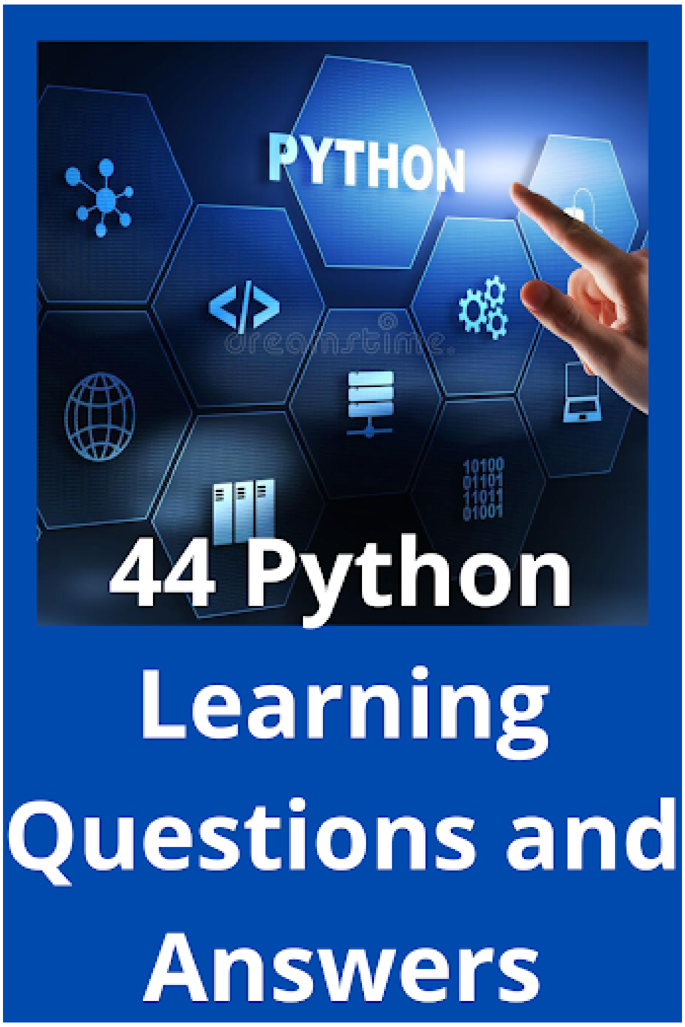
44 Python Learning Questions and Answers

itexamsusa.blogspot.com/2022/07/44-python-learning-questions-and-answers.html

1. What Is Python?



Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Finally, Python is portable: it runs on many Unix variants, on the Mac, and on PCs under MS-DOS, Windows, Windows NT, and OS/2.

2. Is There A Tool To Help Find Bugs Or Perform Static Analysis?

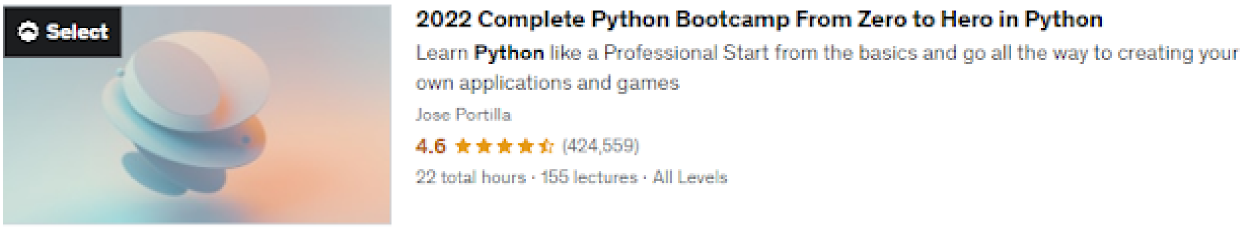
Yes.

PyChecker is a static analysis tool that finds bugs in Python source code and warns about code complexity and style.

Pylint is another tool that checks if a module satisfies a coding standard, and also makes it possible to write plug-ins to add a custom feature.

