

[HOME \(HOME.HTML\)](#) / [CLASSES IN PYTHON \(/CLASSES-IN-PYTHON.HTML\)](#)

Classes in Python

Like { 4 } Tweet Share 1 G+1 { 0 } 47K+

How to write a class in Python - Example

Below is a simple class written in Python

```
1 class base():
2     """This is a sample class in Python"""
3     def foo(self):
4         """This is a dummy method of class base"""
5         pass
6
7 if __name__=="__main__":
8     print "name of class:", base.__name__
9     print "class namespace:", base.__dict__
10    print "class inherited from:", base.__bases__
11    print "Docstring for this class:",base.__doc__
12    print "Module in which class 'base' is defined:", base.__module__
```

```
name of class: base
class namespace: {'__module__': '__main__', 'foo': , '__doc__': 'This is a sample class in Python'}
class inherited from: ()
Docstring for this class: This is a sample class in Python
Module in which class 'base' is defined: __main__
```

Predefine Class Attributes in Python

Predefine attributes of class in Python

Attribute	Type	Read/Write	Description
<code>__name__</code>	string	Read only	Name of class

<code>__dict__</code>	dictionary	Read and writeable	namespace of class
<code>__bases__</code>	Tuple of classes	Read only	class names where this class inherits from
<code>__doc__</code>	String or None	Read and writeable	Documentation string of this class
<code>__module__</code>	String	Read and writeable	Module name in which this class was defined

self in Python Class Methods

self is conventional name for the first parameter of a method of a class. This name isn't mandatory.

Inheritance in Python

This is how you can define a class that inherits methods (behaviors) from other class, referred as base class.

```

1 class derv(base):
2     """this class inherits from class base"""
3
4     def bar(self):
5         """this is just another dummy method"""
6         pass

```

Calling base class functions in Python

```

1 class human_being:
2     def __init__(self, human_name):
3         self.human_name=human_name
4         print "%s.%s" %(human_being.__name__, human_being.__init__.__name__)
5     def print_human_name(self):
6         print self.human_name
7
8 class Men(human_being):
9     def __init__(self, human_name):
10        #overriding functionality
11        print "%s.%s" %(Men.__name__, Men.__init__.__name__)
12        # explicitly have to call human_being's init if required.
13        human_being.__init__(self, human_name)
14
15 if __name__=="__main__":
16     inst=Men("charlie")
17     inst.print_human_name()
18     print isinstance(inst, Men)
19     print isinstance(inst, human_being)

```

```

Men.__init__
human_being.__init__
charlie
True
True

```

Multiple Inheritance in Python

You can have multiple inheritance in Python. That is you can create a class which inherits from multiple classes as shown in below example:

```

1 class human_being(object):
2     def __init__(self, human_name):
3         self.human_name=human_name
4         print "%s.%s" %(human_being.__name__, human_being.__init__.__name__)
5     def print_human_name(self):
6         print self.human_name
7
8 class country(object):
9     def __init__(self, country_name):

```

```

10         self.country_name = country_name
11         print "%s.%s" %(country.__name__, country.__init__.__name__)
12
13     class Men(human_being, country):
14         def __init__(self, human_name, country_name):
15             #overriding functionality
16             print "%s.%s" %(Men.__name__, Men.__init__.__name__)
17             # explicitly have to call human_being's init if required.
18             # Advance way of calling base class's function is using super()
19             super(Men, self).__init__(human_name)
20             country.__init__(self, country_name)
21
22     if __name__ == "__main__":
23         inst=Men("charlie", "Germany")
24         inst.print_human_name()
25         print isinstance(inst, Men)
26         print isinstance(inst, human_being)
27         print Men.__mro__

```

```

Men.__init__
human_being.__init__
country.__init__
charlie
True
True
(<class '__main__.Men'>, <class '__main__.human_being'>, <class '__main__.country'>, <type
'object'>)

```

Note usage of `super()` (<https://docs.python.org/2/library/functions.html#super>) to call super class methods. It follows MRO Method Resolution Order to resolve calls. MRO order for any class can be viewed using `__mro__` attribute as shown in program.

