply_number(6)
print(num_6(1))
```

The output is:

```
22
48
6
```

Q-71: What Are Decorators In Python?

Python decorator gives us the ability to add new behavior to the given objects dynamically. In the example below, we've written a simple example to display a message pre and post the execution of a function.

```
def decorator_sample(func):
def decorator_hook(*args, **kwargs):
print("Before the function call")
result = func(*args, **kwargs)
print("After the function call")
return result
return decorator_hook

@decorator_sample
def product(x, y):
"Function to multiply two numbers."
return x * y

print(product(3, 3))
```

The output is:

```
Before the function call
After the function call
9
```

Q-72: How Do You Create A Dictionary In Python?

Let's take the example of building site statistics. For this, we first need to break up the key-value pairs using a colon(":"). The keys should be of an immutable type, i.e., so we'll use the data-types which don't allow changes at runtime. We'll choose from an int, string, or tuple.

However, we can take values of any kind. For distinguishing the data pairs, we can use a comma(",") and keep the whole stuff inside curly braces({...}).

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> type(site_stats)
<class 'dict'>
>>> print(site_stats)
{'type': 'organic', 'site': 'tecbeamers.com', 'traffic': 10000}
```

Q-73: How Do You Read From A Dictionary In Python?

To fetch data from a dictionary, we can directly access using the keys. We can enclose a "key" using brackets [...] after mentioning the variable name corresponding to the dictionary.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats["traffic"])
```

We can even call the get method to fetch the values from a dict. It also let us set a default value. If the key is missing, then the KeyError would occur.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats.get('site'))
tecbeamers.com
```

Q-74: How Do You Traverse Through A Dictionary Object In Python?

We can use the “for” and “in” loop for traversing the dictionary object.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> for k, v in site_stats.items():
print("The key is: %s" % k)
print("The value is: %s" % v)
print("++++++++++++++++++++++++")
```

The output is:

```
The key is: type
The value is: organic
++++++++++++++++++++++++
The key is: site
The value is: tecbeamers.com
++++++++++++++++++++++++
The key is: traffic
The value is: 10000
++++++++++++++++++++++++
```

Q-75: How Do You Add Elements To A Dictionary In Python?

We can add elements by modifying the dictionary with a fresh key and then set the value to it.

```
>>> # Setup a blank dictionary
>>> site_stats = {}
>>> site_stats['site'] = 'google.com'
>>> site_stats['traffic'] = 10000000000
>>> site_stats['type'] = 'Referral'
>>> print(site_stats)
{'type': 'Referral', 'site': 'google.com', 'traffic': 10000000000}
```

We can even join two dictionaries to get a bigger dictionary with the help of the update() method.

```
>>> site_stats['site'] = 'google.co.in'
>>> print(site_stats)
{'site': 'google.co.in'}
>>> site_stats_new = {'traffic': 1000000, "type": "social media"}
>>> site_stats.update(site_stats_new)
>>> print(site_stats)
{'type': 'social media', 'site': 'google.co.in', 'traffic': 1000000}
```

Q-76: How Do You Delete Elements Of A Dictionary In Python?

We can delete a key in a dictionary by using the del() method.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> del site_stats["type"]
>>> print(site_stats)
{'site': 'google.co.in', 'traffic': 1000000}
```

Another method, we can use is the pop() function. It accepts the key as the parameter. Also, a second parameter, we can pass a default value if the key doesn't exist.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats.pop("type", None))
organic
>>> print(site_stats)
{'site': 'tecbeamers.com', 'traffic': 10000}
```

Q-77: How Do You Check The Presence Of A Key In A Dictionary?

We can use Python's "in" operator to test the presence of a key inside a dict object.

```
>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> 'site' in site_stats
True
>>> 'traffic' in site_stats
True
>>> "type" in site_stats
True
```

Earlier, Python also provided the has_key() method which got deprecated.

Q-78: What Is The Syntax For List Comprehension In Python?

The signature for the list comprehension is as follows:

```
[ expression(var) for var in iterable ]
```

For example, the below code will return all the numbers from 10 to 20 and store them in a list.

```
>>> alist = [var for var in range(10, 20)]
>>> print(alist)
```

Q-79: What Is The Syntax For Dictionary Comprehension In Python?

A dictionary has the same syntax as was for the list comprehension but the difference is that it uses curly braces:

```
{ aKey, itsValue for aKey in iterable }
```

For example, the below code will return all the numbers 10 to 20 as the keys and will store the respective squares of those numbers as the values.

```
>>> adict = {var:var**2 for var in range(10, 20)}
>>> print(adict)
```

Q-80: What Is The Syntax For Generator Expression In Python?

The syntax for generator expression matches with the list comprehension, but the difference is that it uses parenthesis:

```
( expression(var) for var in iterable )
```

For example, the below code will create a generator object that generates the values from 10 to 20 upon using it.

```
>>> (var for var in range(10, 20))
at 0x0000000003668728>
>>> list((var for var in range(10, 20)))
```

Now, see more Python interview questions for practice.

Q-81: How Do You Write A Conditional Expression In Python?

We can utilize the following single statement as a conditional expression.
default_statment if Condition else another_statement

```
>>> no_of_days = 366
>>> is_leap_year = "Yes" if no_of_days == 366 else "No"
>>> print(is_leap_year)
Yes
```

Q-82: What Do You Know About The Python Enumerate?

While using the iterators, sometimes we might have a use case to store the count of iterations. Python gets this task quite easy for us by giving a built-in method known as the enumerate().

The enumerate() function attaches a counter variable to an iterable and returns it as the "enumerated" object.

We can use this object directly in the "for" loops or transform it into a list of tuples by calling the list() method. It has the following signature:

```
enumerate(iterable, to_begin=0)
Arguments:
iterable: array type object which enables iteration
to_begin: the base index for the counter is to get started, its default value is 0
# Example - enumerate function
alist = ["apple","mango", "orange"]
astr = "banana"

# Let's set the enumerate objects
list_obj = enumerate(alist)
str_obj = enumerate(astr)

print("list_obj type:", type(list_obj))
print("str_obj type:", type(str_obj))

print(list(enumerate(alist)) )
# Move the starting index to two from zero
print(list(enumerate(astr, 2)))
```

The output is:

```
list_obj type: <class 'enumerate'>
str_obj type: <class 'enumerate'>
[(0, 'apple'), (1, 'mango'), (2, 'orange')]
[(2, 'b'), (3, 'a'), (4, 'n'), (5, 'a'), (6, 'n'), (7, 'a')]
```

Q-83: What Is The Use Of Globals() Function In Python?

The globals() function in Python returns the current global symbol table as a dictionary object.

Python maintains a symbol table to keep all necessary information about a program. This info includes the names of variables, methods, and classes used by the program.

All the information in this table remains in the global scope of the program and Python allows us to retrieve it using the globals() method.

```
Signature: globals()

Arguments: None
# Example: globals() function
x = 9
def fn():
    y = 3
    z = y + x
# Calling the globals() method
    z = globals()['x'] = z
    return z

# Test Code
ret = fn()
print(ret)
```

The output is:

```
12
```

Q-84: Why Do You Use The Zip() Method In Python?

The zip method lets us map the corresponding index of multiple containers so that we can use them using as a single unit.

```
Signature:
zip(*iterators)
Arguments:
Python iterables or collections (e.g., list, string, etc.)
Returns:
A single iterator object with combined mapped values
# Example: zip() function

emp = [ "tom", "john", "jerry", "jake" ]
age = [ 32, 28, 33, 44 ]
dept = [ 'HR', 'Accounts', 'R&D', 'IT' ]

# call zip() to map values
out = zip(emp, age, dept)

# convert all values for printing them as set
out = set(out)
```

```
# Displaying the final values
print ("The output of zip() is : ",end="")
print (out)
```

The output is:

```
The output of zip() is : {('jerry', 33, 'R&D'), ('jake', 44, 'IT'), ('john', 28, 'Accounts'), ('tom', 32, 'HR')}
```

Q-85: What Are Class Or Static Variables In Python Programming?

In Python, all the objects share common class or static variables.

But the instance or non-static variables are altogether different for different objects.

The programming languages like C++ and Java need to use the static keyword to make a variable as the class variable. However, Python has a unique way to declare a static variable.

All names initialized with a value in the class declaration becomes the class variables. And those which get assigned values in the class methods becomes the instance variables.

```
# Example
class Test:
aclass = 'programming' # A class variable
def __init__(self, ainst):
self.ainst = ainst # An instance variable

# Objects of CSStudent class
test1 = Test(1)
test2 = Test(2)

print(test1.aclass)
print(test2.aclass)
print(test1.ainst)
print(test2.ainst)

# A class variable is also accessible using the class name
print(Test.aclass)
```

The output is:

```
programming
programming
```

```
1
2
programming
```

Let's now answer some advanced-level Python interview questions.

Q-86: How Does The Ternary Operator Work In Python?

The ternary operator is an alternative for the conditional statements. It combines true or false values with a statement that you need to test.

The syntax would look like the one given below.

[onTrue] if [Condition] else [onFalse]

```
x, y = 35, 75
smaller = x if x < y else y
print(smaller)
```

Q-87: What Does The "Self" Keyword Do?

The **self** is a Python keyword which represents a variable that holds the instance of an object.
In almost, all the object-oriented languages, it is passed to the methods as a hidden parameter.

Q-88: What Are The Different Methods To Copy An Object In Python?

There are two ways to copy objects in Python.

- **copy.copy() function**
- It makes a copy of the file from source to destination.
- It'll return a shallow copy of the parameter.
- **copy.deepcopy() function**
- It also produces the copy of an object from the source to destination.
- It'll return a deep copy of the parameter that you can pass to the function.

Q-89: What Is The Purpose Of Docstrings In Python?

In Python, the docstring is what we call as the docstrings. It sets a process of recording Python functions, modules, and classes.

Q-90: Which Python Function Will You Use To Convert A Number To A String?

For converting a number into a string, you can use the built-in function **str()**. If you want an octal or hexadecimal representation, use the inbuilt function **oct()** or **hex()**.

Also Check.

Python Multithreading Quiz

Q-91: How Do You Debug A Program In Python? Is It Possible To Step Through The Python Code?

Yes, we can use the Python debugger (**pdb**) to debug any Python program. And if we start a program using **pdb**, then it let us even step through the code.

Q-92: List Down Some Of The PDB Commands For Debugging Python Programs?

Here are a few PDB commands to start debugging Python code.

- Add breakpoint **(b)**
- Resume execution **(c)**
- Step by step debugging **(s)**
- Move to the next line **(n)**
- List source code **(l)**
- Print an expression **(p)**

Q-93: What Is The Command To Debug A Python Program?

The following command helps run a Python program in debug mode.

```
$ python -m pdb python-script.py
```

Q-94: How Do You Monitor The Code Flow Of A Program In Python?

In Python, we can use **the sys** module's **settrace()** method to setup trace hooks and monitor the functions inside a program.
You need to define a trace callback method and pass it to the **settrace()** function. The callback should specify three arguments as shown below.

```
import sys

def trace_calls(frame, event, arg):
# The 'call' event occurs before a function gets executed.
if event != 'call':
return
# Next, inspect the frame data and print information.
print 'Function name=%s, line num=%s' % (frame.f_code.co_name, frame.f_lineno)
return

def demo2():
print 'in demo2()'

def demo1():
print 'in demo1()'
```

```
demo2()


sys.settrace(trace_calls)
demo1()
```

Q-95: Why And When Do You Use Generators In Python?

A generator in Python is a function which returns an iterable object. We can iterate on the generator object using the **yield** keyword. But we can only do that once because their values don't persist in memory, they get the values on the fly.
Generators give us the ability to hold the execution of a function or a step as long as we want to keep it. However, here are a few examples where it is beneficial to use generators.

- We can replace loops with generators for efficiently calculating results involving large data sets.
- Generators are useful when we don't want all the results and wish to hold back for some time.
- Instead of using a callback function, we can replace it with a generator. We can write a loop inside the function doing the same thing as the callback and turns it into a generator.

Q-96: What Does The Yield Keyword Do In Python?

The **yield** keyword can turn any function into a generator. It works like a standard return keyword. But it'll always return a generator object. Also, a method can have multiple calls to the **yield** keyword.
See the example below.

```
def testgen(index):
weekdays = ['sun','mon','tue','wed','thu','fri','sat']
yield weekdays[index]
yield weekdays[index+1]

day = testgen(0)
print next(day), next(day)

#output: sun mon
```

Q-97: How To Convert A List Into Other Data Types?

Sometimes, we don't use lists as is. Instead, we have to convert them to other types.

Turn A List Into A String.

We can use the **".join()** method which combines all elements into one and returns as a string.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat']
```

```
listAsString = ' '.join(weekdays)
print(listAsString)


#output: sun mon tue wed thu fri sat
```

Turn A List Into A Tuple.

Call Python's **tuple()** function for converting a list into a tuple.
This function takes the list as its argument.

But remember, we can't change the list after turning it into a tuple because it becomes immutable.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat']
listAsTuple = tuple(weekdays)
print(listAsTuple)


#output: ('sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat')
```

Turn A List Into A Set.

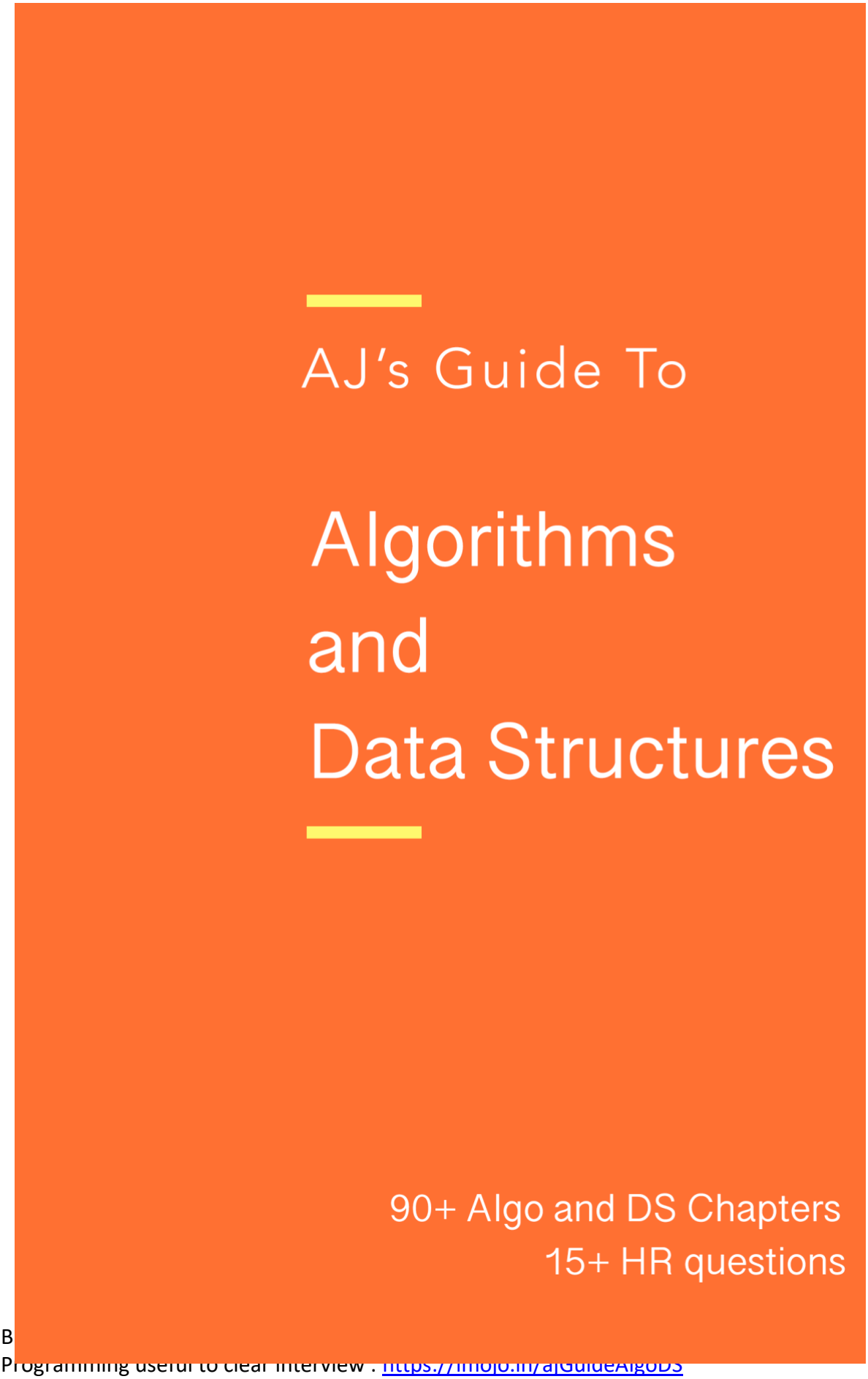
Converting a list to a set poses two side-effects.

- Set doesn't allow duplicate entries so that the conversion will remove any such item.
- A set is an ordered collection, so the order of list items would also change.

However, we can use the **set()** function to convert a list into a Set.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sat','sun','tue']
listAsSet = set(weekdays)
print(listAsSet)


#output: set(['wed', 'sun', 'thu', 'tue', 'mon', 'fri', 'sat'])
```



Turn A List Into A Dictionary.