3. What Are The Rules For Local And Global Variables In Python?

In Python, variables that are only referenced inside a function are implicitly global. If a variable is assigned a new value anywhere within the function's body, it's assumed to be a local. If a variable is ever assigned a new value inside the function, the variable is implicitly local, and you need to explicitly declare it as 'global'. Though a bit surprising at first, a moment's consideration explains this. On one hand, requiring global for assigned variables provides a bar against unintended side-effects. On the other hand, if global was required for all global references, you'd be using global all the time. You'd have to declare as global every reference to a builtin function or to a component of an imported module. This clutter would defeat the usefulness of the global declaration for identifying side-effects.


Select

2022 Complete Python Bootcamp From Zero to Hero in Python

Learn **Python** like a Professional Start from the basics and go all the way to creating your own applications and games

Jose Portilla

4.6 ★★★★☆ (424,559)

22 total hours · 155 lectures · All Levels

4. How Do I Share Global Variables Across Modules?

The canonical way to share information across modules within a single program is to create a special module (often called config or cfg). Just import the config module in all modules of your application; the module then becomes available as a global name. Because there is only one instance of each module, any changes made to the module object get reflected everywhere. For example:

config.py:

```
x = 0 # Default value of the 'x' configuration setting
```

mod.py:

```
import config

config.x = 1
```

main.py:

```
import config

import mod

print config.x
```

5. How Do I Copy An Object In Python?

In general, try copy.copy() or copy.deepcopy() for the general case. Not all objects can be copied, but most can.

Some objects can be copied more easily. Dictionaries have a copy() method:

```
newdict = olddict.copy()
```

Sequences can be copied by slicing:

```
new_l = l[:]
```

6. How Can I Find The Methods Or Attributes Of An Object?

For an instance x of a user-defined class, dir(x) returns an alphabetized list of the names containing the instance attributes and methods and attributes defined by its class.

7. Is There An Equivalent Of C's "?:" Ternary Operator?

NO

8. How Do I Convert A Number To A String?

To convert, e.g., the number 144 to the string '144', use the built-in function str(). If you want a hexadecimal or octal representation, use the built-in functions hex() or oct(). For fancy formatting, use the % operator on strings, e.g. "%04d" % 144 yields '0144' and "%.3f" % (1/3.0) yields '0.333'.

9. What's A Negative Index?

Python sequences are indexed with positive numbers and negative numbers. For positive numbers 0 is the first index 1 is the second index and so forth. For negative indices -1 is the last index and -2 is the penultimate (next to last) index and so forth. Think of seq[-n] as the same as seq[len(seq)-n].

Using negative indices can be very convenient. For example S[:-1] is all of the string except for its last character, which is useful for removing the trailing newline from a string.

100 Days of Code: The Complete Python Pro Bootcamp for 2022

Master **Python** by building 100 projects in 100 days. Learn data science, automation, build websites, games and apps!

Dr. Angela Yu

4.7 ★★★★½ (121,677)

64 total hours · 676 lectures · All Levels

Bestseller

10. How Do I Apply A Method To A Sequence Of Objects?

Use a list comprehension:

```
result = [obj.method() for obj in List]
```

11. What Is A Class?

A class is the particular object type created by executing a class statement. Class objects are used as templates to create instance objects, which embody both the data (attributes) and code (methods) specific to a datatype.

A class can be based on one or more other classes, called its base class(es). It then inherits the attributes and methods of its base classes. This allows an object model to be successively refined by inheritance. You might have a generic Mailbox class that provides basic accessor methods for a mailbox, and sub-classes such as MboxMailbox, MaildirMailbox, OutlookMailbox that handle various specific mailbox formats.

12. What Is A Method?

A method is a function on some object x that you normally call as x.name(arguments...). Methods are defined as functions inside the class definition:

```
class C:

def meth (self, arg):

return arg*2 + self.attribute
```

13. What Is Self?

Self is merely a conventional name for the first argument of a method. A method defined as meth(self, a, b, c) should be called as x.meth(a, b, c) for some instance x of the class in which the definition occurs; the called method will think it is called as meth(x, a, b, c).

