

Here are 3 examples of syntax errors:

```
>>> print('and')
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
Syntax Error: print('and'): <string>, line 112
>>> if 3 > 2
...     print(3)
...
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
Syntax Error: if 3 > 2: <string>, line 19
>>> if 34 345:
...     print(34)
...
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
Syntax Error: if 34 345:: <string>, line 19
>>>
```

But could someone give me a clearcut example of a semantic error? And a definition that delineates semantic errors from syntax errors.

I googled this one up, <http://ubuntuforums.org/showthread.php?t=527855>, where the guy uses '&' where he should have used 'and'. That's clearcut, but is it really a semantic error? If he had used, say, '!' where he used '&' (which it seems has meaning in that context), that would be a syntax error, right?

```
>>> if 3 == 3 & 4 > 3:
...     print("Duh")
...
>>>

>>> if 3 == 3 ! 4 > 3:
...     print("Doh!")
...
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
Syntax Error: if 3 == 3 ! 4 > 3:: <string>, line 111
>>>
```

Is a semantic error one that Python doesn't see as an error -- no error is raised; whereas syntax errors aren't errors unless Python sees them as "Syntax Error"s?

Thanks,

Dick Moores