In a dictionary, each item represents a key-value pair. So converting a list isn't as straightforward as it were for other data types.

However, we can achieve the conversion by breaking the list into a set of pairs and then call the **zip()** function to return them as tuples.
Passing the tuples into the **dict()** function would finally turn them into a dictionary.

```
weekdays = ['sun','mon','tue','wed','thu','fri']
listAsDict = dict(zip(weekdays[0::2], weekdays[1::2]))
print(listAsDict)


#output: {'sun': 'mon', 'thu': 'fri', 'tue': 'wed'}
```

Q-98: How Do You Count The Occurrences Of Each Item Present In The List Without Explicitly Mentioning Them?

Unlike sets, lists can have items with the same values.

In Python, the list has a **count()** function which returns the occurrences of a particular item.

Count The Occurrences Of An Individual Item.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']
print(weekdays.count('mon'))


#output: 3
```

Count The Occurrences Of Each Item In The List.

We'll use the list comprehension along with the **count()** method. It'll print the frequency of each of the items.

```
weekdays = ['sun','mon','tue','wed','thu','fri','sun','mon','mon']
print([[x,weekdays.count(x)] for x in set(weekdays)])


#output: [['wed', 1], ['sun', 2], ['thu', 1], ['tue', 1], ['mon', 3], ['fri', 1]]
```

Q-99: What Is NumPy And How Is It Better Than A List In Python?

NumPy is a Python package for scientific computing which can deal with large data sizes. It includes a powerful N-dimensional array object and a set of advanced functions.

Also, the NumPy arrays are superior to the built-in lists. There are a no. of reasons for this.

- NumPy arrays are more compact than lists.

- Reading and writing items is faster with NumPy.
- Using NumPy is more convenient than to the standard list.
- NumPy arrays are more efficient as they augment the functionality of lists in Python.

Q-100: What Are Different Ways To Create An Empty NumPy Array In Python?

There are two methods which we can apply to create empty NumPy arrays.

The First Method To Create An Empty Array.

```
import numpy
numpy.array([])
```

The Second Method To Create An Empty Array.

```
# Make an empty NumPy array
numpy.empty(shape=(0,0))
```

Q #1) Can Python be used for web client and web server side programming? And which one is best suited to Python?

Answer: Python is best suited for web server-side application development due to its vast set of features for creating business logic, database interactions, web server hosting etc.

However, Python can be used as a web client-side application which needs some conversions for a browser to interpret the client side logic. Also, note that Python can be used to create desktop applications which can run as a standalone application such as utilities for test automation.

Q #2) Mention at least 3-4 benefits of using Python over the other scripting languages such as Javascript.

Answer: Enlisted below are some of the benefits of using Python.

1. Application development is faster and easy.
2. Extensive support of modules for any kind of application development including data analytics/machine learning/math-intensive applications.
3. An excellent support community to get your answers.

Q #3) Explain List, Tuple, Set, and Dictionary and provide at least one instance where each of these collection types can be used.

Answer:

- **List:** Collection of items of different data types which can be changed at run time.
- **Tuple:** Collection of items of different data types which cannot be changed. It only has read-only access to the collection. This can be used when you want to secure your data collection set and does not need any modification.
- **Set:** Collection of items of a similar data type.
- **Dictionary:** Collection of items with key-value pairs.

Generally, List and Dictionary are extensively used by programmers as both of them provide flexibility in data collection.

Q #4) Does Python allow you to program in a structured style?

Answer: Yes. It does allow to code is a structured as well as Object-oriented style. It offers excellent flexibility to design and implement your application code depending on the requirements of your application.

Q #5) What is PIP software in the Python world?

Answer: PIP is an acronym for Python Installer Package which provides a seamless interface to install various Python modules. It is a command line tool which can search for packages over the internet and install them without any user interaction.

Q #6) What should be the typical build environment for Python based application development?

Answer: You just need to install Python software and using PIP, you can install various Python modules from the open source community.

For IDE, Pycharm is highly recommended for any kind of application development with vast support for plugins. Another basic IDE is called a RIDE and is a part of the Python open source community.

Q #7) What tools can be used to unit test your Python code?

Answer: The best and easiest way is to use 'unittest' python standard library to test units/classes. The features supported are very similar to the other unit testing tools such as JUnit, TestNG.

Q #8) How does For loop and While loop differ in Python and when do you choose to use them?

Answer: For loop is generally used to iterate through the elements of various collection types such as List, Tuple, Set, and Dictionary.
While loop is the actual looping feature that is used in any other programming language. This is how Python differs in handling loops from the other programming languages.

Q #9) How are data types defined in Python and how much bytes do integer and decimal data types hold?

Answer: In Python, there is no need to define a variable's data type explicitly.
Based on the value assigned to a variable, Python stores the appropriate data type. In the case of numbers such as Integer, Float, etc, the length of data is unlimited.

Q #10) How do you make use of Arrays in Python?

Answer: Python does not support Arrays. However, you can use List collection type which can store an unlimited number of elements.

Q #11) How do you implement JSON given that Python is best suited for the server-side application?

Answer: Python has built-in support to handle JSON objects.
You just have to import the JSON module and use the functions such as loads and dumps to convert from JSON string to JSON object and vice versa. It is a straightforward way to handle and exchange JSON based data from the server-side.

Q #12) What is the best way to parse strings and find patterns in Python?

Answer: Python has built-in support to parse strings using Regular expression module. Import the module and use the functions to find a sub-string, replace a part of a string, etc.

Q #13) Which databases are supported by Python?

Answer: MySQL (Structured) and MongoDB (Unstructured) are the prominent databases that are supported natively in Python. Import the module and start using the functions to interact with the database.

Q #14) What is the purpose of _init_() function in Python?

Answer: It is the first function that gets executed when an object of a class is instantiated. This is equivalent to the constructor concept in C++.

Q #15) What is the significance of 'self' parameter in an object method? Should we always name this parameter as 'self'?

Answer: Parameter 'self' is used to refer to the object properties of a class.
'self' parameter is supposed to be prefixed to the class object properties. The answer to the second part of the question is No. 'self' parameter can have any name.

Q #16) How does Lambda function differ from a normal function in Python?

Answer: Lambda is similar to the inline function in C programming. It returns a function object. It contains only one expression and can accept any number of arguments.
In case of a normal function, you can define a function name, pass the parameter and mandatorily have a return statement. The Lambda function can be typically used for simple operations without the use of function names. It can also be used in the place of a variable.

Q #17) How is Exception Handling done in Python?

Answer: There are 3 main keywords i.e. try, except and finally which are used to catch exceptions and handle the recovering mechanism accordingly. Try is the block of a code which is monitored for errors. Except block gets executed when an error occurs.
The beauty of the final block is to execute the code after trying for error. This block gets executed irrespective of whether an error occurred or not. Finally block is used to do the required cleanup activities of objects/variables.

Q #18) What is the starting point of Python code execution?
Answer: As Python is an interpreter, it starts reading the code from the source file and starts executing them.
However, if you want to start from the main function, you should have the following special variable set in your source file as:
if__name__== "__main__
main()

Q #19) Name some of the important modules that are available in Python.
Answer: Networking, Mathematics, Cryptographic services, Internet data handling, and Multi-threading modules are prominent modules. Apart from these, there are several other modules that are available in the Python developer community.

Q #20) Which module(s) of Python can be used to measure the performance of your application code?
Answer: Time module can be used to calculate the time at different stages of your application and use the Logging module to log data to a file system in any preferred format.

Q #21) How do you launch sub-processes within the main process of a Python application?
Answer: Python has a built-in module called sub-process. You can import this module and either use run() or Popen() function calls to launch a sub-process and get the control of its return code.

Q #22) As Python is more suitable for the server-side application, it is very important to have threading implemented in your server code. How can you achieve that in Python?
Answer: We should use the threading module to implement, control and destroy threads for parallel execution of the server code. Locks and Semaphores are available as synchronization objects to manage data between different threads.

Q #23) Do we need to call the explicit methods to destroy the memory allocated in Python?
Answer: Garbage collection is an in-built feature in Python which takes care of allocating and de-allocating memory. This is very similar to the feature in Java. Hence, there are very fewer chances of memory leaks in your application code.

Q #24) Does the same Python code work on multiple platforms without any changes?
Answer: Yes. As long as you have the Python environment on your target platform (Linux, Windows, Mac), you can run the same code.

Q #25) How can you create a GUI based application in Python for client-side functionality?
Answer: Python along with standard library Tkinter can be used to create GUI based applications. Tkinter library supports various widgets which can create and handle events which are widget specific.

Q #26) What are the different environment variables identified by Python?

Answer:
PYTHONPATH: This environment variable helps the interpreter as to where to locate the module files imported in the program.
PYTHONSTARTUP: This environment variable contains the path of the Initialization file containing source code.
PYTHONCASEOK: This variable is used to find the first case-insensitive match in the import statement

Q #27) What is Python Tuples and how is it different from Lists?
Answer: Tuples is basically a sequence of elements which are separated by commas and are enclosed in parenthesis.
Lists whereas is a sequence of elements which are separated by commas and are enclosed in brackets. Also, Tuples cannot be updated whereas, in lists, elements can be updated along with their sizes.

Q #28) What does '#' symbol do in Python?
Answer: '#' is used to comment out everything that comes after on the line.
Example:

```
print ("I am a beginner in Python")

#print ("I am a beginner in Python")
```

Output:
I am a beginner in Python

Q #29) What does stringVar.strip() does?
Answer: This is one of the string methods which removes leading/trailing white space.

Q #30) What should be the output of the following code:

```
a="pythontutorial"

print('%. 6s' % a)
```

Answer: Output should be: python

Q #31) Write a command to read:
a. '10' characters from a file
b. Read entire file
c. Write output after executing both commands together.

Where the file name is "softwaretestinghelp.txt".

File text:
Python is a powerful high-level, object-oriented programming language created by Guido van Rossum.
It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.
Answer:

```
f = open ("softwaretestinghelp.txt ", "r")

print (f. read (10))
```

```
print (f. read ())
```

Output:
Python

is a powerful high-level, object-oriented programming language created by Guido van Rossum.

It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Q #32) What are membership operators in Python? Write an example to explain both.
Answer: There are 2 types of membership operators in Python:
in: If the value is found in a sequence, then the result becomes true else false
not in: If the value is not found in a sequence, then the result becomes true else false
Example:

```
1 a=15
2 b=30
3 list= [3,6,15,20,30];
4
5 if (a in list)
6 print "a is available in given list"
7 else
8 print "a is not available in given list"
9
10 if (b not in list)
11 print "b is not available in given list"
12 else
13 print "b is available in given list"
```

Output:
a is available in given list

b is available is list

Q #33) Write a code to display the current time.
Answer:

```
currenttime= time.localtime(time.time())

print ("Current time is", currenttime)
```

Q #34) What is the output of print str[4:] if str = ' Python Language'?
Answer:
Output: on Language

Q #35) Write the command to get all keys from the dictionary.
Answer: *print dict.keys()*
Q #36) Write a command to convert a string into an int in python.
Answer: *int(x [,base])*
Q #37) What are a help () and dir() in python?
Answer: help () is a built-in function that can be used to return the Python documentation of a particular object, method, attributes, etc.
dir () displays a list of attributes for the objects which are passed as an argument. If dir() is without the argument then it returns a list of names in current local space.

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions

Q #38) What does the term 'Monkey Patching' refers to in Python?
Answer: Monkey Patching refers to the modification of a module at run-time.
Q #39) What do you mean by 'suites' in Python?
Answer: The group of individual statements, thereby making a logical block of code is called suites
Example:

```
If expression

Suite

Else

Suite
```

Q #40) What is range () in Python? Give an example to explain it.
Answer: It is a function to iterate over a sequence of numbers.
Example:

```
for var in list(range (10))

Print (var)
```

Q #41) What is the difference between abs () and fabs ()?
Answer: abs () is a built-in function which works with integer, float and complex numbers also.
fabs () is defined in math module which doesn't work with complex numbers.

Q #42) Write the output for the following code:
Code:

```
str = "Python is a programming language"

print (str.isalnum())

str = "This is Interview Question17"

print (str.isalnum())
```

Answer: False
True

Q #43) What is a from import statement and write the syntax for it?
Answer: From statement allows specific attributes to be imported from a module in a current namespace.
Syntax: from modname import name1[, name2[, … nameN]]
Q #44) What is the difference between locals() and globals ()?

Answer: locals() is accessed within the function and it returns all names that can be accessed locally from that function.
globals() returns all names that can be accessed globally from that function.

Q #45) What is the use of Assertions in Python?
Answer: Assert statement is used to evaluate the expression attached. If the expression is false, then python raised AssertionError Exception.
Q #46) What is the difference between 'match' and 'search' in Python?
Answer: Match checks for the match at the beginning of the string whereas search checks for the match anywhere in the string
Q #47) What is the difference between a shallow copy and deep copy?
Answer: Shallow copy is used when a new instance type gets created and it keeps values that are copied whereas deep copy stores values that are already copied.
A shallow copy has faster program execution whereas deep coy makes it slow.

Q #48) What statement is used in Python if the statement is required syntactically but no action is required for the program?
Answer: Pass statement
Example:

```
If(a>10)

print("Python")

else

pass
```

Q #49) What does PEP8 refer to?
Answer: PEP8 is a coding convention which is a set of recommendations of how to make the code more readable.
Q #50) What are *args and *kwargs?
Answer: They are used to pass a variable number of arguments to a function. *args is used to pass non-keyworded, variable length argument list whereas *kwargs is used to pass keyworded, variable length argument list.

Question: **Describe how multithreading is achieved in Python.**
Answer: Even though Python comes with a multi-threading package, if the motive behind multithreading is to speed the code then using the package is not the go-to option.

The package has something called the GIL or Global Interpreter Lock, which is a construct. It ensures that one and only one of the threads execute at any given time. A thread acquires the GIL and then do some work before passing it to the next thread.

This happens so fast that to a user it seems that threads are executing in parallel. Obviously, this is not the case as they are just taking turns while using the same CPU core. Moreover, GIL passing adds to the overall overhead to the execution.

Hence, if you intend to use the threading package for speeding up the execution, using the package is not recommended.

Question: **Draw a comparison between the range and xrange in Python.**
Answer: In terms of functionality, both range and xrange are identical. Both allow for generating a list of integers. The main difference between the two is that while range returns a Python list object, xrange returns an xrange object.

Xrange is not able to generate a static list at runtime the way range does. On the contrary, it creates values along with the requirements via a special technique called yielding. It is used with a type of object known as generators.

If you have a very enormous range for which you need to generate a list, then xrange is the function to opt for. This is especially relevant for scenarios dealing with a memory-sensitive system, such as a smartphone.

The range is a memory beast. Using it requires much more memory, especially if the requirement is gigantic. Hence, in creating an array of integers to suit the needs, it can result in a Memory Error and ultimately lead to crashing the program.

Question: **Explain Inheritance and its various types in Python?**
Answer: Inheritance enables a class to acquire all the members of another class. These members can be attributes, methods, or both. By providing reusability, inheritance makes it easier to create as well as maintain an application.

The class which acquires is known as the child class or the derived class. The one that it acquires from is known as the superclass or base class or the parent class. There are 4 forms of inheritance supported by Python:

- Single Inheritance – A single derived class acquires from on single superclass.
- Multi-Level Inheritance – At least 2 different derived classes acquire from two distinct base classes.
- Hierarchical Inheritance – A number of child classes acquire from one superclass

- Multiple Inheritance – A derived class acquires from several superclasses.

Question: **Explain how is it possible to Get the Google cache age of any URL or webpage using Python.**
Answer: In order to Get the Google cache age of any URL or webpage using Python, following URL format is used:

http://webcache.googleusercontent.com/search?q=cache:URLGOESHERE

Simply replace URLGOESHERE with the web address of the website or webpage whose cache you need to retrieve and see in Python.

Question: **Give a detailed explanation about setting up the database in Django.**
Answer: The process of setting up a database is initiated by using the command edit mysite/setting.py. This is a normal Python module with a module-level representation of Django settings. Django relies on SQLite by default, which is easy to be used as it doesn't require any other installation.

SQLite stores data as a single file in the filesystem. Now, you need to tell Django how to use the database. For this, the project's setting.py file needs to be used. Following code must be added to the file for making the database workable with the Django project:

