

Re: from \_\_future\_\_ import absolute\_import ?

## Re: from \_\_future\_\_ import absolute\_import ?

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-02/msg01348.html>

---

- *From:* Peter Otten <[peter@xxxxxx](mailto:peter@xxxxxx)>
  - *Date:* Fri, 09 Feb 2007 11:27:57 +0100
- 

Ron Adam wrote:

Peter Otten wrote:

Ron Adam wrote:

```
work
|
|- foo.py # print "foo not in bar"
|
`- bar
|
|- __init__.py
|
|- foo.py # print "foo in bar"
|
|- absolute.py # from __futer__ import absolute_import
| # import foo
|
`- relative.py # import foo
```

\* Where "work" is in the path.

(1)

```
C:\work>python -c "import bar.absolute"
foo not in bar
```

```
C:\work>python -c "import bar.relative"
foo in bar
```

(2)

Re: from \_\_future\_\_ import absolute\_import ?

```
C:\work>python -m "bar.absolute"  
foo not in bar
```

```
C:\work>python -m "bar.relative"  
foo not in bar
```

(3)

```
C:\work>python  
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310  
32 bit  
(Intel)] on win 32  
Type "help", "copyright", "credits" or "license" for more  
information.  
>>> import bar.absolute  
foo not in bar  
>>> import bar.relative  
foo in bar
```

(4)

```
C:\work>cd bar
```

A path below the package level is generally a good means to shoot yourself in the foot and should be avoided with or without absolute import.

Seems so. :-/

```
C:\work\bar>python -c "import bar.absolute"  
foo in bar
```

```
C:\work\bar>python -c "import bar.relative"  
foo in bar
```

(5)

```
C:\work\bar>python  
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310  
32 bit  
(Intel)] on win 32  
Type "help", "copyright", "credits" or "license" for more  
information.  
>>> import bar.absolute
```

Re: from \_\_future\_\_ import absolute\_import ?

Re: from \_\_future\_\_ import absolute\_import ?

```
foo in bar  
>>> import bar.relative
```