Top 20 python interview questions and answers

If you need top 7 free ebooks below for your job interview, please visit:
4career.net

- Free ebook: 75 interview questions and answers
- Top 12 secrets to win every job interviews
- 13 types of interview quesitons and how to face them
- Top 8 interview thank you letter samples
- Top 7 cover letter samples
- Top 8 resume samples
- Top 15 ways to search new jobs

Tell me about yourself?

This is probably the most asked question in python interview. It breaks the ice and gets you to talk about something you should be fairly comfortable with. Have something prepared that doesn't sound rehearsed. It's not about you telling your life story and quite frankly, the interviewer just isn't interested. Unless asked to do so, stick to your education, career and current situation. Work through it chronologically from the furthest back to the present.

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 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.

What Can You Do for Us That Other Candidates Can't?

What makes you unique? This will take an assessment of your experiences, skills and traits. Summarize concisely: "I have a unique combination of strong technical skills, and the ability to build strong customer relationships. This allows me to use my knowledge and break down information to be more user-friendly."

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.

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[:]

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'. See the library reference manual for details.

Is there a scanf() or sscanf() equivalent?

Not as such.
For simple input parsing, the easiest approach is usually to split the line into whitespace-delimited words using the split() method of string objects and then convert decimal strings to numeric values using int() or float(). split() supports an optional "sep" parameter which is useful if the line uses something other than whitespace as a separator.

For more complicated input parsing, regular expressions more powerful than C's sscanf() and better suited for the task.

How do I convert between tuples and lists?

The function tuple(seq) converts any sequence (actually, any iterable) into a tuple with the same items in the same order.

For example, tuple([1, 2, 3]) yields (1, 2, 3) and tuple('abc') yields ('a', 'b', 'c'). If the argument is a tuple, it does not make a copy but returns the same object, so it is cheap to call tuple() when you aren't sure that an object is already a tuple.

The function list(seq) converts any sequence or iterable into a list with the same items in the same order. For example, list((1, 2, 3)) yields [1, 2, 3] and list('abc') yields ['a', 'b', 'c']. If the argument is a list, it makes a copy just like seq[:] would.

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.

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 subclasses such as MboxMailbox, MaildirMailbox, OutlookMailbox that handle various specific mailbox formats.

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()
```

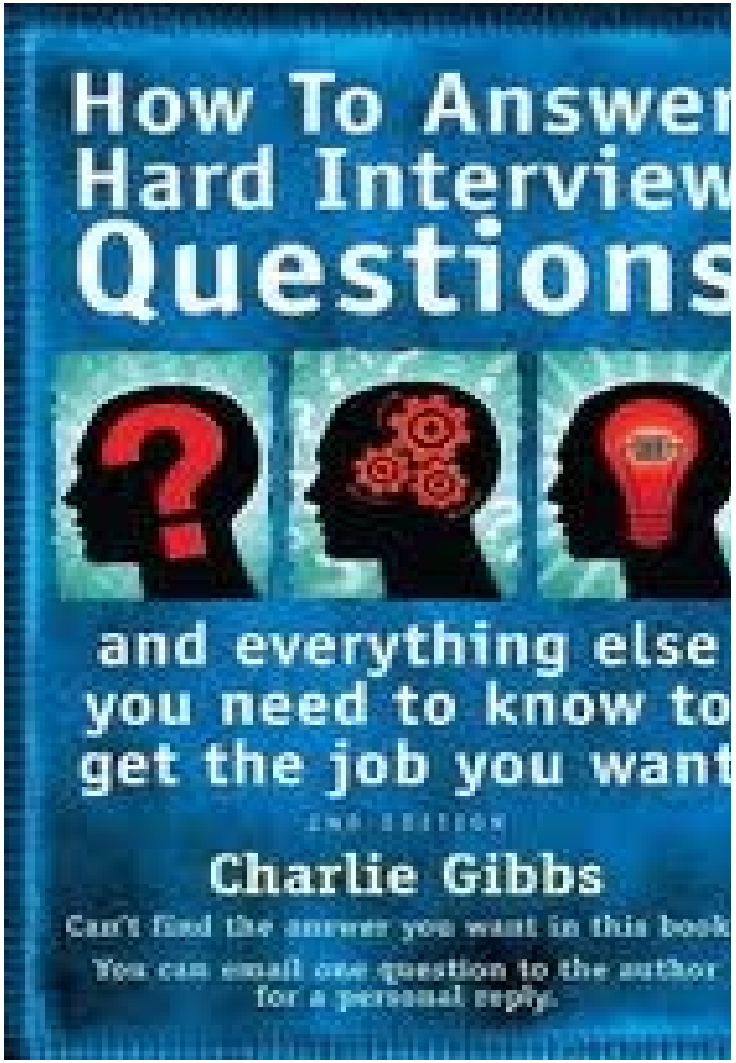
If you're using classic classes: For a class definition such as class Derived(Base): ... you can call method meth() defined in Base (or one of Base's base classes) as Base.meth(self, arguments...). Here, Base.meth is an unbound method, so you need to provide the self argument.

How can I organize my code to make it easier to change the base class?

You could define an alias for the base class, assign the real base class to it before your class definition, and use the alias throughout your class. Then all you have to change is the value assigned to the alias. Incidentally, this trick is also handy if you want to decide dynamically (e.g. depending on availability of resources) which base class to use. Example:

```
BaseAlias = <real base class>
class Derived(BaseAlias):
def meth(self):
BaseAlias.meth(self)
```

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
```

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).
opening position.