```
DATABASES = {

    'default': {

'ENGINE' : 'django.db.backends.sqlite3',

'NAME' : os.path.join(BASE_DIR, 'db.sqlite3'),

    }

}
```

If you need to use a database server other than the SQLite, such as MS SQL, MySQL, and PostgreSQL, then you need to use the database's administration tools to create a brand new database for your Django project.

You have to modify the following keys in the DATABASE 'default' item to make the new database work with the Django project:

- **ENGINE** – For example, when working with a MySQL database replace 'django.db.backends.sqlite3' with 'django.db.backends.mysql'
- **NAME** – Whether using SQLite or some other database management system, the database is typically a file on the system. The NAME should contain the full path to the file, including the name of that particular file.

NOTE: – Settings like Host, Password, and User needs to be added when not choosing SQLite as the database.

Check out the advantages and disadvantages of Django.

Question: **How will you differentiate between deep copy and shallow copy?**
Answer: We use shallow copy when a new instance type gets created. It keeps the values that are copied in the new instance. Just like it copies the values, the shallow copy also copies the reference pointers.

Reference points copied in the shallow copy reference to the original objects. Any changes made in any member of the class affects the original copy of the same. Shallow copy enables faster execution of the program.

Deep copy is used for storing values that are already copied. Unlike shallow copy, it doesn't copy the reference pointers to the objects. Deep copy makes the reference to an object in addition to storing the new object that is pointed by some other object.

Changes made to the original copy will not affect any other copy that makes use of the referenced or stored object. Contrary to the shallow copy, deep copy makes execution of a program slower. This is due to the fact that it makes some copies for each object that is called.

Question: **How will you distinguish between NumPy and SciPy?**
Answer: Typically, NumPy contains nothing but the array data type and the most basic operations, such as basic element-wise functions, indexing, reshaping, and sorting. All the numerical code resides in SciPy.

As one of NumPy's most important goals is compatibility, the library tries to retain all features supported by either of its predecessors. Hence, NumPy contains a few linear algebra functions despite the fact that these more appropriately belong to the SciPy library.

SciPy contains fully-featured versions of the linear algebra modules available to NumPy in addition to several other numerical algorithms.

People Also Read:

- NumPy Matrix Multiplication

Question: **Observe the following code:**

```
A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))

A1 = range(10)A2 = sorted([i for i in A1 if i in A0])

A3 = sorted([A0[s] for s in A0])

A4 = [i for i in A1 if i in A3]

A5 = {i:i*i for i in A1}

A6 = [[i,i*i] for i in A1]

print(A0,A1,A2,A3,A4,A5,A6)
```

Write down the output of the code.
Answer:

```
A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4} # the order may vary
A1 = range(0, 10)
A2 = []
A3 = [1, 2, 3, 4, 5]
A4 = [1, 2, 3, 4, 5]
A5 = {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]
```

Question: **Python has something called the dictionary. Explain using an example.**
Answer: A dictionary in Python programming language is an unordered collection of data values such as a map. Dictionary holds key:value pair. It helps in defining a one-to-one relationship between keys and values. Indexed by keys, a typical dictionary contains a pair of keys and corresponding values.

Let us take an example with three keys, namely Website, Language, and Offering. Their corresponding values are hackr.io, Python, and Tutorials. The code for the example will be:

```
dict={'Website':'hackr.io','Language':'Python':'Offering':'Tutorials'}

print dict[Website] #Prints hackr.io

print dict[Language] #Prints Python

print dict[Offering] #Prints Tutorials
```

Question: **Python supports negative indexes. What are they and why are they used?**
Answer: The sequences in Python are indexed. It consists of the positive and negative numbers. Positive numbers use 0 as the first index, 1 as the second index, and so on. Hence, any index for a positive number n is n-1.

Unlike positive numbers, index numbering for the negative numbers start from -1 and it represents the last index in the sequence. Likewise, -2 represents the penultimate index. These are known as negative indexes. Negative indexes are used for:

- Removing any new-line spaces from the string, thus allowing the string to except the last character, represented as S[:-1]
- Showing the index to representing the string in the correct order

Question: **Suppose you need to collect and print data from IMDb top 250 Movies page. Write a program in Python for doing so. (NOTE: – You can limit the displayed information for 3 fields; namely movie name, release year, and rating.)**
Answer:

```
from bs4 import BeautifulSoup

import requests
```

```
import sys

url = 'http://www.imdb.com/chart/top'

response = requests.get(url)

soup = BeautifulSoup(response.text)

tr = soup.findChildren("tr")

tr = iter(tr)

next(tr)

for movie in tr:

title = movie.find('td', {'class': 'titleColumn'} ).find('a').contents[0]

year = movie.find('td', {'class': 'titleColumn'} ).find('span', {'class':
'secondaryInfo'}).contents[0]

rating = movie.find('td', {'class': 'ratingColumn imdbRating'}
).find('strong').contents[0]

row = title + ' - ' + year + ' ' + ' ' + rating

print(row)
```

Question: Take a look at the following code:

```
try:

if '1' != 1:

raise "someError"

else:

print("someError has not occured")

except "someError":

print ("someError has occured")
```

What will be the output?
Answer: The output of the program will be "invalid code." This is because a new exception class must inherit from a BaseException.

Question: **What do you understand by monkey patching in Python?**
Answer: The dynamic modifications made to a class or module at runtime are termed as monkey patching in Python. Consider the following code snippet:

```
# m.py

class MyClass:

def f(self):

print "f()"
```

We can monkey-patch the program something like this:

```
import m

def monkey_f(self):

print "monkey_f()"

m.MyClass.f = monkey_f

obj = m.MyClass()

obj.f()
```

Output for the program will be monkey_f().

The examples demonstrate changes made in the behavior of f() in MyClass using the function we defined i.e. monkey_f() outside of the module m.

Question: **What do you understand by the process of compilation and linking in Python?**
Answer: In order to compile new extensions without any error, compiling and linking is used in Python. Linking initiates only and only when the compilation is complete.

In the case of dynamic loading, the process of compilation and linking depends on the style that is provided with the concerned system. In order to provide dynamic loading of the configuration setup files and rebuilding the interpreter, the Python interpreter is used.

Question: **What is Flask and what are the benefits of using it?**
Answer: Flask is a web microframework for Python with Jinja2 and Werkzeug as its dependencies. As such, it has some notable advantages:

- Flask has little to no dependencies on external libraries
- Because there is a little external dependency to update and fewer security bugs, the web microframework is lightweight to use.
- Features an inbuilt development server and a fast debugger.

Question: **What is the map() function used for in Python?**
Answer: The map() function applies a given function to each item of an iterable. It then returns a list of the results. The value returned from the map() function can then be passed on to functions to the likes of the list() and set().

Typically, the given function is the first argument and the iterable is available as the second argument to a map() function. Several tables are given if the function takes in more than one arguments.

Question: **What is Pickling and Unpickling in Python?**
Answer: The Pickle module in Python allows accepting any object and then converting it into a string representation. It then dumps the same into a file by means of the dump function. This process is known as pickling.

The reverse process of pickling is known as unpickling i.e. retrieving original Python objects from a stored string representation.

Question: **Whenever Python exits, all the memory isn't deallocated. Why is it so?**
Answer: Upon exiting, Python's built-in effective cleanup mechanism comes into play and try to deallocate or destroy every other object.

However, Python modules that are having circular references to other objects or the objects that are referenced from the global namespaces aren't always deallocated or destroyed.

This is because it is not possible to deallocate those portions of the memory that are reserved by the C library.

Question: **Write a program in Python for getting indices of N maximum values in a NumPy array.**
Answer:

```
import numpy as np

arr = np.array([1, 3, 2, 4, 5])

print(arr.argsort()[-3:][::-1])

Output:

[4 3 1]
```

Question: **Write code to show randomizing the items of a list in place in Python along with the output.**
Answer:

```
from random import shuffle

x = ['hackr.io', 'Is', 'The', 'Best', 'For', 'Learning', 'Python']
```

```
shuffle(x)

print(x)

Output:

['For', 'Python', 'Learning', 'Is', 'Best', 'The', 'hackr.io']
```

That's All!

1. What is Python?

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, it can be used to build almost any type of application with the right tools/libraries. Additionally, python supports objects, modules, threads, exception-handling and automatic memory management which help in modelling real-world problems and building applications to solve these problems.

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions

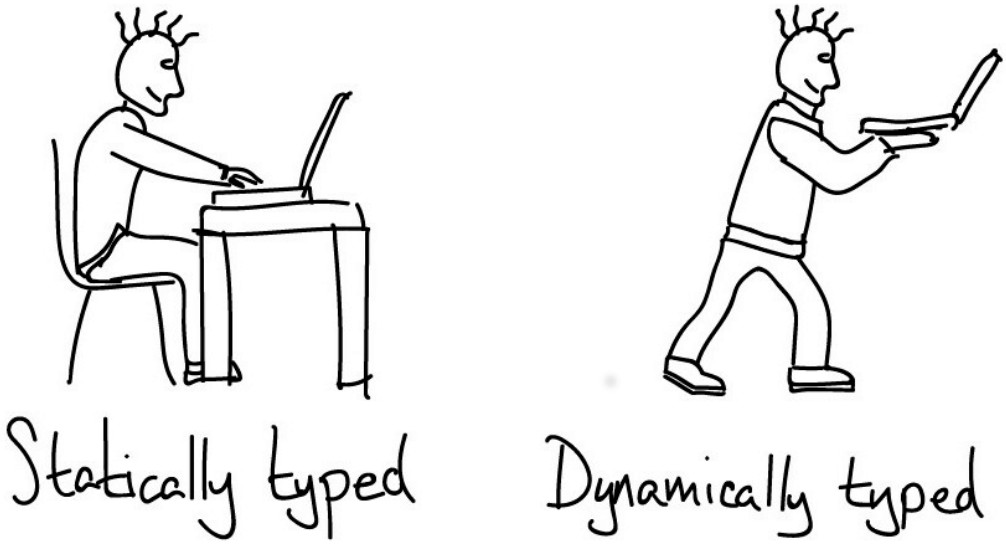
2. What are the benefits of using Python?

Python is a general-purpose programming language that has simple, easy-to-learn syntax which emphasizes readability and therefore reduces the cost of program maintenance. Moreover, the language is capable of scripting, completely open-source and supports third-party packages encouraging modularity and code-reuse. Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment.

3. What is a dynamically typed language?

Before we understand what a dynamically typed language, we should learn about what typing is. **Typing** refers to type-checking in programming languages. In a ***strongly-typed*** language, such as Python, **"1" + 2** will result in a type error, since these languages don't allow for **"type-coercion"** (implicit conversion of data types). On the other hand, a ***weakly-typed*** language, such as Javascript, will simply output **"12"** as result.
Type-checking can be done at two stages -

1. **Static -** Data Types are checked before execution.
2. **Dynamic -** Data Types are checked during execution.
   Python being an interpreted language, executes each statement line by line and thus type-checking is done on the fly, during execution. Hence, Python is a Dynamically Typed language.



4. What is an Interpreted language?

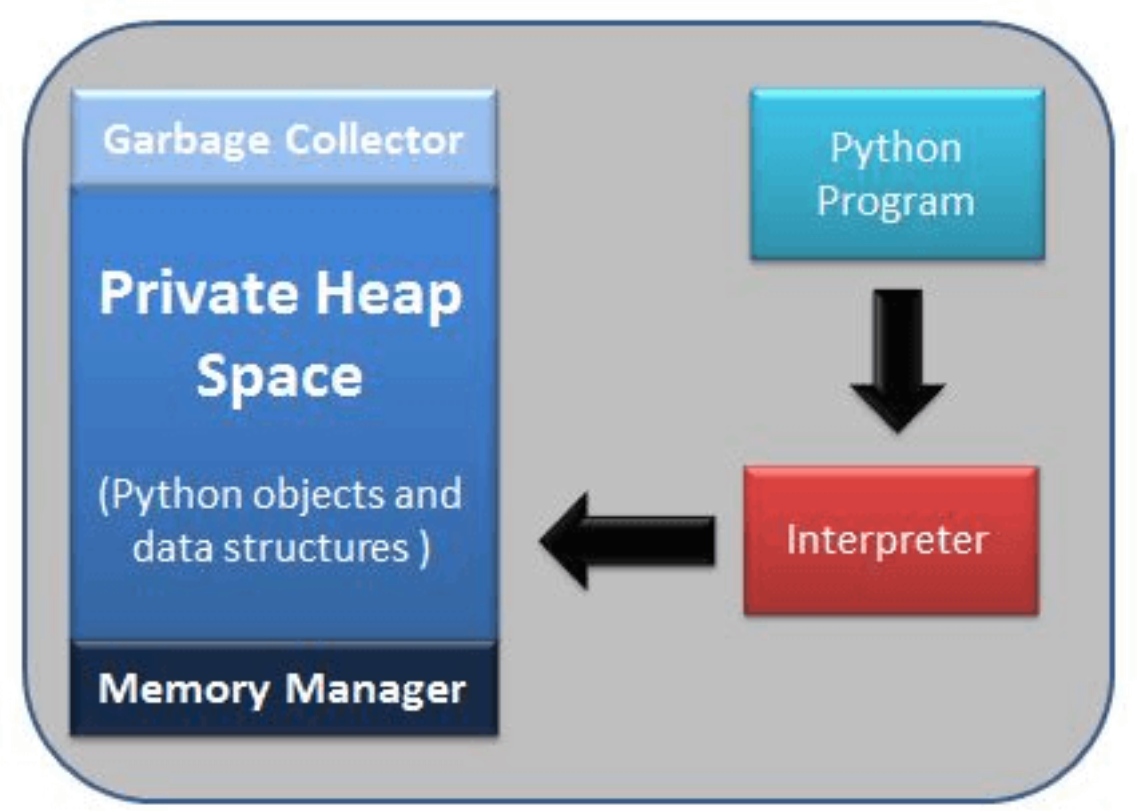
An **Interpreted** language executes its statements line by line. Languages such as Python, Javascript, R, PHP and Ruby are prime examples of Interpreted languages. Programs written in an interpreted language runs directly from the source code, with no intermediary compilation step.

5. What is PEP 8 and why is it important?

PEP stands for **Python Enhancement Proposal**. A PEP is an official design document providing information to the Python Community, or describing a new feature for Python or its processes. **PEP 8** is especially important since it documents the style guidelines for Python Code. Apparently contributing in the Python open-source community requires you to follow these style guidelines sincerely and strictly.

6. How is memory managed in Python?

Memory management in Python is handled by the **Python Memory Manager**. The memory allocated by the manager is in form of a **private heap space** dedicated for Python. All Python objects are stored in this heap and being private, it is inaccessible to the programmer. Though, python does provide some core API functions to work upon the private heap space.
Additionally, Python has an in-built garbage collection to recycle the unused memory for the private heap space.

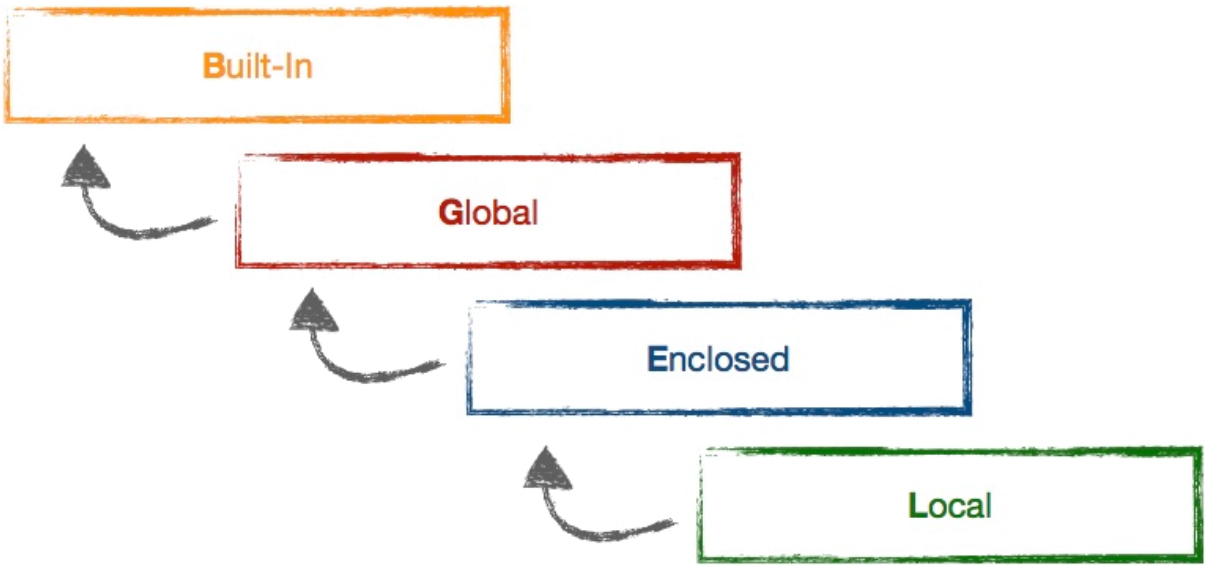


7. What are Python namespaces? Why are they used?

A namespace in Python ensures that object names in a program are unique and can be used without any conflict. Python implements these **namespaces as dictionaries**with 'name as key' mapped to a corresponding 'object as value'. This allows for multiple namespaces to use the same name and map it to a separate object. A few examples of namespaces are as follows:

- **Local Namespace** includes local names inside a function. the namespace is temporarily created for a function call and gets cleared when the function returns.
- **Global Namespace** includes names from various imported packages/ modules that is being used in the current project. This namespace is created when the package is imported in the script and lasts until the execution of the script.
- **Built-in Namespace** includes built-in functions of core Python and built-in names for various types of exceptions.

Lifecycle of a namespace depends upon the scope of objects they are mapped to. If the scope of an object ends, the lifecycle of that namespace comes to an end. Hence, it isn't possible to access inner namespace objects from an outer namespace.



8. What is Scope in Python?

Every object in Python functions within a scope. A **scope** is a block of code where an object in Python remains relevant. Namespaces uniquely identify all the objects inside a program. However, these namespaces also have a scope defined for them where you could use their objects without any prefix. A few examples of scope created during code execution in Python are as follows:

1. A **local scope** refers to the local objects available in the current function.
2. A **global scope** refers to the objects available throught the code execution since their inception.
3. A **module-level scope** refers to the global objects of the current module accessible in the program.
4. An **outermost scope** refers to all the built-in names callable in the program. The objects in this scope are searched last to find the name referenced.

*Note: Local scope objects can be synced with global scope objects using keywords such as **global**.*

9. What is Scope Resolution in Python?

Sometimes objects within the same scope have the same name but function differently. In such cases, scope resolution comes into play in Python automatically. A few examples of such behaviour are:

- Python modules namely 'math' and 'cmath' have a lot of functions that are common to both of them - `log10()`, `acos()`, `exp()` etc. To resolve this amiguity, it is necessary to prefix them with their respective module, like `math.exp()` and `cmath.exp()`.
- Consider the code below, an object temp has been initialized to 10 globally and then to 20 on function call. However, the function call didn't change the value of the temp globally. Here, we can observe that Python draws a clear line between global and local variables treating both their namespaces as separate identities.

