

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ×

get python class parent(s)



How can I get the parent(s) object of python class?

[python](#) [oop](#)

asked Apr 10 '10 at 1:32



[John Smith](#)

4,344 6 42 81

- 27** @FranckDemonicourt This thread is now the top Google result for "python get parent class". It is slightly more relevant than the other SO questions that make up the next 3 results. The 5th result is a Python docs page that doesn't give the answer, and the 6th result is finally a Python docs page that gives the answer, albeit deep into a long page of text. — [Rodrigo Queiro](#) May 17 '12 at 18:26
- 14** Why would someone read the docs when explanations on SO are far more cogent and easy to access? It took me less than 5 seconds from the time this question occurred to me until the time I arrived at this page from Google, whereas searching the docs would have taken me at least five times as long and possibly a couple of minutes. — [Jon Crowell](#) Dec 31 '12 at 2:04
- 2** As someone who wanted to quickly find this answer, searched google, found this link, and got done what I needed to get done: Why are people (I assume SLott's comment was removed) complaining about a valid question being asked? And yes, I realize I'm extremely late to the party here. — [limasxgoesto0](#) Oct 25 '13 at 18:11

[add a comment](#)

4 Answers

Use the following attribute:

```
cls.__bases__
```

From the [docs](#):

The tuple of base classes of a class object.

Example:

```
>>> str.__bases__
(<type 'basestring'>,)
```

Another example:

```
>>> class A(object):
...     pass
...
>>> class B(object):
...     pass
...
>>> class C(A, B):
...     pass
...
>>> C.__bases__
(<class '__main__.A'>, <class '__main__.B'>)
```

answered Apr 10 '10 at 1:35



[Ayman Hourieh](#)

41k 7 99 100

[add a comment](#)

CAREERS 2.0
by stackoverflow

+

Creating a profile just got easier!
Import your LinkedIn to get started!

If you want all the *ancestors* rather than just the immediate ones, use `inspect.getmro`:

```
import inspect
print inspect.getmro(cls)
```

Usefully, this gives you all ancestor classes in the "method resolution order" -- i.e. the order in which the ancestors will be checked when resolving a method (or, actually, any other attribute -- methods and other attributes live in the same namespace in Python, after all;-).

answered Apr 10 '10 at 2:00

[Alex Martelli](#)**347k** 54 666 977[add a comment](#)

New-style classes have an `mro` method you can call which returns a list of parent classes in method resolution order.

answered Apr 10 '10 at 3:56

[DasIch](#)**1,615** 6 19[add a comment](#)

If you want to ensure they all get called, use `super` at all levels.

answered Apr 10 '10 at 2:29

[Mike Graham](#)**23.7k** 4 44 83

Once you use `super` you have to use it in all levels anyway, which is why you should document it's use explicitly. Also you might want to know that `super` doesn't work on every class... — [DasIch](#) Apr 10 '10 at 3:57

[add a comment](#)

Not the answer you're looking for? Browse other questions tagged [python](#) [oop](#) or [ask your own question](#).