14. How Do I Call A Method Defined In A Base Class From A Derived Class That Overrides It?

If you're using new-style classes, use the built-in super() function:

```
class Derived(Base):

def meth (self):

super(Derived, self).meth()
```

If you're using classic classes: For a class definition such as class Derived(Base): ... you can call method meth() defined in Base (or one of Base's base classes) as Base.meth(self, arguments...). Here, Base.meth is an unbound method, so you need to provide the self argument.

15. How Do I Find The Current Module Name?

A module can find out its own module name by looking at the predefined global variable __name__. If this has the value '__main__', the program is running as a script. Many modules that are usually used by importing them also provide a command-line interface or a self-test, and only execute this code after checking __name__: def main():

```
print 'Running test...'

...

if __name__ == '__main__':

main()

__import__('x.y.z') returns
```

Try:

```
__import__('x.y.z').y.z
```

For more realistic situations, you may have to do something like

```
m = __import__(s)

for i in s.split(".")[1:]:

m = getattr(m, i)
```

16. Where Is The Math.py (socket.py, Regex.py, Etc.) Source File?

There are (at least) three kinds of modules in Python:

1. modules written in Python (.py);

2. modules written in C and dynamically loaded (.dll, .pyd, .so, .sl, etc);

3. modules written in C and linked with the interpreter; to get a list of these, type:

```
import sys

print sys.builtin_module_names
```

17. How Do I Delete A File?

Use os.remove(filename) or os.unlink(filename);

18. How Do I Copy A File?

The shutil module contains a copyfile() function.

PYTHON2.png (50.33KiB)

Get the course here

19. How Do I Run A Subprocess With Pipes Connected To Both Input And Output?

Use the popen2 module. For example:

```
import popen2

fromchild, tochild = popen2.popen2("command")

tochild.write("input\n")

tochild.flush()

output = fromchild.readline()
```

20. How Do I Avoid Blocking In The Connect() Method Of A Socket?

The select module is commonly used to help with asynchronous I/O on sockets.

21. Are There Any Interfaces To Database Packages In Python?

Yes.

Python 2.3 includes the bsddb package which provides an interface to the BerkeleyDB library. Interfaces to disk-based hashes such as DBM and GDBM are also included with standard Python.

22. How Do I Generate Random Numbers In Python?

The standard module random implements a random number generator. Usage is simple:

```
import random

random.random()
```

This returns a random floating point number in the range [0, 1).

Complete Python Developer in 2022: Zero to Mastery

How to become a **Python** 3 Developer and get hired! Build 12+ projects, learn Web Development, Machine Learning + more!

Andrei Neagoie, Zero To Mastery

4.6 ★★★★☆ (38,246)

31 total hours · 333 lectures · All Levels

23. Can I Create My Own Functions In C?

Yes, you can create built-in modules containing functions, variables, exceptions and even new types in C.

24. Can I Create My Own Functions In C++?

Yes, using the C compatibility features found in C++. Place extern "C" { ... } around the Python include files and put extern "C" before each function that is going to be called by the Python interpreter. Global or static C++ objects with constructors are probably not a good idea.

25. How Can I Execute Arbitrary Python Statements From C?

The highest-level function to do this is PyRun_SimpleString() which takes a single string argument to be executed in the context of the module __main__ and returns 0 for success and -1 when an exception occurred (including SyntaxError).

26. How Can I Evaluate An Arbitrary Python Expression From C?

Call the function PyRun_String() from the previous with the start symbol Py_eval_input; it parses an expression, evaluates it and returns its value.

27. How Do I Interface To C++ Objects From Python?

Depending on your requirements, there are many approaches. To do this manually, begin by reading the "Extending and Embedding" document. Realize that for the Python run-time system, there isn't a whole lot of difference between C and C++ -- so the strategy of building a new Python type around a C structure (pointer) type will also work for C++ objects.