```
temp = 10          # global-scope variable
```

```
def func():
    temp = 20  # local-scope variable
    print(temp)

print(temp)         # output => 10
func()              # output => 20
print(temp)         # output => 10
```

This behaviour can be overriden using the `global` keyword inside the function, as shown in the following example:

```
temp = 10           # global-scope variable

def func():
    global temp
    temp = 20  # local-scope variable
    print(temp)

print(temp)         # output => 10
func()              # output => 20
print(temp)         # output => 20
```

10. What are decorators in Python?

Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented by the `@decorator_name` in Python and are called in bottom-up fashion. For example:

```
# decorator function to convert to lowercase
def lowercase_decorator(function):
   def wrapper():
      func = function()
      string_lowercase = func.lower()
      return string_lowercase
   return wrapper

# decorator function to split words
def splitter_decorator(function):
   def wrapper():
```

```
    func = function()
    string_split = func.split()
    return string_split
  return wrapper

@splitter_decorator         # this is executed next
@lowercase_decorator        # this is executed first
def hello():
  return 'Hello World'

hello()    # output => [ 'hello' , 'world' ]
```

The beauty of the decorators lies in the fact that besides adding functionality to the output of the method, they can even **accept arguments** for functions and can further modify those arguments before passing it to the function itself. The **inner nested function**, i.e. 'wrapper' function, plays a significant role here. It is implemented to enforce **encapsulation** and thus, keep itself hidden from the global scope.

```
# decorator function to capitalize names
def names_decorator(function):
  def wrapper(arg1, arg2):
    arg1 = arg1.capitalize()
    arg2 = arg2.capitalize()
    string_hello = function(arg1, arg2)
    return string_hello
  return wrapper

@names_decorator
def say_hello(name1, name2):
  return 'Hello ' + name1 + '! Hello ' + name2 + '!'

say_hello('sara', 'ansh')      # output => 'Hello Sara! Hello Ansh!'
```

11. What are lists and tuples? What is the key difference between the two?

Lists and **Tuples** are both **sequence data types** that can store a collection of objects in Python. The objects stored in both sequences can have **different data types**. Lists are represented with **square brackets** `['sara', 6, 0.19]`, while tuples are represented with **parantheses** `('ansh', 5, 0.97)`.

But what is the real difference between the two? The key difference between the two is that while **lists are mutable**, **tuples** on the other hand **are immutable** objects. This

means that lists can be modified, appended or sliced on-the-go but tuples remain constant and cannot be modified in any manner. You can run the following example on Python IDLE to confirm the difference:

```
my_tuple = ('sara', 6, 5, 0.97)
my_list = ['sara', 6, 5, 0.97]

print(my_tuple[0])    # output => 'sara'
print(my_list[0])    # output => 'sara'

my_tuple[0] = 'ansh'    # modifying tuple => throws an error
my_list[0] = 'ansh'    # modifying list => list modified

print(my_tuple[0])    # output => 'sara'
print(my_list[0])    # output => 'ansh'
```

12. What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic sugar** constructs that help **build altered and filtered** lists, dictionaries or sets from a given list, dictionary or set. Using comprehensions, saves a lot of time and code that might be considerably more verbose (containing more lines of code). Let's check out some examples, where comprehensions can be truly beneficial:

- **Performing mathematical operations on the entire list**

```
my_list = [2, 3, 5, 7, 11]

squared_list = [x**2 for x in my_list]    # list comprehension
# output => [4 , 9 , 25 , 49 , 121]

squared_dict = {x:x**2 for x in my_list}    # dict comprehension
# output => {11: 121, 2: 4 , 3: 9 , 5: 25 , 7: 49}
```

- **Performing conditional filtering operations on the entire list**

```
my_list = [2, 3, 5, 7, 11]

squared_list = [x**2 for x in my_list if x%2 != 0]    # list comprehension
# output => [9 , 25 , 49 , 121]

squared_dict = {x:x**2 for x in my_list if x%2 != 0}    # dict comprehension
# output => {11: 121, 3: 9 , 5: 25 , 7: 49}
```

- **Combining multiple lists into one**

Comprehensions allow for multiple iterators and hence, can be used to combine multiple lists into one.

```
a = [1, 2, 3]
b = [7, 8, 9]

[(x + y) for (x,y) in zip(a,b)]  # parallel iterators
# output => [8, 10, 12]

[(x,y) for x in a for y in b]   # nested iterators
# output => [(1, 7), (1, 8), (1, 9), (2, 7), (2, 8), (2, 9), (3, 7), (3, 8), (3, 9)]
```

- **Flattening a multi-dimensional list**

  A similar approach of nested iterators (as above) can be applied to flatten a multi-dimensional list or work upon its inner elements.

```
my_list = [[10,20,30],[40,50,60],[70,80,90]]

flattened = [x for temp in my_list for x in temp]
# output => [10, 20, 30, 40, 50, 60, 70, 80, 90]
```

Note: *List comprehensions have the same effect as the* `map` *method in other languages. They follow the* ***mathematical set builder notation*** *rather than* `map` *and* `filter` *functions in Python.*

13. What are the common built-in data types in Python?

There are several built-in data types in Python. Although, Python doesn't require data types to be defined explicitly during variable declarations but type errors are likely to occur if the knowledge of data types and their compatibility with each other are neglected. Python provides `type()` and `isinstance()` functions to check the type of these variables. These data types can be grouped into the following catetgories-

- **None Type**

  `None` keyword represents the null values in Python. Boolean equality operation can be performed using these NoneType objects.

Class Name	Description
NoneType	Represents the **NULL** values in Python

- **Numeric Types**

  There are three distint numeric types - ***integers***, ***floating-point numbers***, and ***complex numbers***. Additionally, ***booleans*** are a sub-type of integers.

Class Name	Description
int	Stores integer literals including hex, octal and binary numbers as integers
float	Stores literals containing decimal values and/or exponent sign as floating
complex	Stores complex number in the form (A + Bj) and has attributes: `real` and `im`

bool	Stores boolean value (True or False)

- ***Note:*** *The standard library also includes* ***fractions*** *to store rational numbers and* ***decimal*** *to store floating-point numbers with user-defined precision.*

- **Sequence Types**

  According to Python Docs, there are three basic Sequence Types - ***lists***, ***tuples***, and ***range objects***. Sequence types have the `in` and `not in` operators defined for their traversing their elements. These operators share the same priority as the comparison operations.

Class Name	Description
list	Mutable sequence used to store collection of items.
tuple	Immutable sequence used to store collection of items.
range	Represents an immutable sequence of numbers generated during exe
str	Immutable sequence of Unicode code points to store textual data.

- ***Note:*** *The standard library also includes additional types for processing:*
  1. ***Binary data*** *such as* `bytearray` `bytes` `memoryview`, *and*
  2. ***Text strings*** *such as* `str`.

- **Mapping Types**

  A mapping object can map *hashable values* to random objects in Python. Mappings objects are mutable and there is currently only one standard mapping type, the ***dictionary***.

Class Name	Description
dict	Stores comma-separated list of **key: value** pairs

-

- **Set Types**

  Currently, Python has two built-in set types - **set** and **frozenset**. **set** type is mutable and supports methods like `add()` and `remove()`. **frozenset** type is immutable and can't be modified after creation.

Class Name	Description
set	Mutable unordered collection of distinct hashable objects
frozenset	Immutable collection of distinct hashable objects

- ***Note:*** `set` *is mutable and thus cannot be used as key for a dictionary. On the other hand,* `frozenset` *is immutable and thus, hashable, and can be used as a dictionary key or as an element of another set.*

- **Modules**

  Module is an additional built-in type supported by the Python Interpreter. It supports one special operation, i.e., **attribute access**: `mymod.myobj`, where **mymod** is a module and **myobj** references a name defined in m's symbol table. The module's symbol table resides in a very special attribute of the module **__dict__**, but direct assignment to this

module is neither possible nor recommended.

- **Callable Types**
  Callable types are the types to which function call can be applied. They can be **user-defined functions**, **instance methods**, **generator functions**, and some other **built-in functions**, **methods** and **classes**.
  Refer the documentation at docs.python.org for a detailed view into the **callable types**.

14. What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period. Lambda functions can be used in either of the two ways:

- Assigning lambda functions to a variable

```
mul = lambda a, b : a * b
print(mul(2, 5))    # output => 10
```

- Wrapping lambda functions inside another function

```
def myWrapper(n):
  return lambda a : a * n

mulFive = myWrapper(5)
print(mulFive(2))    # output => 10
```

15. What is pass in Python?

The `pass` keyword represents a null operation in Python. It is generally used for the purpose of filling up empty blocks of code which may execute during runtime but has yet to be written. Without the **pass** statement in the following code, we may run into some errors during code execution.

```
def myEmptyFunc():
  # do nothing
  pass


myEmptyFunc()    # nothing happens


## Without the pass keyword
# File "<stdin>", line 3
# IndentationError: expected an indented block
```

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions

16. How do you copy an object in Python?

In Python, the assignment statement (= operator) does not copy objects. Instead, it creates a binding between the existing object and the target variable name. To create copies of an object in Python, we need to use the **copy** module. Moreover, there are two ways of creating copies for the given object using the **copy** module -

- **Shallow Copy** is a bit-wise copy of an object. The copied object created has an exact copy of the values in the original object. If either of the values are references to other objects, just the reference addresses for the same are copied.
- **Deep Copy** copies all values recursively from source to target object, i.e. it even duplicates the objects referenced by the source object.

```
from copy import copy, deepcopy


list_1 = [1, 2, [3, 5], 4]

## shallow copy

list_2 = copy(list_1)
list_2[3] = 7
list_2[2].append(6)

list_2   # output => [1, 2, [3, 5, 6], 7]
list_1   # output => [1, 2, [3, 5, 6], 4]

## deep copy

list_3 = deepcopy(list_1)
list_3[3] = 8
list_3[2].append(7)

list_3   # output => [1, 2, [3, 5, 6, 7], 8]
list_1   # output => [1, 2, [3, 5, 6], 4]
```

17. What is the difference between xrange and range in Python?

xrange() and **range()** are quite similar in terms of functionality. They both generate a sequence of integers, with the only difference that `range()` returns a **Python list**, whereas, `xrange()` returns an **xrange object**.

So how does that make a difference? It sure does, because unlike range(), xrange() doesn't generate a static list, it creates the value on the go. This technique is commonly used with an object type **generators** and has been termed as **"yielding"**.
Yielding is crucial in applications where memory is a constraint. Creating a static list as in range() can lead to a `Memory Error` in such conditions, while, xrange() can handle it optimally by using just enough memory for the generator (significantly less in comparison).

```
for i in xrange(10):    # numbers from o to 9
   print i      # output => 0 1 2 3 4 5 6 7 8 9


for i in xrange(1,10):    # numbers from 1 to 9
   print i      # output => 1 2 3 4 5 6 7 8 9


for i in xrange(1, 10, 2):    # skip by two for next
   print i      # output => 1 3 5 7 9
```

***Note: xrange** has been **deprecated** as of **Python 3.x**. Now `range` does exactly the same what `xrange` used to do in **Python 2.x**, since it was way better to use xrange() than the original range() function in Python 2.x.*

18. What are modules and packages in Python?

Python packages and Python modules are two mechanisms that allow for **modular programming** in Python. Modularizing ahs several advantages -

1. **Simplicity:** Working on a single module helps you focus on a relatively small portion of the problem at hand. This makes development easier and less error-prone.
2. **Maintainability:** Modules are designed to enforce logical boundaries between different problem domains. If they are written in a manner that reduces interdependency, it is less likely that modifications in a module might impact other parts of the program.
3. **Reusability:** Functions defined in a module can be easily reused by other parts of the application.
4. **Scoping:** Modules typically define a separate **namespace**, which helps avoid confusion between identifiers from other parts of the program.

Modules, in general, are simply Python files with a .py extension and can have a set of functions, classes or variables defined and implemented. They can be imported and initialized once using the `import` statement. If partial functionality is needed, import the requisite classes or functions using `from foo import bar`.

Packages allow for hierarchial structuring of the module namespace using **dot notation**. As, **modules** help avoid clashes between global variable names, in a similary manner, **packages** help avoid clashes between module names. Creating a package is easy since it makes use of the system's inherent file structure. So just stuff the modules into a folder and there you have it, the folder name as the package name. Importing a module or its contents from this package requires the package name as prefix to the module name joined by a dot.

***Note:** You can technically import the package as well, but alas, it doesn't import the modules within the package to the local namespace, thus, it is practically useless.*

19. What are global, protected and private attributes in Python?

- **Global** variables are public variables that are defined in the global scope. To use the variable in the global scope inside a function, we use the `global` keyword.
- **Protected** attributes are attributes defined with a underscore prefixed to their identifier eg. **_sara**. They can still be accessed and modified from outside the class they are defined in but a responsible developer should refrain from doing so.
- **Private** attributes are attributes with double underscore prefixed to their identifier eg. **__ansh**. They cannot be accessed or modified from the outside directly and will result in an AttributeError if such an attempt is made.

20. What is self in Python?

Self is a keyword in Python used to define an instance or an object of a class. In Python, it is explicity used as the first paramter, unlike in Java where it is optional. It helps in disinguishing between the methods and attributes of a class from its local variables.

21. What is __init__?

`__init__` is a contructor method in Python and is automatically called to allocate memory when a new object/instance is created. All classes have a **__init__** method associated with them. It helps in distinguishing methods and attributes of a class from local variables.