How do I apply a method to a sequence of objects?

Use a list comprehension:

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

More generically, you can try the following function:

```
def method_map(objects, method, arguments):
"""method_map([a,b], "meth", (1,2))
gives [a.meth(1,2), b.meth(1,2)]"""
nobjects = len(objects)
methods = map(getattr, objects, [method]*nobjects)
return map(apply, methods, [arguments]*nobjects)
```

Why don't my signal handlers work?

The most common problem is that the signal handler is declared with the wrong argument list. It is called as

handler(signum, frame)

so it should be declared with two arguments:

def handler(signum, frame):

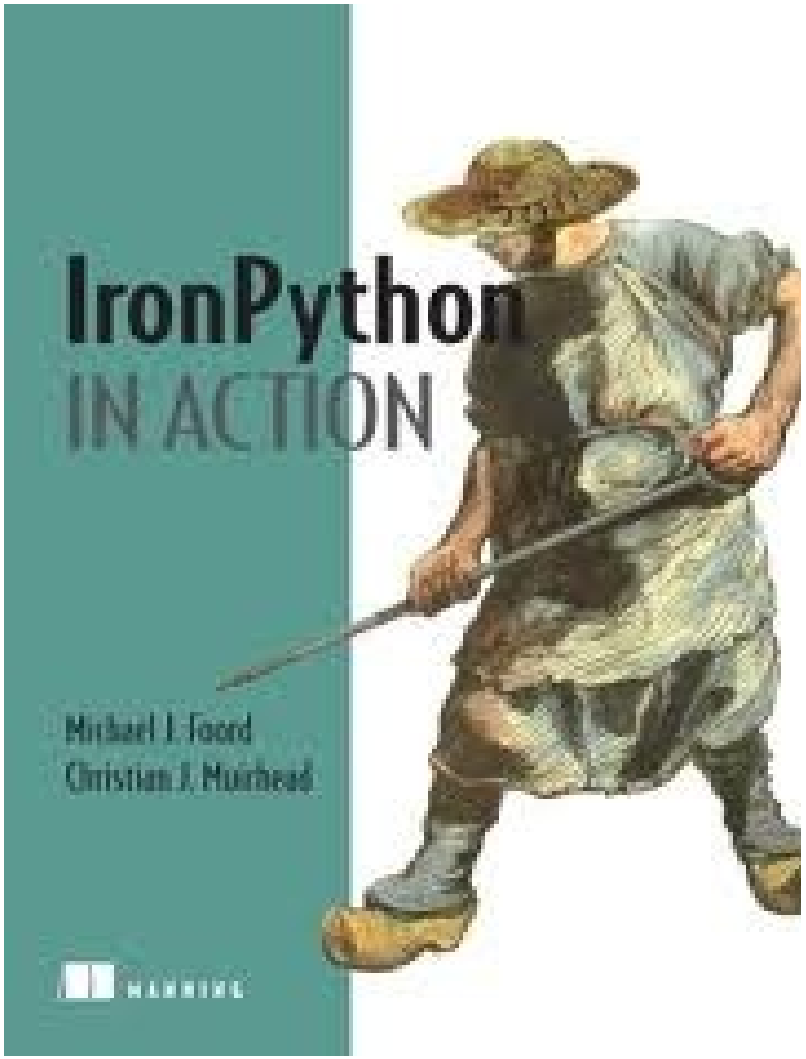
How can I execute arbitrary Python statements from C?

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). If you want more control, use PyRun_String(); see the source for PyRun_SimpleString() in Python/pythonrun.c.

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.

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

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).