MRO examples:

1. The MRO order is left to right and depth first

```

1 class A(object):
2     pass
3 class B():
4     pass
5 class C(A):
6     pass
7 class D(C,B):
8     pass
9
10 if __name__ == "__main__":
11     print D.__mro__

```

```

(<class '__main__.D'>, <class '__main__.C'>, <class '__main__.A'>, <type 'object'>, <class
__main__.B at 0x7f9e62bb92c0>)

```

2.

```

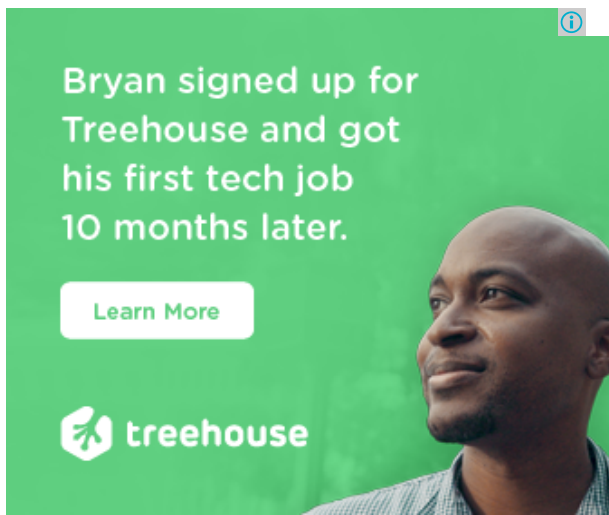
1 class A(object):
2     pass
3 class B(object):
4     pass
5 class C(A,B):
6     pass
7 if __name__ == "__main__":
8     print C.__mro__

```

```

(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>, <type 'object'>)

```



INTERVIEW QUESTIONS

[C++ \(/INTERVIEW-QUESTIONS/CPP/1\)](/INTERVIEW-QUESTIONS/CPP/1)

[JAVA \(/INTERVIEW-QUESTIONS/JAVA/1\)](/INTERVIEW-QUESTIONS/JAVA/1)

PYTHON

[PYTHON INTERVIEW QUESTIONS \(/PYTHON-INTERVIEW-QUESTIONS.HTML\)](/PYTHON-INTERVIEW-QUESTIONS.HTML)

[CLASSES IN PYTHON \(/CLASSES-IN-PYTHON.HTML\)](/CLASSES-IN-PYTHON.HTML)

[PYTHON CTYPES TUTORIAL \(/PYTHON-CTYPES.HTML\)](/PYTHON-CTYPES.HTML)

[PYTHON OBJECTS \(/PYTHON-OBJECT.HTML\)](/PYTHON-OBJECT.HTML)

[PYTHON PACKAGE INSTALLATIONS \(/HOW-TO-INSTALL-PACKAGES-IN-PYTHON.HTML\)](/HOW-TO-INSTALL-PACKAGES-IN-PYTHON.HTML)

C/C++ QUICK REFERENCE

[C++ STL - VECTOR LISTS DEQUE \(/CPP-STL-LIST-VECTOR-DEQUE.HTML\)](/CPP-STL-LIST-VECTOR-DEQUE.HTML)

[C++ STL - MAP, SET \(/CPP-STL-MAP-SET.HTML\)](/CPP-STL-MAP-SET.HTML)

[C++ STL - ITERATORS \(/CPP-STL-ITERATORS.HTML\)](/CPP-STL-ITERATORS.HTML)

[C++ - INLINE FUNCTIONS \(/INLINE-FUNCTIONS.HTML\)](/INLINE-FUNCTIONS.HTML)

[C++ - 2D DYNAMIC ARRAY \(/CPP-2D-ARRAY-CREATION-DYNAMICALLY.HTML\)](/CPP-2D-ARRAY-CREATION-DYNAMICALLY.HTML)

[STRING FUNCTIONS \(/STRING-FUNCTIONS-IN-C.HTML\)](/STRING-FUNCTIONS-IN-C.HTML)

[FILE HANDLING IN C \(/C-FILE-OPERATIONS.HTML\)](/C-FILE-OPERATIONS.HTML)

BINARY PROGRAMS IN C

[BINARY PRINT A INTEGER \(/C-PROGRAM-TO-PRINT-INTEGER-IN-BINARY.HTML\)](/C-PROGRAM-TO-PRINT-INTEGER-IN-BINARY.HTML)

[REVERSE BINARY PRINT A INTEGER \(/C-PROGRAM-TO-REVERSE-BINARY-PRINT-A-INTEGER.HTML\)](/C-PROGRAM-TO-REVERSE-BINARY-PRINT-A-INTEGER.HTML)

[REVERSE A INTEGER \(/C-PROGRAM-TO-REVERSE-A-INTEGER-VARIABLE.HTML\)](/C-PROGRAM-TO-REVERSE-A-INTEGER-VARIABLE.HTML)