```
# class definition
class Student:
    def __init__(self, fname, lname, age, section):
        self.firstname = fname
        self.lastname = lname
        self.age = age
        self.section = section

# creating a new object
stu1 = Student("Sara", "Ansh", 22, "A2")
```

22. What is break, continue and pass in Python?

Break	The break statement terminates the loop immediately and the control flows to the statement after the body of the loop.
Continue	The continue statement terminates the current iteration of the statement, skips the rest of the code in the current iteration and the control flows to the next
Pass	As explained above, pass keyword in Python is generally used to fill-up empty bl and is similar to an empty statement represented by a semi-colon in languages s

```
pat = [1, 3, 2, 1, 2, 3, 1, 0, 1, 3]


for p in pat:
    pass
    if (p == 0):
        current = p
        break
    elif (p % 2 == 0):
        continue
    print(p)    # output => 1 3 1 3 1

print(current)    # output => 0
```

23. What is pickling and unpickling?

Python library offers a feature - **serialization** out of the box. Serializing a object refers to transforming it into a format that can be stored, so as to be able to deserialize it later on, to obtain the original object. Here, the **pickle** module comes into play.

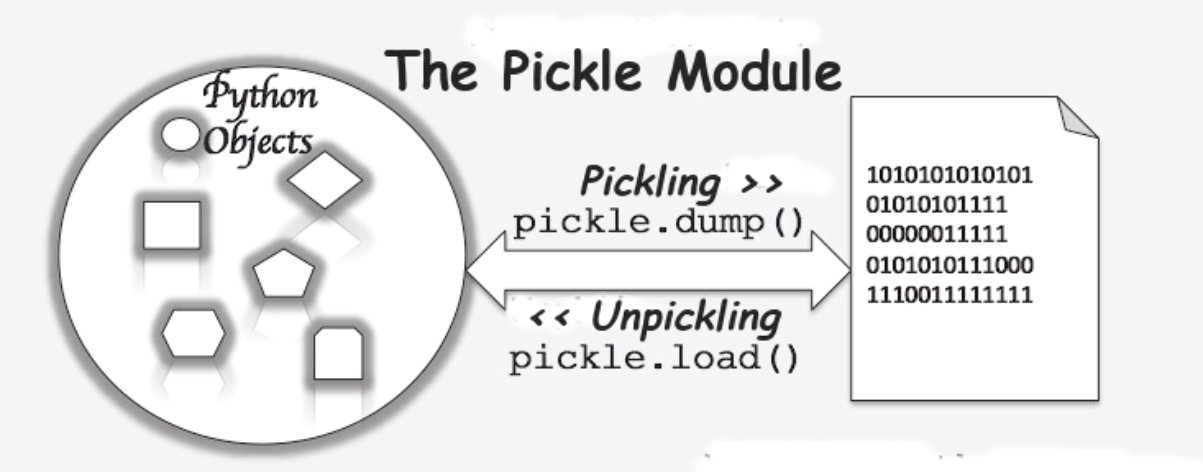
Pickling

Pickling is the name of the serialization process in Python. Any object in Python can be serialized into a byte stream and dumped as a file in the memory. The process of pickling is compact but pickle objects can be compressed further. Moreover, pickle keeps track of the objects it has serialized and the serialization is portable across versions. The function used for the above process is `pickle.dump()`.

Unpickling

Unpickling is the complete inverse of pickling. It deserializes the byte stream to recreate the objects stored in the file, and loads the object to memory. The function used for the above process is `pickle.load()`.

Note: *Python has another, more primitive, serialization module called* ***marshall****, which exists primarily to* ***support .pyc files*** *in Python and* ***differs significantly from pickle****.*



24. What are generators in Python?

Generators are functions that return an iterable collection of items, one at a time, in a set manner. Generators, in general, are used to create iterators with a different approach. They employ the use of `yield` keyword rather than `return` to return a **generator object**. Let's try and build a generator for fibonacci numbers -

```
## generate fibonacci numbers upto n
def fib(n):
    p, q = 0, 1
    while(p < n):
        yield p
        p, q = q, p + q

x = fib(10)  # create generator object

## iterating using __next__(), for Python2, use next()
x.__next__()  # output => 0
x.__next__()  # output => 1
x.__next__()  # output => 1
x.__next__()  # output => 2
x.__next__()  # output => 3
x.__next__()  # output => 5
x.__next__()  # output => 8
x.__next__()  # error

## iterating using loop
for i in fib(10):
```

```
print(i)   # output => 0 1 1 2 3 5 8
```

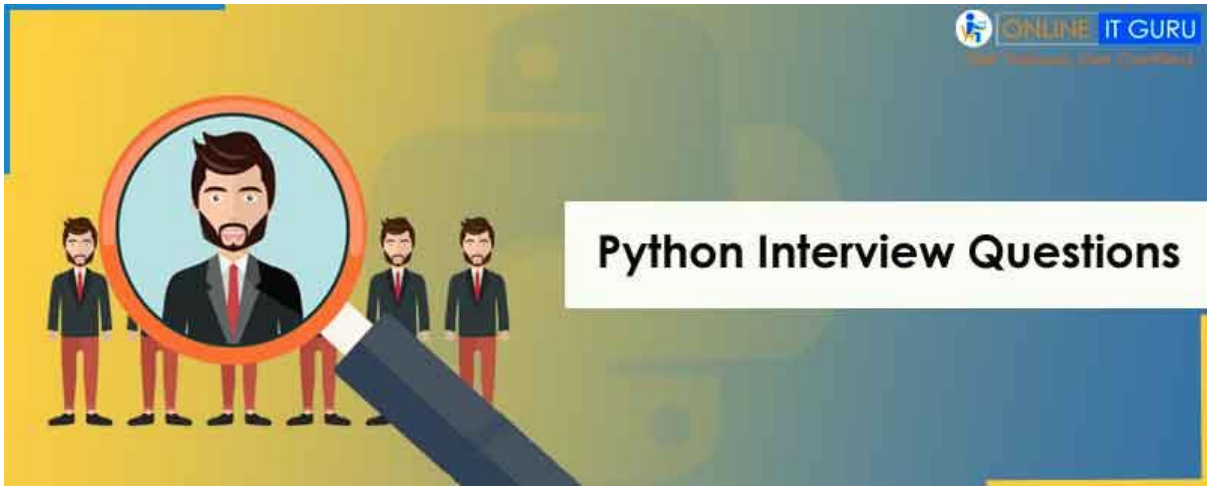
25. What is PYTHONPATH in Python?

PYTHONPATH is an environment variable which you can set to add additional directories where Python will look for modules and packages. This is especially useful in maintaining Python libraries that you do not wish to install in the global default location.

26. What is the use of help() and dir() functions?

help() function in Python is used to display the **documentation** of modules, classes, functions, keywords, etc. If no parameter is passed to the help() function, then an interactive **help utility** is launched on the console.
dir() function tries to return a valid list of attributes and methods of the object it is called upon. It behaves differently with different objects, as it aims to produce the most relevant data, rather than the complete information.

- For Modules/Library objects, it returns a list of all attributes, contained in that module.
- For Class Objects, it returns a list of all valid attributes and base attributes.
- With no arguments passed, it returns a list of attributes in the current scope.



You are right place, If you are looking for **python interview questions** and answers, get more confidence to crack interview by reading this questions and answers we will update more and more latest questions for you…

1. What is Python?

Python is an Interpreted, high level, and object-arranged programming language. Python is intended to be exceptionally clear. It utilizes the English language frequently whereas different languages use punctuation, and it has less syntactical construction than the other languages.

[Related Article – What is Python Programming?]

2. Mention the python features

Following are a portion of the remarkable highlights of python −

It supports functional and structured programming techniques just as OOP.

Python tends to be utilized as a scripting language or can be aggregated to byte-code for structure enormous applications.

It gives high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection

Python tends to be effectively integrated with C, C++, COM, ActiveX, CORBA, and Java.

3. What is the reason for PYTHONPATH condition variable?

PYTHONPATH – It has a job like PATH. This variable advises the Python interpreter where to find the module records brought into a program. It should incorporate the Python source library registry and the indexes containing Python source code. PYTHONPATH is some times preset by the Python installer.

4. Mention the python supported data types?

Python has five standard data types −

```
Numbers

String

List

Tuple

Dictionary
```

5. How Python is an interpreted language?

An interpreted language is any programming language which isn't in machine level code before runtime. So, Python is a translated language.

[Related Article – Python Basics]

6. What is namespace in Python?

A namespace is a naming system used to ensure that names are extraordinary to avoid naming clashes.

7. What is PYTHONPATH?

It is an environment variable which is utilized when a module is imported. At whatever point a module is imported, PYTHONPATH is looked up to check for the presence of the imported modules in different registries. The translator utilizes it to figure out which module is load.

8. What are the functions in Python?

A function is the block of code which is executed just when it is called. To define the function in Python, the def keyword is utilized.

```
Ex:

def Newfunc():

print ("Hi, Welcome to onlineit")

Newfunc(); #calling the functiondef Newfunc():
```

[Related Article – Python Function]

9. What is a lambda function?

An anonymous function is known as a lambda function. This function can have any number of parameters at the same time, but only one statement.

```
Example:

1.a = lambda x,y : x+y

2.print(a(5, 6))

Result: 11
```

10. What is self in Python?

The self is an example or an object of a class. In Python, this is explicitly included as the principal parameter. Be that as it may, this isn't the situation in Java where it's optional. It separates between the techniques and characteristics of a class with local variables

The self-variable in the init technique refers to the recently made object while in different methods, it refers to the object whose method is called.

11. What are python iterators?

Iterators are objects which can be crossed traversed or iterated upon.

[Related Article – Python Iterator]

12. In what capacity will you capitalize the first letter of string?

In Python, the capitalize () method capitalizes the first letter of a string. if the string now comprises of a capital letter toward the start, at that point, it restores the original string.

13. How to comment multiple lines in python?

Multi-line comments show up in more than one line. Every one of the lines to be commented is to be prefixed by a #. You can likewise an excellent easy route technique to comment on different lines. You should simply hold the Ctrl key and left snap in each spot any place you need to incorporate a # character and type a # just once. This will comment on each line where you presented your cursor.

14. What are docstrings in Python?

Docstrings are not really comments, at the same time, they are documentation strings. These docstrings are inside triple statements. They are not assigned to any factor and considered as comments as well.

```
Example:
1."""
Utilizing docstring as a remark.
This code combines 2 numbers
4."""
x=8
y=4
z=x/y
print(z)
Result:  2.0
```

15. What is the reason for is, not and in operators?

Operators are specifical functions. They take at least one value and produce a comparing result.

Is: returns genuine when 2 operands are valid (Example: “a” will be ‘a’)

Not: restores the backwards of the Boolean value

In: checks if some component is available in some sequence

[Related Article – Python Operators]

16. What is the utilization of help() and dir() function in Python?

Help() and dir() the two functions are available from the Python interpreter and utilized for survey a united dump of built-in functions

Help() function: The assistance() work is utilized to show the documentation string and furthermore encourages you to see the assistance identified with modules, watchwords, qualities, and so on.

Dir() function: The dir() function is utilized to show the defined symbols.

17. How To Find Bugs Or Perform Static Analysis In A Python Application?

PyChecker, which is a static analyser is the best tool to find bugs (or) to perform static analysis. It distinguishes the bugs in Python project and furthermore uncovers the style and complexity related bugs. Another tool is Pylint, which checks whether the Python module fulfills the coding standard.

18. When Is The Python Decorator Used?

Python decorator is a relative change that you do in Python programming structure to alter the functions rapidly.

19. What Is The Main Difference Between A List And The Tuple?

List Vs. Tuple.

The vital contrast between a list and the tuple is that the list is alterable while the tuple isn't.

A tuple is permitted to be hashed, for instance, utilizing it as a key for dictionaries.

20. How Does Python Handle Memory Management?

Python utilizes private heaps to keep up its memory. So the data holds all the Python objects and the data structures. This area is just open to the Python translator; developers can't utilize it. Additionally, it's the Python memory chief that handles the Private heap. It does the required designation of the memory for Python objects. Python utilizes a built – in garbage collector, which rescues all the unused memory and offloads it to the heap space.

21. What Is A String In Python?

A string in Python is a combination of alpha-numeric characters. They are unchanging objects. It implies that they don't permit alteration once they get allocated a value. Python provides a few methods, for example, join(), supplant(), or split() to change strings. Be that as it may, none of these changes the original object.

22. What Is Slicing In Python?

Slicing is a string activity for separating a piece of the string or some piece of a list. In Python, a string (say data) starts at list 0, and the nth character stores at position text[n-1]. Python can likewise perform invert ordering, i.e., in the regressive heading, with the assistance of negative numbers. In Python, the Slice() is additionally a constructor function which produces a slice object. The outcome is a lot of files referenced by range(start, stop, step). The Slice() strategy permits three parameters. 1. Start – beginning number for the slice to start. 2. stop – the number which demonstrates the finish of slice. 3. step – the value to augment after each record (default = 1).

23. What Is The Index In Python?

The Index is an integer data type which indicates a position inside an ordered list or a string. In Python, strings are likewise list of characters. We can get to them utilizing the file which starts from zero and goes to the length short one.

```
For instance, in the string "Program," the ordering happens this way:


Program 0 1 2 3 4 5
```

24. What Is Docstring In Python?

A docstring is one of a kind content that happens to be the first statements in the accompanying Python constructs:

Module, Function, Class, or Method definition.

```
A docstring gets added to the __doc__ property of the string object.
```

25. What Is A Function In Python Programming?

A function is an object which describes to a piece of code and is a reusable element. It carries measured quality to a program and a higher level of code re usability. Python has given us many built-in capacities, for example, print () and gives the ability to create user-defined functions.

[Related Article – Python Function]

26. What Is The Return Value Of The Trunc() Function?

The Python trunc() work plays out mathematical operations to expel the decimal qualities from a particular expression and gives integer value as the output.

27. It Mandatory For A Python Function To Return A Value?

It isn't at all essential for a function to restore any value. In any case, if necessary, we can utilize none as the incoming value.

28. Explain the role of *Continue* in Python

The continue is a jump in Python which moves the control to execute the next cycle in a loop leaving all the rest of the guidelines in the block unexecuted.

29. What Is The Purpose Of Id() Function In Python?

The id() is one of the built-in functions in Python.

```
syntax: id(object)
```

30. When Should You Use The "Break" In Python?

Python provides a break statement to exit from a circle. At any point, if the break hits in the code, the control of the program promptly exits from the body of the circle. The break statement in the nested loop causes the controlled exit from the inner iterative square.

31. What Is the Difference between Pass and Continue In Python?

The continue statement makes the loop to continue from the next cycle. And the pass statement instructs to do nothing, and the rest of the code executes obviously.

PHP is a recursive acronym for PHP Hypertext Preprocessor. It is a widely used open-source programming language especially suited for creating dynamic websites and mobile API's.

Below PHP interview questions are helpful for **1 year, 2 years, 5 years experience PHP developer**

Core PHP Interview Questions for Beginners

1. What is namespaces in PHP?

PHP Namespaces provide a way of grouping related classes, interfaces, functions and constants.

```
# define namespace and class in namespace
namespace Modules\Admin\;
class CityController {
}
# include the class using namesapce
use Modules\Admin\CityController ;
```

2. How to add 301 redirects in PHP?

You can add **301 redirect in PHP** by adding below code snippet in your file.

```
header("HTTP/1.1 301 Moved Permanently");
header("Location: /option-a");
exit();
```

3. What is the difference between unset and unlink ?

Unlink: Is used to remove a file from server.
usage:unlink('path to file');

Unset: Is used unset a variable.
usage: unset($var);

4. What is Pear in PHP?

PEAR stand for Php Extension and Application Repository.PEAR provides:

- A structured library of code
- maintain a system for distributing code and for managing code packages
- promote a standard coding style
- provide reusable components.

5. Explain Type hinting in PHP ?

In PHP **Type hinting** is used to specify the excepted data type of functions argument.
Type hinting is introduced in PHP 5.

Example usage:-

//send Email function argument $email Type hinted of Email Class. It means to call this function you must have to pass an email object otherwise an error is generated.

```
<?php
function sendEmail (Email $email)
{
```

```
  $email->send();
}
?>
```

6. What is default session time and path in PHP. How to change it ?

Default session time in PHP is 1440 seconds (24 minutes) and the Default session storage path is temporary **folder/tmp** on server.

You can change default session time by using below code

```
<?php
// server should keep session data for AT LEAST 1 hour
ini_set('session.gc_maxlifetime', 3600);

// each client should remember their session id for EXACTLY 1 hour
session_set_cookie_params(3600);
?>
```

7. What is T_PAAMAYIM_NEKUDOTAYIM ?

T_PAAMAYIM_NEKUDOTAYIM is scope resolution operator used as :: (double colon) .Basically, it used to call static methods/variables of a Class.

Example usage:-

```
 $Cache::getConfig($key);
```

8. What is difference between Method overriding and overloading in PHP?

Overriding and Overloading both are oops concepts.

In Overriding, a method of the parent class is defined in the child or derived class with the same name and parameters. Overriding comes in inheritance.

An example of Overriding in PHP.

```
<?php
```

```
class A {
   function showName() {
      return "Ajay";
   }
}

class B extends A {
   function showName() {
      return "Anil";
   }
}

$foo = new A;
$bar = new B;
echo($foo->myFoo()); //"Ajay"
echo($bar->myFoo()); //"Anil"
?>
```

In Overloading, there are multiple methods with the same name with different signatures or parameters. Overloading is done within the same class. Overloading is also called early binding or compile time polymorphism or static binding.

An example of Overloading in PHP

```
<?php
class A{

function sum($a,$b){
            return $a+$b;

}

function sum($a,$b,$c){
            return $a+$b+$c;
```

```
}

}

$obj= new A;
$obj->sum(1,3); // 4
$obj->sum(1,3,5); // 9
?>
```

9. What is cURL in PHP ?

cURL is a library in PHP that allows you to make HTTP requests from the server.

10. What is a composer in PHP?

It is an application-level package manager for PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

11. How to remove duplicate values from a PHP Array?

You can use library function **array_unique()** for removing duplicated values for an array. Here is syntax to use it.