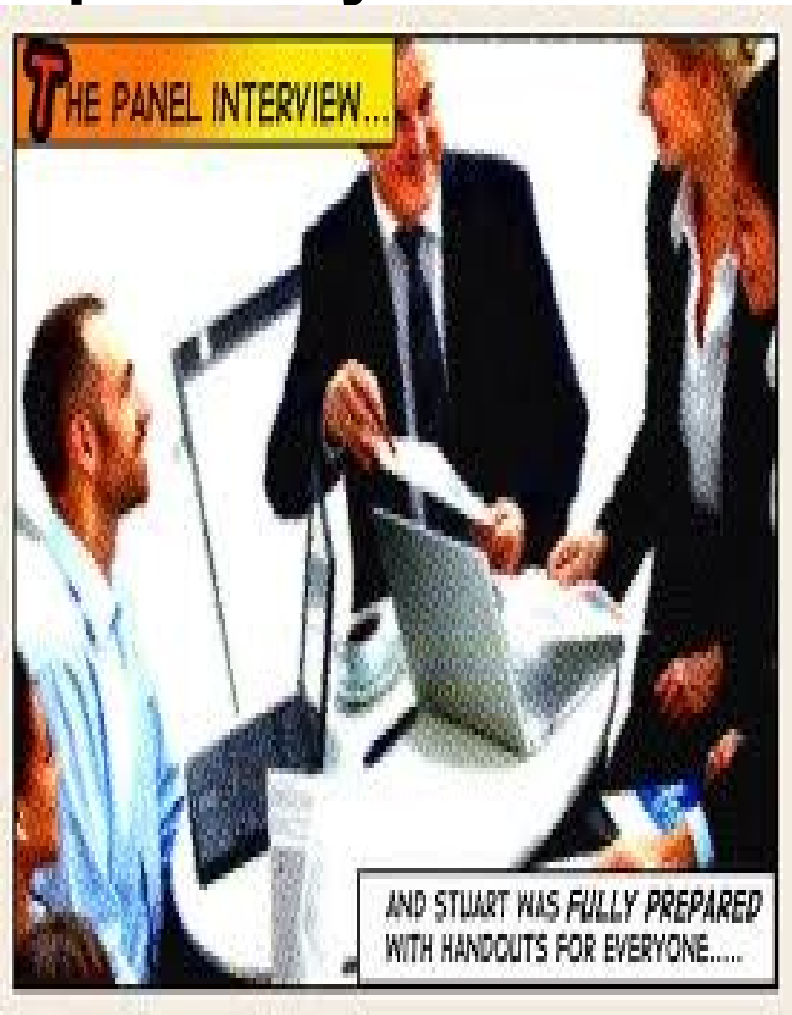
Useful job interview materials:

If you need top free ebooks below for your job interview, please visit:
4career.net

- Free ebook: 75 interview questions and answers
- Top 12 secrets to win every job interviews
- Top 36 situational interview questions
- 440 behavioral interview questions
- 95 management interview questions and answers
- 30 phone interview questions
- Top 8 interview thank you letter samples
- 290 competency based interview questions
- 45 internship interview questions
- Top 7 cover letter samples
- Top 8 resume samples
- Top 15 ways to search new jobs

Top 6 tips for job interview

Tip 1: Do your homework



You'll likely be asked difficult questions during the interview. Preparing the list of likely questions in advance will help you easily transition from question to question.

Spend time researching the company. Look at its site to understand its mission statement, product offerings, and management team. A few hours spent researching before your interview can impress the hiring manager greatly. Read the company's annual report (often posted on the site), review the employee's LinkedIn profiles, and search the company on Google News, to see if they've been mentioned in the media lately. The more you know about a company, the more you'll know how you'll fit in to it.

Ref material: 4career.net/job-interview-checklist-40-points

Tip 2: First impressions

When meeting someone for the first time, we instantaneously make our minds about various aspects of their personality.
Prepare and plan that first impression long before you walk in the door. Continue that excellent impression in the days following, and that job could be yours.
Therefore:

- Never arrive late.
- Use positive body language and turn on your charm right from the start.
- Switch off your mobile before you step into the room.
- Look fabulous; dress sharp and make sure you look your best.
- Start the interview with a handshake; give a nice firm press and then some up and down movement.
- Determine to establish a rapport with the interviewer right from the start.
- Always let the interviewer finish speaking before giving your response.
- Express yourself fluently with clarity and precision.

Useful material: 4career.net/top-10-elements-to-make-a-

Tip 3: The “Hidden” Job Market

Many of us don’t recognize that hidden job market is a huge one and accounts for 2/3 of total job demand from enterprises. This means that if you know how to exploit a hidden job market, you can increase your chance of getting the job up to 300%.

In this section, the author shares his experience and useful tips to exploit hidden job market.

Here are some sources to get penetrating into a hidden job market: Friends; Family; Ex-coworkers; Referral; HR communities; Field communities; Social networks such as Facebook, Twitter…; Last recruitment ads from recruiters; HR emails of potential recruiters…

Tip 4: Do-It-Yourself Interviewing Practice

There are a number of ways to prepare for an interview at home without the help of a professional career counselor or coach or a fee-based service.

You can practice interviews all by yourself or recruit friends and family to assist you.

Useful material: 4career.net/free-ebook-75-interview-questions-and-answers

Tip 5: Ask questions

Do not leave the interview without ensuring that you know all that you want to know about the position. Once the interview is over, your chance to have important questions answered has ended. Asking questions also can show that you are interested in the job. Be specific with your questions. Ask about the company and the industry. Avoid asking personal questions of the interviewer and avoid asking questions pertaining to politics, religion and the like.

Ref material: 4career.net/25-questions-to-ask-employers-during-your-job-interview

Tip 6: Follow up and send a thank-you note

Following up after an interview can help you make a lasting impression and set you apart from the crowd.
Philip Farina, CPP, a security career expert at Manta Security Management Recruiters, says: "Send both an email as well as a hard-copy thank-you note, expressing excitement, qualifications and further interest in the position. Invite the hiring manager to contact you for additional information. This is also an excellent time to send a strategic follow-up letter of interest."

Ref material: 4career.net/top-8-interview-thank-you-letter-samples