[FIND POSITION OF LSB BIT SET \(/C-PROGRAM-TO-FIND-POSITION-OF-LSB-BIT-SET.HTML\)](/C-PROGRAM-TO-FIND-POSITION-OF-LSB-BIT-SET.HTML)

MULTITHREADING & MULTIPROCESSING

[MULTITHREADING WITH PTHREADS \(/PTHREADS.HTML\)](#)

[MULTIPROCESSING - FORK\(\) & EXEC \(/MULTIPROCESS-FORK-EXEC.HTML\)](#)

[PARALLEL PROCESSING \(/PARALLEL-PROCESSING.HTML\)](#)

IPC TECHNIQUES

[FIFO \(/NAMED-PIPES-FIFO.HTML\)](#)

[MESSAGE QUEUE \(/MESSAGE-QUEUE.HTML\)](#)

[PIPES \(/PIPES.HTML\)](#)

[PIPES USING POPEN\(\), PCLOSE\(\) \(/PIPES-TO-LINUX-PROCESS-USING-POPEN-PCLOSE.HTML\)](#)

[SIGNALS & HANDLERS \(/C-SIGNALS.HTML\)](#)

[SOCKET PROGRAMMING \(/SOCKET-PROGRAMMING.HTML\)](#)

IPC SYNCHRONIZATION TECHNIQUES

[CONDITION VARIABLE \(/CONDITION-VARIABLE.HTML\)](#)

[MUTEX \(/MUTEX.HTML\)](#)

[SEMAPHORES \(/SEMAPHORE.HTML\)](#)

[SPIN LOCKS \(/PTHREAD-SPINLOCK.HTML\)](#)

SORTING TECHNIQUES

[BUBBLE SORT \(/SORT/BUBBLE-SORT.HTML\)](#)

[SELECTION SORT \(/SORT/SELECTION-SORT.HTML\)](#)

[INSERTION SORT \(/SORT/INSERTION-SORT.HTML\)](#)

ARTICLES

[ATOM FEED TUTORIAL \(/ATOM-FEED.HTML\)](#)

[C CODE AUTO INDENT \(/C-CODE-INDENT.HTML\)](#)

[DJANGO ON SHARED SERVER \(/DJANGO-ON-SHARED-SERVER.HTML\)](#)

[FTP \(/FTP-UNIX-COMMAND.HTML\)](#)

[GDB DEBUGGING \(/GDB-TUTORIAL.HTML\)](#)

[GLUSTERFS INSTALLATION & PERFORMANCE \(/GLUSTERFS-INSTALLATION-PERFORMANCE.HTML\)](#)

[IPV6 SOCKET APIS \(/IPV6-SOCKET-API.HTML\)](#)

[IPV6 TUTORIALS \(/IPV6-TUTORIALS.HTML\)](#)

[JAVA EXECUTORSERVICE \(/JAVA-CONCURRENCY-EXECUTORSERVICE.HTML\)](#)

[JAVA PROGRAM FOR FILE APPEND \(/JAVA-FILE-APPEND-PROGRAM.HTML\)](/JAVA-FILE-APPEND-PROGRAM.HTML)


[ESSENTIAL LINUX COMMANDS \(/LINUX-COMMANDS.HTML\)](/LINUX-COMMANDS.HTML)

[LOOP DEVICE IN LINUX \(/LOOP-DEVICE-IN-LINUX.HTML\)](/LOOP-DEVICE-IN-LINUX.HTML)

[POSTGRESQL SETUP \(/POSTGRESQL.HTML\)](/POSTGRESQL.HTML)





[RESUME TIPS \(/RESUME.HTML\)](/RESUME.HTML)


[TAYGA \(/TAYGA.HTML\)](/TAYGA.HTML)


 Questions ...

Like Page

3 friends like this








**Spirite
Phone Sticky
2600mAh
Powerbank with
Aviator**

MRP - ~~₹1499~~

Selling Price
₹339

BUY NOW

 Ads by Google

► [Interview Questions](#)

► [Python Course](#)

► [Python Decorator](#)

► [Class in Python](#)

► [Python Code](#)

► [Python Examples](#)

Questions Compiled

[About \(/about.html\)](#)

[Contact Us \(/contactus.html\)](#)

[Feedback \(/feedback.html\)](#)

© 2012-2014. All Rights Reserved. [Read our Privacy Policy \(/privacypolicy.html\)](#)