```
<?php
        $a=array("a"=>"home","b"=>"town","c"=>"town","php");
        print_r(array_unique($a));
?>
```

12. How to check curl is enabled or not in PHP

use **function_exists('curl_version')** function to check curl is enabled or not. This function returns true if curl is enabled other false

Example :

```
if(function_exists('curl_version') ){
  echo "Curl is enabled";
}else{

echo "Curl is not enabled";

}
```

13. How to create a public static method in PHP?

Static **method** is a member of class that is called directly by class name without creating an instance.In PHP uou can create a static method by using static keyword.

Example:

```
class A {
    public static function sayHello() {
        echo 'hello Static';
    }
}

A::sayHello();
```

14. How can you get web browser's details using PHP?

get_browser() function is used to retrieve the client browser details in PHP. This is a library function is PHP which looks up the browscap.ini file of the user and returns the capabilities of its browser.

Syntax:

get_browser(user_agent,return_array)

Example Usage:

```
$browserInfo = get_browser(null, true);
print_r($browserInfo);
```

15. What is the difference between an interface and an abstract class?

16. What is Type hinting in PHP ?

17. Difference between array_combine and array_merge?

array_combine is used to combine two or more arrays while array_merge is used to append one array at the end of another array.

array_combine is used to create a new array having keys of one array and values of another array that are combined with each other whereas array_merge is used to create a new array in such a way that values of the second array append at the end of the first array.

array_combine doesn't override the values of the first array but
in **array_merge**values of the first array overrides with the values of the second one.

Example of array_combine

```
<?php
$arr1    = array("sub1","sub2","sub3");
$arr2    = array(("php","html","css");
$new_arr = array_combine($arr1, $arr2);
print_r($new_arr);
?>
```

OUTPUT:

```
Array([sub1] => php [sub2] => html [sub3 =>css)
```

Example of array_merge

```
<?php
$arr1 = array("sub1" => "node", "sub2" => "sql");
$arr2 = array("s1"=>"jQuery", "s3"=>"xml", "sub4"=>"Css");
$result = array_merge($arr1, $arr2);
 print_r($result);
?>
```

OUTPUT:

```
Array ([s1] => jquery [sub2] => sql [s3] => xml [sub4] =>Css )
```

18. What is php.ini & .htacess file?

Both are used to make the changes to your PHP setting. These are explained below:

php.ini: It is a special file in PHP where you make changes to your PHP settings. It works when you run PHP as CGI. It depends on you whether you want to use the default settings or changes the setting by editing a php.ini file or, making a new text file and save it as php.ini.

.htaccess: It is a special file that you can use to manage/change the behavior of your site. It works when PHP is installed as an Apache module. These changes include such as redirecting your domain's page to https or www, directing all users to one page, etc.

19. How to terminate the execution of a script in PHP ?

To terminate the script in PHP, **exit()** function is used. It is an inbuilt function which outputs a message and then terminates the current script. The message which is you want to display is passed as a parameter to the exit () function and this function terminates the script after displaying the message. It is an alias function of die () function. It doesn't return any value.

Syntax: exit(message)

Where massage is a parameter to be passed as an argument. It defines message or status.

Exceptions of exit():

- If no status is passed to exit(), then it can be called without parenthesis.

- If a passed status is an integer then the value will not be printed but used as the exit status.
- The range of exit status is from 0 to 254 where 255 is reserved in PHP.

Errors And Exceptions

- exit() can be called without parentheses if no status is passed to it. Exit() function is also known by the term language construct.
- If the status passed to an exit() is an integer as a parameter, that value will be used as the exit status and not be displayed.
- The range of exit status should be in between 0 to 25. the exit status 255 should not be used because it is reserved by PHP.

20. What is difference between md5 and SHA256?

Both MD5 and SHA256 are used as hashing algorithms. They take an input file and generate an output which can be of 256/128-bit size. This output represents a checksum or hash value. As, collisions are very rare between hash values, so no encryption takes place.

- The difference between MD5 and SHA256 is that the former takes less time to calculate than later one.
- SHA256 is difficult to handle than MD5 because of its size.
- SHA256 is less secure than MD5
- MD5 result in an output of 128 bits whereas SHA256 result output of 256 bits.

Concluding all points, it will be better to use MDA5 if you want to secure your files otherwise you can use SHA256.

21. What is the difference between nowdoc and heredoc?

Heredoc and **nowdoc** are the methods to define the string in PHP in different ways.

- Heredoc process the $variable and special character while nowdoc doesn't do the same.
- Heredoc string uses double quotes "" while nowdoc string uses single quote ''
- Parsing is done in heredoc but not in nowdoc.

22. What is a path Traversal ?

Path Traversal also is known as **Directory Traversal** is referring to an attack which is attacked by an attacker to read into the files of the web application. Also, he/she can reveal the content of the file outside the root directory of a web server or any application. Path traversal operates the web application file with the use of dot-dot-slash (../) sequences, as ../ is a cross-platform symbol to go up in the directory.

Path traversal basically implements by an attacker when he/she wants to gain secret passwords, access token or other information stored in the files. Path traversal attack allows the attacker to exploit vulnerabilities present in web file.

23. Write logic to print Floyd's triangle in PHP?

Floyd's triangle is the right-angled triangle which starts with 1 and filled its rows with a consecutive number. The count of elements in next row will increment by one and the first row contains only one element.

Example of Floyd's triangle having 4 rows

The logic to print Floyd's triangle

```
<?php


echo "print Floyd's triangle";
echo "<pre>


 $key = 1;
for ($i = 1; $i <= 4; $i++) {
  for ($j = 1; $j <= $i; $j++) {
    echo $key;
    $key++;
   if ($j == $i) {
     echo "<br/>";
   }
 }
}
echo "";
?>
```

Output:

1

2 3

4 5 6

7 8 9 10

24. What is Cross-site scripting?

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications. XSS enables attackers to inject client-side script into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

25. What are the encryption functions available in PHP ?

crypt(), Mcrypt(), hash() are used for encryption in PHP

26. What is purpose of @ in Php ?

In PHP @ is used to suppress error messages.When we add @ before any statement in php then if any runtime error will occur on that line, then the error handled by PHP

27. What is difference between strstr() and stristr() in PHP?

28. What is list in PHP?

List is similar to an array but it is not a function, instead it is a language construct. This is used for assignment of a list of variables in one operation. If you are using PHP 5 version, then the list values start from a rightmost parameter, and if you are using PHP 7 version, then your list starts with a left-most parameter. Code is like:

```
<?php
```

```
$info = array('red', 'sign', 'danger');
// Listing all the variables
list($color, $symbol, $fear) = $info;
echo "$color is $symbol of $fear";?php>
```

29. What is difference Between PHP 5 and 7?

There are many differences between PHP 5 and 7. Some of the main points are:

- **Performance**: it is obvious that later versions are always better than the previous versions if they are stable. So, if you execute code in both versions, you will find the performance of PHP 7 is better than PHP5. This is because PHP 5 use Zend II and PHP & uses the latest model PHP-NG or Next Generation.
- **Return Type**: In PHP 5, the programmer is not allowed to define the type of return value of a function which is the major drawback. But in PHP 7, this limitation is overcome and a programmer is allowed to define the return type of any function.
- **Error handling**: In PHP 5, there is high difficulty to manage fatal errors but in PHP 7, some major errors are replaced by exceptions which can be managed effortlessly. In PHP 7, the new engine Exception Objects has been introduced.
- **64-bit support**: PHP 5 doesn't support 64-bit integer while PHP 7 supports native 64-bit integers as well as large files.
- **Anonymous Class**: Anonymous class is not present n PHP 5 but present in PHP 7. It is basically used when you need to execute the class only once to increase the execution time.
- **New Operators**: Some new operators have added in PHP 7 such as <=> which is called a three-way comparison operator. Another operator has added is a null coalescing operator which symbol as?? and use to find whether something exists or not.
- **Group Use declaration**: PHP 7 includes this feature whereas PHP 5 doesn't have this feature.

30. What is difference between ksort() and usort() functions.

- ksort() function is used to sort an array according to its key values whereas asort() function is used to sort an array according to its values.
- They both used to sort an associative array in PHP.

Example of asort():

```
<?php
$age = array("Peter"=>"37", "Ben"=>"27", "Joe"=>"43");
asort($age);
?>
```

Output: Key=Ben, Value=37 Key=Joe, Value=43 Key=Peter, Value=35

Example of ksort():

```
<?php
$age = array("Peter"=>"37", "Ben"=>"27", "Joe"=>"43");
ksort($age);
?>
```

Output: Key=Ben, Value=37

Key=Joe, Value=43

Key=Peter, Value=35

31. What should be the length of variable for SHA256?

It is clear from the name SHA256 that the length is of 256 bits long. If you are using hexadecimal representation, then you require 64 digits to replace 256 bits, as one digit represents four bits. Or if you are using a binary representation which means one byte equals to eight bits, then you need 32 digits.

32. What is MIME?

MIME stands for Multipurpose Internet Mail Extensions is an extension of the email protocol. It supports exchanging of various data files such as audio, video, application programs, and many others on the internet. It can also handle ASCII texts and Simple Mail Transport Protocol on the internet.

33. What are access specifiers?

An access specifier is a code element that is used to determine which part of the program is allowed to access a particular variable or other information. Different programming languages have different syntax to declare access specifiers. There are three types of access specifiers in PHP which are:

- Private: Members of a class are declared as private and they can be accessed only from that class.
- Protected: The class members declared as protected can be accessed from that class or from the inherited class.
- Public: The class members declared as public can be accessed from everywhere.

34. What is difference between explode() or split() in PHP?

The explode() and split() functions are used in PHP to split the strings. Both are defined here:

Split() is used to split a string into an array using a regular expression whereas **explode()** is used to split the string by string using the delimiter.

Example of split():

split(":", "this:is:a:split"); //returns an array that contains this, is, a, split.

Output: Array ([0] => this,[1] => is,[2] => a,[3] => split)

Example of explode():

explode ("take", "take a explode example "); //returns an array which have value "a explode example"

Output: array([0] => "a explode example")

35. How can i execute PHP File using Command Line?

It is easy and simple to execute PHP scripts from the windows command prompt. You just follow the below steps:

1. Open the command prompt. Click on Start button->Command Prompt.

2. In the Command Prompt, write the full path to the PHP executable(php.exe) which is followed by the full path of a script that you want to execute. You must add space in between each component. It means to navigate the command to your script location.

For example,

let you have placed PHP in C:\PHP, and your script is in C:\PHP\sample-php-script.php,

then your command line will be:

```
C:\PHP\php.exe C:\PHP\sample-php-script.php
```

3. Press the enter key and your script will execute.

36. How to block direct directory access in PHP?

You can use a .htaccess file to block the direct access of directory in PHP. It would be best if you add all the files in one directory, to which you want to deny access.

For Apache, you can use this code:

```
<&lt  Order deny, allow  Deny from all</&lt
```

But first, you have to create a .htaccess file, if it is not present. Create the .htaccess file in the root of your server and then apply the above rule.

37. Explain preg_Match and preg_replace?

These are the commonly used regular expressions in PHP. These are an inbuilt function that is used to work with other regular functions.

preg-Match: This is the function used to match a pattern in a defined string. If the patterns match with string, it returns true otherwise it returns false.

Preg_replace: This function is used to perform a replace operation. In this, it first matches the pattern in the string and if pattern matched, ten replace that match with the specified pattern.

38. Explain mail function in PHP with syntax?

The mail function is used in PHP to send emails directly from script or website. It takes five parameters as an argument.

Syntax **of mail ()**: mail (to, subject, message, headers, parameters);

- **to** refers to the receiver of the email
- **Subject** refers to the subject of an email
- **the message** defines the content to be sent where each line separated with /n and also one line can't exceed 70 characters.
- **Headers** refer to additional information like from, Cc, Bcc. These are optional.

- **Parameters** refer to an additional parameter to the send mail program. It is also optional

39. List few sensible functions in PHP?

40. How can you get the size of an image in PHP?

41. How can we convert the time zones using PHP?

42. How to access a Static Member of a Class in PHP?

43. What are differences between PECL and PEAR?

44. What is difference between session and cookie in PHP ?

- Session and cookie both are used to store values or data.
- cookie stores data in your browser and a session is stored on the server.
- Session destroys that when browser close and cookie delete when set time expires.

45. How to get length of an array in PHP ?

PHP count function is used to get the length or numbers of elements in an array

```
<?php
// initializing an array in PHP
$array=['a','b','c'];
// Outputs 3
echo count($array);
?>
```

46. Which Scripting Engine PHP uses?

Zend Engine is used by PHP. The current stable version of Zend Engine is 3.0. It is developed by Andi Gutmans and Zeev Suraski at Technion – Israel Institute of Technology.

47. What is the difference between runtime exception and compile time exception?

An exception that occurs at compile time is called a checked exception. This exception cannot be ignored and must be handled carefully. For example, in Java if you use **FileReader** class to read data from the file and the file specified in class constructor does not exist, then a **FileNotFoundException** occurs and you will have to manage that exception. For the purpose, you will have to write the code in a try-catch block and handle the exception. On the other hand, an exception that occurs at runtime is called unchecked-exception Note: Checked exception is not handled so it becomes an unchecked exception. This exception occurs at the time of execution.

48. Code to open file download dialog in PHP ?

You can open a file download dialog in PHP by setting Content-Disposition in the header.

Here is a usage sample:-

```
// outputting a PDF file
header('Content-type: application/pdf');
// It will be called downloaded.pdf
header('Content-Disposition: attachment; filename="downloaded.pdf"');
```

```
// The PDF source is in original.pdf
readfile('original.pdf');
```

49. Explain Traits in PHP ?

50. How to get the IP address of the client/user in PHP?

You can use **$_SERVER['REMOTE_ADDR']** to get IP address of user/client in PHP, But sometime it may not return the true IP address of the client at all time. Use Below code to get true IP address of user.

```
function getTrueIpAddr(){
 if (!empty($_SERVER['HTTP_CLIENT_IP']))   //check ip from share internet
   {
     $ip=$_SERVER['HTTP_CLIENT_IP'];
   }
elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))   //to check ip is pass from proxy
 {
     $ip=$_SERVER['HTTP_X_FORWARDED_FOR'];
 }
   else
 {
     $ip=$_SERVER['REMOTE_ADDR'];
 }
   return $ip;
}
```

51. How will you calculate days between two dates in PHP?

Calculating days between two dates in PHP

```
<?Php
$date1 = date('Y-m-d');
```

```
$date2 = '2015-10-2';
$days = (strtotime($date1)-strtotime($date2))/(60*60*24);
echo $days;
?>
```

52. How to get no of arguments passed to a PHP Function?

func_get_args() function is used to get number of arguments passed in a PHP function.

Sample Usage:

```
function foo() {
   return  func_get_args();
}
echo foo(1,5,7,3);//output 4;
echo foo(a,b);//output 2;
echo foo();//output 0;
```

53. How to Pass JSON Data in a URL using CURL in PHP ?

Code to post JSON Data in a URL using CURL in PHP

```
$url='https://www.onlineinterviewquestions.com/get_details';
$jsonData='{"name":"phpScots",
"email":"phpscots@onlineinterviewquestions.com"
,'age':36
}';
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 0);
curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonData);
curl_close($ch);
```

54. What is the difference between == and === operator in PHP ?

In PHP == is equal operator and returns TRUE if $a is equal to $b after type juggling and === is Identical operator and return TRUE if $a is equal to $b, and they are of the same data type.

Example Usages:

```
<?php
  $a=true ;
  $b=1;
  // Below condition returns true and prints a and b are equal
  if($a==$b){
   echo "a and b are equal";
  }else{
   echo "a and b are not equal";
  }
  //Below condition returns false and prints a and b are not equal because $a and $b are of
different data types.
  if($a===$b){
   echo "a and b are equal";
  }else{
   echo "a and b are not equal";
  }
?>
```

55. How to register a variable in PHP session ?

In PHP 5.3 or below we can register a variable session_register() function.It is deprecated now and we can set directly a value in $_SESSION Global.

Example usage:

```
<?php
  // Starting session
   session_start();
  // Use of session_register() is deprecated
   $username = "PhpScots";
```

```
    session_register("username");
   // Use of $_SESSION is preferred
    $_SESSION["username"] = "PhpScots";
?>
```

Also Read **Laravel interview questions**

56. What is session in PHP. How to remove data from a session?

As HTTP is a stateless protocol. To maintain states on the server and share data across multiple pages PHP session are used. PHP sessions are the simple way to store data for individual users/client against a unique session ID. Session IDs are normally sent to the browser via session cookies and the ID is used to retrieve existing session data, if session id is not present on server PHP creates a new session, and generate a new session ID.

Example Usage:-

```
<?php

// starting a session

session_start();

// Creating a session

$_SESSION['user_info'] = ['user_id' =>1,
'first_name' =>
'Ramesh', 'last_name' =>
'Kumar', 'status' =>
'active'];

// checking session

if (isset($_SESSION['user_info']))
```

```
        {
        echo "logged In";
        }

// un setting remove a value from session

unset($_SESSION['user_info']['first_name']);

// destroying complete session

session_destroy();


?>
```

57. Code to upload a file in PHP ?

//PHP code to process uploaded file and moving it on server

```
if($_FILES['photo']['name'])
{
        //if no errors...
        if(!$_FILES['file']['error'])
        {
                //now is the time to modify the future file name and validate the file
                $new_file_name = strtolower($_FILES['file']['tmp_name']); //rename file
                if($_FILES['file']['size'] > (1024000)) //can't be larger than 1 MB
                {
                        $valid_file = false;
                        $message = 'Oops!  Your file\'s size is to large.';
                }

                //if the file has passed the test
                if($valid_file)
```

```
                {
                        //move it to where we want it to be
                        move_uploaded_file($_FILES['file']['tmp_name'],
'uploads/'.$new_file_name);
                        $message = 'File uploaded successfully.';
                }
        }
        //if there is an error...
        else
        {
                //set that to be the returned message
                $message = 'Something got wrong while uploading file:
'.$_FILES['file']['error'];
        }
}
```

58. What is difference between strstr() and stristr() ?

In PHP both functions are used to find the first occurrence of substring in a string except
stristr() is case-insensitive and strstr is case-sensitive,if no match is found then FALSE will be returned.

Sample Usage:

```
<?php
$email = ‘abc@xyz.com’;
$hostname = strstr($email, ‘@’);
echo $hostname;
output: @xyz.com
stristr() does the same thing in Case-insensitive manner
?>
```

59. What are constructor and destructor in PHP ?

PHP constructor and a destructor are special type functions which are automatically called when a PHP class object is created and destroyed.

Generally Constructor are used to intializes the private variables for class and Destructors to free the resources created /used by class .

Here is sample class with constructor and destructer in PHP.