28. How Do I Make Python Scripts Executable?

On Windows 2000, the standard Python installer already associates the .py extension with a file type (Python.File) and gives that file type an open command that runs the interpreter (D:Program FilesPythonpython.exe "%1" %*). This is enough to make scripts executable from the command prompt as 'foo.py'. If you'd rather be able to execute the script by simple typing 'foo' with no extension you need to add .py to the PATHEXT environment variable.

On Windows NT, the steps taken by the installer as described above allow you to run a script with 'foo.py', but a longtime bug in the NT command processor prevents you from redirecting the input or output of any script executed in this way. This is often important.

The incantation for making a Python script executable under WinNT is to give the file an extension of .cmd and add the following as the first line:

```
@setlocal enableextensions & python -x %~f0 %* & goto :EOF
```

29. How Do I Debug An Extension?

When using GDB with dynamically loaded extensions, you can't set a breakpoint in your extension until your extension is loaded.

In your .gdbinit file (or interactively), add the command: br _PyImport_LoadDynamicModule

Then, when you run GDB:

```
$ gdb /local/bin/python

gdb) run myscript.py

gdb) continue # repeat until your extension is loaded

gdb) finish # so that your extension is loaded

gdb) br myfunction.c:50

gdb) continue
```

30. Where Is Freeze For Windows?

"Freeze" is a program that allows you to ship a Python program as a single stand-alone executable file. It is not a compiler; your programs don't run any faster, but they are more easily distributable, at least to platforms with the same OS and CPU.

31. Is A *.pyd File The Same As A Dll?

Yes .

32. How Do I Emulate Os.kill() In Windows?

Use win32api:

```
def kill(pid):
```

```
"""kill function for Win32"""

import win32api

handle = win32api.OpenProcess(1, 0, pid)

return (0 != win32api.TerminateProcess(handle, 0))
```

33. Explain About The Programming Language Python?

Python is a very easy language and can be learnt very easily than other programming languages. It is a dynamic object oriented language which can be easily used for software development. It supports many other programming languages and has extensive library support for many other languages.

University of Michigan

Python for Everybody

Skills you'll gain: Application Programming Interfaces, Computationa...

☆ **4.8** (250.8k reviews)

Beginner · Specialization

34. Explain About The Use Of Python For Web Programming?

Python can be very well used for web programming and it also has some special features which make you to write the programming language very easily. Some of the features which it supports are Web frame works, Cgi scripts, Webservers, Content Management systems, Web services, Webclient programming, Webservices, etc. Many high end applications can be created with Python because of the flexibility it offers.

35. State Some Programming Language Features Of Python?

Python supports many features and is used for cutting edge technology. Some of them are

1) A huge pool of data types such as lists, numbers and dictionaries.

2) Supports notable features such as classes and multiple inheritance.

3) Code can be split into modules and packages which assists in flexibility.

4) It has good support for raising and catching which assists in error handling.

5) Incompatible mixing of functions, strings, and numbers triggers an error which also helps in good programming practices.

6) It has some advanced features such as generators and list comprehensions.

7) This programming language has automatic memory management system which helps in greater memory management.

PYTHON2.png (50.33KiB)

Get the course here

36. How Is Python Interpreted?

Python has an internal software mechanism which makes your programming easy. Program can run directly from the source code. Python translates the source code written by the programmer into intermediate language which is again translated it into the native language of computer. This makes it easy for a programmer to use python.

37. Does Python Support Object Oriented Scripting?

Python supports object oriented programming as well as procedure oriented programming. It has features which make you to use the program code for many functions other than Python. It has useful objects when it comes to data and functionality. It is very powerful in object and procedure oriented programming when compared to powerful languages like C or Java.

