

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:



Ask programming questions



Answer and help your peers

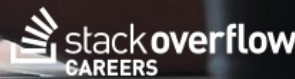


Get recognized for your expertise

Received "UnboundLocalError: local variable 'e' referenced before assignment" when the variable was initialized

Work on work you love.

From home.



[Community edit to give reproducible example:]

```
def main():
    e = None
    print(locals())
    while not e:
        try:
            raise Exception
        except Exception as e:
            pass

main()
```

produces

```
~/coding$ python3.3 quiz2.py
{'e': None}
Traceback (most recent call last):
  File "quiz2.py", line 11, in <module>
    main()
  File "quiz2.py", line 5, in main
    while not e:
UnboundLocalError: local variable 'e' referenced before assignment
```

[EDITED] to include a reproducible code

I am trying to run a while-loop, and the condition I use is that the loop continues when the variable `e==None` . The relevant code is below:

```
print("\nThe current score list contains the following people's scores: ")
score_list = open("score_list.dat", "rb")
score_name = []
e = None
while not e:
    try:
        score = pickle.load(score_list)
        name = pickle.load(score_list)
        score_name.append([score, name])
    except EOFError as e:
        pass
score_list_sorted=sorted(score_list)
sort_list.close()
for item in score_list_sorted:
    print("Score: ", item[0], "\t", item[1])
```

the complete code is here: [https://www.dropbox.com/s/1lj5xwexzfsoppv/stats\\_quiz\\_feb24\\_2013.py](https://www.dropbox.com/s/1lj5xwexzfsoppv/stats_quiz_feb24_2013.py)

The data file it requires (for the quiz to run) is in this link: [https://www.dropbox.com/s/70pbcb80kss2k9e/stats\\_quiz.dat](https://www.dropbox.com/s/70pbcb80kss2k9e/stats_quiz.dat)

`main()` needs to be edited to use the proper data file address:

The complete error message I received is below. This is weird because I initialized `e` right before the while-loop. I hope someone can help me resolve this problem. Thanks!

```
Traceback (most recent call last):
  File "<pyshell#217>", line 1, in <module>
    main()
  File "/Users/Dropbox/folder/stats_quiz_feb24_2013.py", line 83, in main
    while not e:
UnboundLocalError: local variable 'e' referenced before assignment
```

python exception exception-handling python-3.x while-loop

edited Feb 25 '13 at 22:45

asked Feb 25 '13 at 22:12

4 Aside: the line `score_name = score = name = []` probably doesn't do what you think it does. It doesn't make three lists, it makes *one* list, and binds three names to it. Try that at the console and then `score.append(3); print (score, score_name, name)` . It probably doesn't matter here because you rebind `score` and `name` , but still. – DSM Feb 25 '13 at 22:15

Ugh, I totally forgot about the issue of reference (coming from R, which doesn't have this issue). Thanks for the pointing that out! – Alex Feb 25 '13 at 22:17

Cannot reproduce. Please provide a minimal self-contained example. – wRAR Feb 25 '13 at 22:18

I don't know how this could be reproduced- there must be something in the code we can't see. Can you try making a reproducible example- that is, a short version of the code that can be run and produces this error? (It is likely that in the process of doing so, you will discover what is causing the problem in the first place). – David Robinson Feb 25 '13 at 22:18

The full code works for me. Please provide a minimal failing code, not this monstrosity. – wRAR Feb 25 '13 at 22:33

## 2 Answers

Well, I don't know what's causing the actual problem, but why don't you just use `break` when an exception occurs? Your loop becomes:

```
while True:
    try:
        score = pickle.load(score_list)
        name = pickle.load(score_list)
        score_name.append([score, name])
    except EOFError as e:
        break
```

As far as I know, this is the idiomatic way of achieving "run loop while there's no exception"

### Edit: Why this happens

It would seem that in python3, once you exit the scope of an exception handler, the variable that the exception was bound to is removed from the namespace. I modified the code to the following:

```
def main():
    e = None
    print(locals())
    while not e:
        try:
            raise Exception
        except Exception as e:
            pass
    print(locals())

main()
```

Output:

```
{'e': None}
{}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 4, in main
UnboundLocalError: local variable 'e' referenced before assignment
```

This is not the case in python2. Given that the syntax was changed for how you assign an exception to a variable, I'm not surprised that the semantics of it were changed as well. Although I do think that this is "surprising" behaviour (in the sense that it is not what you would expect).

In any case, the proper way to exit a loop when an exception occurs is in the code above. If you want to keep the exception outside of the exception handler's scope I guess you could still do

something like this:

```
def main():
    e = None
    print(locals())
    while not e:
        try:
            raise Exception
        except Exception as ex:
            e = ex
    print(locals())

main()
```

Which produces the following output:

```
{'e': None}
{'e': Exception()}
```

But you really shouldn't be doing that for your particular use-case.

edited Feb 25 '13 at 23:14

answered Feb 25 '13 at 22:49



entropy  
2,221 ● 10 ● 16

Thank you for the simple solution and the explanation! Much appreciated – [Alex](#) Feb 25 '13 at 23:16

- 1 nymk's answer has a much better explanation than mine about why this happens given that my experience with python3 is limited. The PEP he links to is particularly instructional if you have time to read it :) – [entropy](#) Feb 25 '13 at 23:21

Work on work you love. From home.



stackoverflow  
CAREERS

This error is caused by the new `try...except...` scope, which is a Python 3 feature.  
See [PEP-3110](#)

In Python 3, the following block

```
try:
    try_body
except E as N:
    except_body
...
```

gets translated to (in Python 2.5 terms)

```
try:
    try_body
except E, N:
    try:
        except_body
    finally:
        N = None
        del N
...
```

Therefore, this function in Python 3

```
def main():
    e = None
    print(locals())
    while not e:
        try:
            raise Exception
        except Exception as e:
            pass
```

is equivalent to

```
def main():
    e = None
    print(locals())
    if not e:
        try:
            raise Exception
        except Exception as e:
            pass
    del e
    if not e:
        try:
            raise Exception
        except Exception as e:
            pass
    del e
...
```

`e` was initialized, but it has been deleted after the first `try except` block.  
Thus, `UnboundLocalError` is inevitable.

answered Feb 25 '13 at 23:08



nymk  
2,427 ● 2 ● 17 ● 27

I haven't been using python3 too much(I'm still stuck in my old python2 ways for now). Thank you for that explanation :) I guessed the problem was changed semantics from python2 to python3 but couldn't figure out exactly how – [entropy](#) Feb 25 '13 at 23:16

- 1 Thank you for the explanation! I just started using Python, and was told by my friends that there are quite a few inconvenient differences between 3 and 2. (my friends only use python 2) – [Alex](#) Feb 25 '13 at 23:18