```
<?php
class Foo {

    private $name;
    private $link;

    public function __construct($name) {
        $this->name = $name;
    }

    public function setLink(Foo $link){
        $this->link = $link;
    }

    public function __destruct() {
        echo 'Destroying: ', $this->name, PHP_EOL;
    }
}
?>
```

60. What is PECL?

PECL is an online directory or repository for all known PHP extensions. It also provides hosting facilities for downloading and development of PHP extensions.

You can read More about PECL from https://pecl.php.net/

61. What is Gd PHP?

GD is an open source library for creating dynamic images.

- PHP uses GD library to create PNG, JPEG and GIF images.
- It is also used for creating charts and graphics on the fly.
- GD library requires an ANSI C compiler to run.

Sample code to generate an image in PHP

```
<?php
        header("Content-type: image/png");

        $string = $_GET['text'];
        $im = imagecreatefrompng("images/button1.png");
        $mongo = imagecolorallocate($im, 220, 210, 60);
        $px = (imagesx($im) - 7.5 * strlen($string)) / 2;
        imagestring($im, 3, $px, 9, $string, $mongo);
        imagepng($im);

        imagedestroy($im);
?>
```

62. Is multiple inheritance supported in PHP ?

NO, multiple inheritance is not supported by PHP

63. How is a constant defined in a PHP script?

Defining a Constant in PHP

```
define('CONSTANT_NAME',value);
```

64. What is the use of Mbstring?

Mbstring

Mbstring is an extension used in PHP to handle non-ASCII strings. Mbstring provides multibyte specific string functions that help us to deal with multibyte encodings in PHP. Multibyte character encoding schemes are used to express more than 256 characters in the regular byte-wise coding system. Mbstring is designed to handle Unicode-based encodings such as UTF-8 and UCS-2 and many single-byte encodings for convenience PHP Character Encoding Requirements.

You may also Like **Codeigniter interview questions**

Below are some features of mbstring

1. It handles the character encoding conversion between the possible encoding pairs.
2. Offers automatic encoding conversion between the possible encoding pairs.
3. Supports function overloading feature which enables to add multibyte awareness to regular string functions.
4. Provides multibyte specific string functions that properly detect the beginning or ending of a multibyte character. For example, mb_strlen() and mb_split()

65. How to get number of days between two given dates using PHP ?

```
<?php
$tomorrow = mktime(0, 0, 0, date(“m”) , date(“d”)+1, date(“Y”));
$lastmonth = mktime(0, 0, 0, date(“m”)-1, date(“d”), date(“Y”));
echo ($tomorrow-$lastmonth)/86400;
?>
```

66. What are the differences between GET and POST methods in form submitting, give the case where we can use get and we can use post methods?

In PHP, one can specify two different submission methods for a form. The method is specified inside a FORM element, using the METHOD attribute. The difference between METHOD=”GET” (the default) and METHOD=”POST” is primarily defined in terms of form data encoding. According to the technical HTML specifications, GET means that form data is to be encoded (by a browser) into a URL while POST means that the form data is to appear within the message body of the HTTP request.

	Get	Post
History:	Parameters remain in browser history because they are part of the URL	Parameters are not saved in browser history.
Bookmarked:	Can be bookmarked.	Can not be bookmarked.

	Get	Post
BACK button/re-submit behavior:	GET requests are re-executed but may not be re-submitted to the server if the HTML is stored in the browser cache.	The browser usually alerts the user that data will need to be re-submitted.
Encoding type (enctype attribute):	application/x-www-form-urlencoded	multipart/form-data or application/x-www-form-urlencoded Use multipart encoding for binary data.
Parameters:	can send but the parameter data is limited to what we can stuff into the request line (URL). Safest to use less than 2K of parameters, some servers handle up to 64K	Can send parameters, including uploading files, to the server.
Hacked:	Easier to hack for script kiddies	More difficult to hack
Restrictions on form data type:	Yes, only ASCII characters allowed.	No restrictions. Binary data is also allowed.
Security:	GET is less secure compared to POST because data sent is part of the URL. So it's saved in browser history and server logs in plaintext.	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs.
Restrictions on form data length:	Yes, since form data is in the URL and URL length is restricted. A safe URL length limit is often 2048 characters but varies by browser and web server.	No restrictions
Usability:	GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive information.
Visibility:	GET method is visible to everyone (it will be displayed in the browsers address bar) and has limits on the amount of information to send.	POST method variables are not displayed in the URL.
Cached:	Can be cached	Not Cached
Large variable values:	7607 characters maximum size.	8 Mb max size for the POST method.

67. What are PHP Magic Methods/Functions. List them.

In PHP all functions starting with __ names are magical functions/methods. Magical methods always lives in a PHP class.The definition of magical function are defined by programmer itself.

Here are list of magic functions available in PHP

__construct(), __destruct(), __call(), __callStatic(), __get(), __set(), __isset(), __unset(), __sleep(), __wakeup(), __toString(), __invoke(), __set_state(), __clone() and __debugInfo() .

68. Why should I store logs in a database rather than a file?

A database provides more flexibility and reliability than does logging to a file. It is easy to run queries on databases and generate statistics than it is for flat files. Writing to a file has more overhead and will cause your code to block or fail in the event that a file is unavailable. Inconsistencies caused by slow replication in AFS may also pose a problem to errors logged to files. If you have access to MySQL, use a database for logs, and when the database is unreachable, have your script automatically send an e-mail to the site administrator.

69. What are different types of Print Functions available in PHP?

PHP is a server side scripting language for creating dynamic web pages. There are so many functions available for displaying output in PHP. Here, I will explain some basic functions for displaying output in PHP. The basic functions for displaying output in PHP are as follows:

- print() Function
- echo() Function
- printf() Function
- sprintf() Function
- Var_dump() Function
- print_r() Function

70. How to access standard error stream in PHP?

You can access standard error stream in PHP by using following code snippet:

```
$stderr = fwrite("php://stderr");


$stderr = fopen("php://stderr", "w");


$stderr = STDERR;
```

71. Advantages and Application Areas of PHP

PHP (Hypertext Preprocessor) is an open source, a server-side scripting language that is used for the web applications development.

The web pages can be designed using HTML and the execution of the code is done on the user's browser.

And, with PHP server-side scripting language, the code is executed on the server before being executed on the web browser of the user.

PHP programming language is considered as a friendly language with abilities to easily connect with Oracle, MySQL, and many other databases.

Uses and Application Areas of PHP

PHP scripts are used on popular operating systems like Linux, UNIX, Solaris, Windows, MAC OS, Microsoft and on many other operating systems. It supports many web servers that include Apache and IIS.
The use of PHP affords web developers the freedom to choose the operating system and the web server.
The following main areas of web development use the PHP programming language.

- **Command line scripting** In this area of web development, with just the use of PHP parser, the PHP script is executed with no requirement of any server program or browser. This type of use of the PHP script is normally employed for simple and easy processing tasks.
- **Server-side scripting** Server-side scripting is the main area of operation of PHP scripts in PHP. The following are involved in Server-side scripting:
    - Web server – It is a program that executes the files from web pages, from user requests.
    - Web browser – It is an application that is used to display the content on WWW (World Wide Web).
    - PHP parser – It is a program that converts the human-readable code into a format that is easier to understand by the computer.
- **Desktop Application Development** PHP is also used to create the client-side application such as desktop applications. They are usually characterized by the GUI (Graphic User Interface). The client-side applications can be developed with knowledge of using the advanced features like PHP-GTK.

Advantages of PHP

Now, let's have a quick look at why PHP is used; that is the advantages of PHP.

- **Open Source**
  PHP is an open source software, which means that it is freely available for modifications and redistribution, unlike any other programming language.

There is also an active team of PHP developers who are ready to provide any kind of technical support when needed.

- **Cross Platform**
  The PHP programming language is easy to use and modify and is also highly compatible with the leading operating systems as well as the web browsers. And, that made the deployment of the applications across a wide range of operating systems and browsers much easier than before.
  PHP not only supports the platforms like Linux, Windows, Mac OS, Solaris but is also applied to web servers like Apache, IIS and many others.
- **Suits Web Development**
  PHP programming language perfectly suits the web development and can be directly embedded into the HTML code.

Also Read: PHP Interview Questions

72. What are the difference between echo and print?

Difference between echo and print in PHP

echo in PHP

- echo is language constructs that display strings.
- echo has a void return type.
- echo can take multiple parameters separated by comma.
- echo is slightly faster than print.

Print in PHP

- print is language constructs that display strings.
- print has a return value of 1 so it can be used in expressions.
- print cannot take multiple parameters.
- print is slower than echo.

73. What is Mcrypt used for?

74. What are different types of errors available in Php ?

There are 13 types of errors in PHP, We have listed all below

- **E_ERROR:** A fatal error that causes script termination.

- **E_WARNING:** Run-time warning that does not cause script termination.
- **E_PARSE:** Compile time parse error.
- **E_NOTICE:** Run time notice caused due to error in code.
- **E_CORE_ERROR**: Fatal errors that occur during PHP initial startup. (installation)
- **E_CORE_WARNING:** Warnings that occur during PHP initial startup.
- **E_COMPILE_ERROR:** Fatal compile-time errors indication problem with script.
- **E_USER_ERROR:** User-generated error message.
- **E_USER_WARNING:** User-generated warning message.
- **E_USER_NOTICE:** User-generated notice message.
- **E_STRICT:** Run-time notices.
- **E_RECOVERABLE_ERROR:** Catchable fatal error indicating a dangerous error
- **E_ALL:** Catches all errors and warnings.

75. How to increase the execution time of a PHP script ?

The default max execution time for PHP scripts is set to 30 seconds. If a php script runs longer than 30 seconds then PHP stops the script and reports an error.
You can increase the execution time by changing max_execution_time directive in your php.ini file or calling **ini_set('max_execution_time', 300);** //300 seconds = 5 minutes function at the top of your php script.

76. List data types in PHP ?

PHP supports 9 primitive types

4 scalar types:

- integer
- boolean
- float
- string

3 compound types:

- array
- object
- callable

And 2 special types:

- resource

- NULL

Read **Zend Framework interview questions**

77. What is difference between include,require,include_once and require_once() ?

Include :-Include is used to include files more than once in single PHP script.You can include a file as many times you want.

Syntax:- include("file_name.php");

Include Once:-Include once include a file only one time in php script.Second attempt to include is ignored.

Syntax:- include_once("file_name.php");

Require:-Require is also used to include files more than once in single PHP script.Require generates a Fatal error and halts the script execution,if file is not found on specified location or path.You can require a file as many time you want in a single script.

Syntax:- require("file_name.php");

Require Once :-Require once include a file only one time in php script.Second attempt to include is ignored. Require Once also generates a Fatal error and halts the script execution ,if file is not found on specified location or path.

Syntax:- require_once("file_name.php");

78. Where sessions stored in PHP ?

PHP sessions are stored on server generally in text files in a temp directory of server.
That file is not accessible from outside word. When we create a session PHP create a unique session id that is shared by client by creating cookie on clients browser.That session id is sent by client browser to server each time when a request is made and session is identified.
The default session name is "PHPSESSID".

79. What is PHP ?

PHP: Hypertext Preprocessor is open source server-side scripting language that is widely used for creation of dynamic web applications.It was developed by Rasmus Lerdorf also know as Father of PHP in 1994.

PHP is a loosely typed language , we didn't have to tell PHP which kind of Datatype a Variable is. PHP automatically converts the variable to the correct datatype , depending on its value.

Dear readers, these **Python Programming Language Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Python Programming Language**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer −

What is Python?

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Name some of the features of Python.

Following are some of the salient features of python −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

What is the purpose of PYTHONPATH environment variable?

PYTHONPATH - It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.

What is the purpose of PYTHONSTARTUP environment variable?

PYTHONSTARTUP - It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.

What is the purpose of PYTHONCASEOK environment variable?

PYTHONCASEOK − It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.

What is the purpose of PYTHONHOME environment variable?

PYTHONHOME − It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

Is python a case sensitive language?

Yes! Python is a case sensitive programming language.

What are the supported data types in Python?

Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary

What is the output of print str if str = 'Hello World!'?

It will print complete string. Output would be Hello World!.

What is the output of print str[0] if str = 'Hello World!'?

It will print first character of the string. Output would be H.

What is the output of print str[2:5] if str = 'Hello World!'?

It will print characters starting from 3rd to 5th. Output would be llo.

What is the output of print str[2:] if str = 'Hello World!'?

It will print characters starting from 3rd character. Output would be llo World!.

What is the output of print str * 2 if str = 'Hello World!'?

It will print string two times. Output would be Hello World!Hello World!.

What is the output of print str + "TEST" if str = 'Hello World!'?

It will print concatenated string. Output would be Hello World!TEST.

What is the output of print list if list = ['abcd', 786 , 2.23, 'john', 70.2]?

It will print complete list. Output would be ['abcd', 786, 2.23, 'john', 70.200000000000003].

What is the output of print list[0] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

It will print first element of the list. Output would be abcd.

What is the output of print list[1:3] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

It will print elements starting from 2nd till 3rd. Output would be [786, 2.23].

What is the output of print list[2:] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

It will print elements starting from 3rd element. Output would be [2.23, 'john', 70.200000000000003].

What is the output of print tinylist * 2 if tinylist = [123, 'john']?

It will print list two times. Output would be [123, 'john', 123, 'john'].

What is the output of print list1 + list2, if list1 = ['abcd', 786 , 2.23, 'john', 70.2] and ist2 = [123, 'john']?

It will print concatenated lists. Output would be ['abcd', 786, 2.23, 'john', 70.2, 123, 'john']

What are tuples in Python?

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

What is the difference between tuples and lists in Python?

The main differences between lists and tuples are − Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists.

What is the output of print tuple if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?

It will print complete tuple. Output would be ('abcd', 786, 2.23, 'john', 70.200000000000003).

What is the output of print tuple[0] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?

It will print first element of the tuple. Output would be abcd.

What is the output of print tuple[1:3] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?

It will print elements starting from 2nd till 3rd. Output would be (786, 2.23).

What is the output of print tuple[2:] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?

It will print elements starting from 3rd element. Output would be (2.23, 'john', 70.200000000000003).

What is the output of print tinytuple * 2 if tinytuple = (123, 'john')?

It will print tuple two times. Output would be (123, 'john', 123, 'john').

What is the output of print tuple + tinytuple if tuple = ('abcd', 786 , 2.23, 'john', 70.2) and tinytuple = (123, 'john')?

It will print concatenated tuples. Output would be ('abcd', 786, 2.23, 'john', 70.200000000000003, 123, 'john').

What are Python's dictionaries?

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

How will you create a dictionary in python?

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

```
dict = {}
dict['one'] = "This is one"
dict[2]     = "This is two"
tinydict = {'name': 'john','code':6734, 'dept': 'sales'}
```

How will you get all the keys from the dictionary?

Using dictionary.keys() function, we can get all the keys from the dictionary object.

```
print dict.keys()    # Prints all the keys
```

How will you get all the values from the dictionary?

Using dictionary.values() function, we can get all the values from the dictionary object.