38. Describe About The Libraries Of Python?

Python library is very huge and has some extensive libraries. These libraries help you do various things involving CGI, documentation generation, web browsers, XML, HTML, cryptography, Tk, threading, web browsing, etc. Besides the standard libraries of python there are many other libraries such as Twisted, wx python, python imaging library, etc.

39. State And Explain About Strings?

Strings are almost used everywhere in python. When you use single and double quotes for a statement in python it preserves the white spaces as such. You can use double quotes and single quotes in triple quotes. There are many other strings such as raw strings, Unicode strings, once you have created a string in Python you can never change it again.

40. Explain About Classes In Strings?

Classes are the main feature of any object oriented programming. When you use a class it creates a new type. Creating class is the same as in other programming languages but the syntax differs. Here we create an object or instance of the class followed by parenthesis.

41. What Is Tuple?

Tuples are similar to lists. They cannot be modified once they are declared. They are similar to strings. When items are defined in parenthesis separated by commas then they are called as Tuples. Tuples are used in situations where the user cannot change the context or application; it puts a restriction on the user.

Google

Google IT Automation with Python

Skills you'll gain: Application Development, Cloud Computing, Cloud…

☆ **4.7** (30.9k reviews)

Beginner · Professional Certificate

42. Explain And Statement About List?

As the name specifies list holds a list of data items in an orderly manner. Sequence of data items can be present in a list. In python you have to specify a list of items with a comma and to make it understand that we are specifying a list we have to enclose the statement in square brackets. List can be altered at any time.

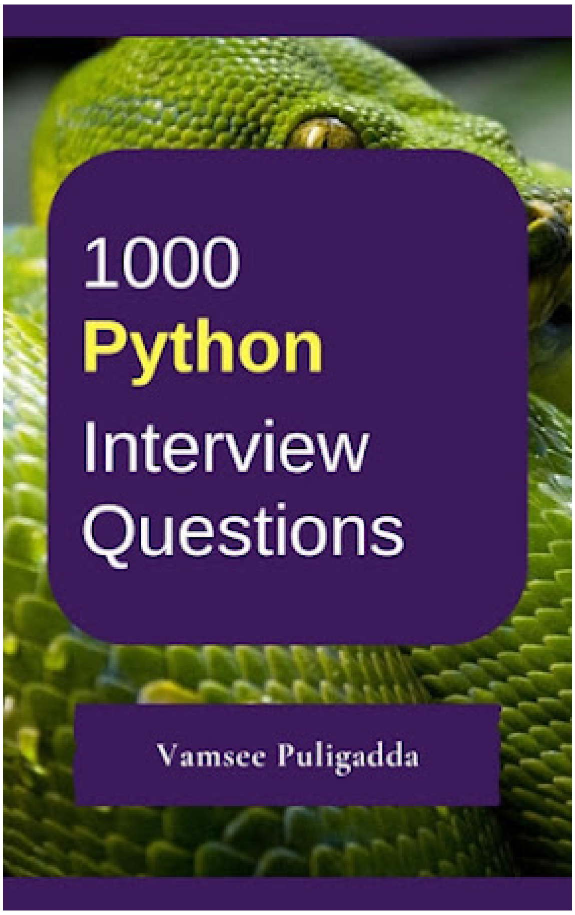
43. Explain About The Dictionary Function In Python?

A dictionary is a place where you will find and store information on address, contact details, etc. In python you need to associate keys with values. This key should be unique because it is useful for retrieving information. Also note that strings should be passed as keys in python. Notice that keys are to be separated by a colon and the pairs are separated themselves by commas. The whole statement is enclosed in curly brackets.

44. Explain About Indexing And Slicing Operation In Sequences?

Tuples, lists and strings are some examples about sequence. Python supports two main operations which are indexing and slicing. Indexing operation allows you to fetch a particular item in the sequence and slicing operation allows you to retrieve an item from the list of sequence. Python starts from the beginning and if successive numbers are not specified it starts at the last. In python the start position is included but it stops before the end statement.

Get the book here



==

Loading...