```
print dict.values()    # Prints all the values
```

How will you convert a string to an int in python?

int(x [,base]) - Converts x to an integer. base specifies the base if x is a string.

How will you convert a string to a long in python?

long(x [,base]) - Converts x to a long integer. base specifies the base if x is a string.

How will you convert a string to a float in python?

float(x) − Converts x to a floating-point number.

How will you convert a object to a string in python?

str(x) − Converts object x to a string representation.

How will you convert a object to a regular expression in python?

repr(x) − Converts object x to an expression string.

How will you convert a String to an object in python?

eval(str) − Evaluates a string and returns an object.

How will you convert a string to a tuple in python?

tuple(s) − Converts s to a tuple.

How will you convert a string to a list in python?

list(s) − Converts s to a list.

How will you convert a string to a set in python?

set(s) − Converts s to a set.

How will you create a dictionary using tuples in python?

dict(d) − Creates a dictionary. d must be a sequence of (key,value) tuples.

How will you convert a string to a frozen set in python?

frozenset(s) − Converts s to a frozen set.

How will you convert an integer to a character in python?

chr(x) − Converts an integer to a character.

How will you convert an integer to an unicode character in python?

unichr(x) − Converts an integer to a Unicode character.

How will you convert a single character to its integer value in python?

ord(x) − Converts a single character to its integer value.

How will you convert an integer to hexadecimal string in python?

hex(x) − Converts an integer to a hexadecimal string.

How will you convert an integer to octal string in python?

oct(x) − Converts an integer to an octal string.

What is the purpose of ** operator?

** Exponent − Performs exponential (power) calculation on operators. a**b = 10 to the power 20 if a = 10 and b = 20.

What is the purpose of // operator?

// Floor Division − The division of operands where the result is the quotient in which the digits after the decimal point are removed.

What is the purpose of is operator?

is − Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. x is y, here is results in 1 if id(x) equals id(y).

What is the purpose of not in operator?

not in − Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. x not in y, here not in results in a 1 if x is not a member of sequence y.

What is the purpose break statement in python?

break statement − Terminates the loop statement and transfers execution to the statement immediately following the loop.

What is the purpose continue statement in python?

continue statement − Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

What is the purpose pass statement in python?

pass statement − The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

How can you pick a random item from a list or tuple?

choice(seq) − Returns a random item from a list, tuple, or string.

How can you pick a random item from a range?

randrange ([start,] stop [,step]) − returns a randomly selected element from range(start, stop, step).

How can you get a random number in python?

random() − returns a random float r, such that 0 is less than or equal to r and r is less than 1.

How will you set the starting value in generating random numbers?

seed([x]) − Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None.

How will you randomizes the items of a list in place?

shuffle(lst) − Randomizes the items of a list in place. Returns None.

How will you capitalizes first letter of string?

capitalize() − Capitalizes first letter of string.

How will you check in a string that all characters are alphanumeric?

isalnum() − Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.

How will you check in a string that all characters are digits?

isdigit() − Returns true if string contains only digits and false otherwise.

How will you check in a string that all characters are in lowercase?

islower() − Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.

How will you check in a string that all characters are numerics?

isnumeric() − Returns true if a unicode string contains only numeric characters and false otherwise.

How will you check in a string that all characters are whitespaces?

isspace() − Returns true if string contains only whitespace characters and false otherwise.

How will you check in a string that it is properly titlecased?

istitle() − Returns true if string is properly "titlecased" and false otherwise.

How will you check in a string that all characters are in uppercase?

isupper() − Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.

How will you merge elements in a sequence?

join(seq) − Merges (concatenates) the string representations of elements in sequence seq into a string, with separator string.

How will you get the length of the string?

len(string) − Returns the length of the string.

How will you get a space-padded string with the original string left-justified to a total of width columns?

ljust(width[, fillchar]) − Returns a space-padded string with the original string left-justified to a total of width columns.

How will you convert a string to all lowercase?

lower() − Converts all uppercase letters in string to lowercase.

How will you remove all leading whitespace in string?

lstrip() − Removes all leading whitespace in string.

How will you get the max alphabetical character from the string?

max(str) − Returns the max alphabetical character from the string str.

How will you get the min alphabetical character from the string?

min(str) − Returns the min alphabetical character from the string str.

How will you replaces all occurrences of old substring in string with new string?

replace(old, new [, max]) − Replaces all occurrences of old in string with new or at most max occurrences if max given.

How will you remove all leading and trailing whitespace in string?

strip([chars]) − Performs both lstrip() and rstrip() on string.

How will you change case for all letters in string?

swapcase() − Inverts case for all letters in string.

How will you get titlecased version of string?

title() − Returns "titlecased" version of string, that is, all words begin with uppercase and the rest are lowercase.

How will you convert a string to all uppercase?

upper() − Converts all lowercase letters in string to uppercase.

How will you check in a string that all characters are decimal?

isdecimal() − Returns true if a unicode string contains only decimal characters and false otherwise.

What is the difference between del() and remove() methods of list?

To remove a list element, you can use either the del statement if you know exactly which element(s) you are deleting or the remove() method if you do not know.

What is the output of len([1, 2, 3])?

3.

What is the output of [1, 2, 3] + [4, 5, 6]?

[1, 2, 3, 4, 5, 6]

What is the output of ['Hi!'] * 4?

['Hi!', 'Hi!', 'Hi!', 'Hi!']

What is the output of 3 in [1, 2, 3]?

True

What is the output of for x in [1, 2, 3]: print x?

```
1
2
3
```

What is the output of L[2] if L = [1,2,3]?

3, Offsets start at zero.

What is the output of L[-2] if L = [1,2,3]?

1, Negative: count from the right.

What is the output of L[1:] if L = [1,2,3]?

2, 3, Slicing fetches sections.

How will you compare two lists?

cmp(list1, list2) − Compares elements of both lists.

How will you get the length of a list?

len(list) − Gives the total length of the list.

How will you get the max valued item of a list?

max(list) − Returns item from the list with max value.

How will you get the min valued item of a list?

min(list) − Returns item from the list with min value.

How will you get the index of an object in a list?

list.index(obj) − Returns the lowest index in list that obj appears.

How will you insert an object at given index in a list?

list.insert(index, obj) − Inserts object obj into list at offset index.

How will you remove last object from a list?

list.pop(obj=list[-1]) − Removes and returns last object or obj from list.

How will you remove an object from a list?

list.remove(obj) − Removes object obj from list.

How will you reverse a list?

list.reverse() − Reverses objects of list in place.

How will you sort a list?

list.sort([func]) − Sorts objects of list, use compare func if given.

What is lambda function in python?

'lambda' is a keyword in python which creates an anonymous function. Lambda does not contain block of statements. It does not contain return statements.

What we call a function which is incomplete version of a function?

Stub.

When a function is defined then the system stores parameters and local variables in an area of memory. What this memory is known as?

Stack.

A canvas can have a foreground color? (Yes/No)

Yes.

Is Python platform independent?

No

There are some modules and functions in python that can only run on certain platforms.

Do you think Python has a complier?

Yes

Yes it has a complier which works automatically so we don't notice the compiler of python.

What are the applications of Python?

Django (Web framework of Python).

2. Micro Frame work such as Flask and Bottle.

3. Plone and Django CMS for advanced content Management.

What is the basic difference between Python version 2 and Python version 3?

Table below explains the difference between Python version 2 and Python version 3.

S.No	Section	Python Version2	Python Version3
1.	Print Function	Print command can be used without parentheses.	Python 3 needs parentheses to print any string. It will raise error without parentheses.
2.	Unicode	ASCII str() types and separate Unicode() but there is no byte type code in Python 2.	Unicode (utf-8) and it has two byte classes − • Byte • Bytearray S.
3.	Exceptions	Python 2 accepts both new and old notations of syntax.	Python 3 raises a SyntaxError in turn when we don't enclose the exception argument in parentheses.
4.	Comparing Unorderable	It does not raise any error.	It raises 'TypeError' as warning if we try to compare unorderable types.

Which programming Language is an implementation of Python programming language designed to run on Java Platform?

Jython

(Jython is successor of Jpython.)

Is there any double data type in Python?

No

Is String in Python are immutable? (Yes/No)

Yes.

Can True = False be possible in Python?

No.

Which module of python is used to apply the methods related to OS.?

OS.

When does a new block begin in python?

A block begins when the line is intended by 4 spaces.

Write a function in python which detects whether the given two strings are anagrams or not.

```
def check(a,b):
   if(len(a)!=len(b)):
      return False
   else:
      if(sorted(list(a)) == sorted(list(b))):
         return True
   else:
      return False
```

Name the python Library used for Machine learning.

Scikit-learn python Library used for Machine learning

What does pass operation do?

Pass indicates that nothing is to be done i.e. it signifies a no operation.

Name the tools which python uses to find bugs (if any).

Pylint and pychecker.

Write a function to give the sum of all the numbers in list?

Sample list − (100, 200, 300, 400, 0, 500)

Expected output − 1500

Program for sum of all the numbers in list is −

```
def sum(numbers):
   total = 0
   for num in numbers:
      total+=num
   print(''Sum of the numbers: '', total)
sum((100, 200, 300, 400, 0, 500))
```

We define a function ‘sum’ with numbers as parameter. The in for loop we store the sum of all the values of list.

Write a program in Python to reverse a string without using inbuilt function reverse string?

Program to reverse a string in given below −

```
def string_reverse(str1):

rev_str = ' '
index = len(str1) #defining index as length of string.
while(index>0):
   rev_str = rev_str + str1[index-1]
   index = index-1
   return(rev_str)

print(string_reverse('1tniop'))
```

First we declare a variable to store the reverse string. Then using while loop and indexing of string (index is calculated by string length) we reverse the string. While loop starts when index is greater than zero. Index is reduced to value 1 each time. When index reaches zero we obtain the reverse of string.

Write a program to test whether the number is in the defined range or not?

Program is −

```
def test_range(num):
   if num in range(0, 101):
      print(''%s is in range''%str(num))
   else:
      print(''%s is not in range''%str(num))
```

Output −

test_range(101)

101 is not in the range

To test any number in a particular range we make use of the method 'if..in' and else condition.

Write a program to calculate number of upper case letters and number of lower case letters?

Test on String: "Tutorials POINT"

Program is −

```
def string_test(s):

a = { ''Lower_Case'':0 , ''Upper_Case'':0} #intiail count of
lower and upper
for ch in s: #for loop
   if(ch.islower()): #if-elif-else condition
      a[''Lower_Case''] = a[''Lower_Case''] + 1
   elif(ch.isupper()):
      a[''Upper_Case''] = a [''Upper_Case''] + 1
   else:
      pass
```

```
print(''String in testing is: '',s) #printing the statements.
print(''Number of Lower Case characters in String:
'',a[''Lower_Case''])
print(''Number of Upper Case characters in String:
'',a[''Upper_Case''])
```

Output −

string_test("Tutorials POINT")

String in testing is: Tutorials POINT

Number of Lower Case characters in String: 8

Number of Upper Case characters in String: 6

We make use of the methods .islower() and .isupper(). We initialise the count for lower and upper. Using if and else condition we calculate total number of lower and upper case characters.

What is Next?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

String Reversal & Palindrome

String reversal questions can provide some information as to how well, certain candidates have been with dealing with text in python and at handling basic operations.

Question 1:

Question: Reverse the String " "the fox jumps over the lazy dog"

Answer:

```
a = "the fox jumps over the lazy dog"

a[::-1]


or ''.join(x for x in reversed(a)) [less efficient]

or ''.join(a[-x] for x in range(1, len(a)+1)) [less efficient]
```

Assessment:

- This is more of a warmup question than anything else and while it is good to know the shortcut notation, especially as it denotes some knowledge of how python deals with strings (eg substr a[0:7] for the fox) it is not necessary for most data-science's purpose

Question 2:

Question: identity all words that are palindromes in the following sentence "Lol, this is a gag, I didn't laugh so much in a long time"

Answer:

```
def isPalindrome(word: str) -> bool:
    if(word == word[::-1]):
        return True
    return Falsedef getPalindromesFromStr(inputStr: str) -> list:
    cleanStr = inputStr.replace(",","").lower()
    words = set(cleanStr.split(" "))
    wPalindromes = [
```

```
    word for word in words
    if isPalindrome(word) and word != ""
  ]
  return wPalindromesgetPalindromesFromStr("Lol, this is a gag, I didn't laugh so much
in a long time")
```

Assessment:

- Does the candidate thinks about cleaning his/her inputs?
- Does the candidate know the basic or word processing in python such as replace / split / lower?
- Does the candidate know how to use list comprehension?
- How does the candidate structure his/her code?

FizzBuzz

FizzBuzz is a traditional programming screening question, that allows to test if a candidate can think through a problem that is not a simple if else statement. The approach that they take can also shed some light to their understanding of the language.

Question: Write a program that prints the number for 1 to 50, for number multiple of 2 print fizz instead of a number, for numbers multiple of 3 print buzz, for numbers which are multiple of both 2 and 3 fizzbuzz.

Answer:

```
def fizzbuzzfn(num) -> str:
  mod_2 = (num % 2 == 0)
  mod_3 = (num % 3 == 0)
  if (mod_2 or mod_3):
    return (mod_2 * 'Fizz') + (mod_3 * 'Buzz')
  return str(num)print('\n'.join([fizzbuzzfn(x) for x in range(1,51)]))
```

Assessment:

- Do they know the modulo operator and are able to apply it?
- Are they storing the result of the modulo operators in variables for re-use?
- Do they understand how True/False interact with a String?
- Are they bombarding their code with if statements?
- Do they return a consistent type or mix both integer and string?

First Duplicate word

First finding of duplicate word allows to identity if candidates know the basic of text processing in python as well as are able to handle some basic data structure.

Question 1

Question: Given a string find the first duplicate word, example string: “this is just a wonder, wonder why do I have this in mind”

Answer:

```
string = "this is just a wonder, wonder why do I have this in mind"def firstduplicate(string:
str) -> str:
    import re
    cleanStr = re.sub("[^a-zA-Z -]", "", string)


    words = cleanStr.lower().split(" ")
    seen_words = set()
    for word in words:
        if word in seen_words:
            return word
        else:
```

```
        seen_words.add(word)
    return Nonefirstduplicate(string)
```

Assessment:

- Do I have constraint I need to work with, for instance in terms of memory?
- Cleans the string from punctuation? Replace or Regexp? If use regexp replace, should I compile the regexp expression or used it directly?
- Knows the right data-structure to check for existence.
- Does it terminate the function as soon as the match is found or?

Question 2:

Question: What if we wanted to find the first word with more than 2 duplicates in a string?

Answer:

```
string = "this is just a wonder, wonder why do I have this in mind. This is all that
matters."def first2timesduplicate(string: str) -> str:
    import re
    cleanStr = re.sub("[^a-zA-Z -]", "", string)


    words = cleanStr.lower().split(" ")
    seen_words = dict()    for word in words:
        previous_count = seen_words.get(word, 0)
        seen_words[word] = previous_count + 1
        if previous_count >= 2:
            return word
    return Nonefirst2timesduplicate(string)
```

Assessment:

- Some small modification is needed to be able to accommodate that change, the main one is arising from the use of a dictionary data-structure rather than a set. Counters are also a valid data-structure for this use case.

- There is little difficulty on modifying the previous function to cope with this change request, it is worth checking that the candidate does instantiate the specific key correctly, taking into account default values.

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions

Quick Fire questions

Some quick fire questions can also be asked to test the general knowledge of the python language.

Question 1:

Question: Replicate the sum for any number of variables, eg sum(1,2,3,4,5..)

Answer

```
def sum(*args):
 val = 0
 for arg in args:
  val += arg
 return val
```

Assessment:

- Quick interview question to check the knowledge of variable arguments, and how to setup one of the most basic functions.

Question 2:

Questions around the Fibonacci series is a classic of programming interviews and candidates should in general be at least familiar with them. They allow to test recursive thinking.

Question: Fibonacci sequences are defined as follow:

F_0 = 0 ; F_1 = 1

F_n = F_{-1} + F_{-2}

Write a function that gives the sum of all fibonacci numbers from 0 to n.

Answer:

```
def fibonacci(n: int) -> int:
   # fib series don't exist < 0
```

```
# might want to throw an error or a null
# for that
if n <= 0:
    return 0
if n == 1:
    return 1
else:
    return fibonacci(n-1) + fibonacci(n-2)def naiveFibSum(n: int) -> int:
return sum([fibonacci(x) for x in range(0, n+1)])def sumFib(n: int) -> int:
return fibonacci(n + 2) -1
```

Assessment:

- First, is the candidate able to think recursively?
- Is the candidate only thinking about a naive solution to the sum of fibonacci series or is s/he understanding that it can also summarized to a in a more effective way?

Wrap up

These questions are just meant to be a first screener for data-scientist and should be combined with statistical and data manipulation types of questions. They are meant to give a quick glimpse on whether a candidate has the basic minimum knowledge to go through a full interview rounds.

More advanced programming questions for Python would tend to cover the use of generators, decorators, cython or the efficient use of libraries such as pandas/numpy.

AJ's Guide To

Algorithms and Data Structures

90+ Algo and DS Chapters
15+ HR questions