

Re: try: finally: question

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-07/0957.html>

From: Antoon Pardon (apardon_at_forel.vub.ac.be)

Date: 07/08/04

Date: 8 Jul 2004 07:57:37 GMT

Op 2004-07-07, Peter Hansen schreef <peter@engcorp.com>:

> *Antoon Pardon wrote:*

>

>> *Op 2004-07-06, Peter Hansen schreef <peter@engcorp.com>:*

> *[about using finally and except on the same try block]*

>>> *You used to be able to do that, according to a recent thread on the
>>>subject. It was removed because it was ambiguous what would happen.*

>>> *For example, if an exception were raised in a finally clause which
>>>came just before the except clause, would the except clause catch it?*

>>> *Since the finally is technically outside of the try block, some
>>>people thought it shouldn't be caught by the following except.*

>>> *Others did. Guido nuked it.*

>>>

>>> *Anyway, it would be purely syntactic sugar for what we have now,
>>>saving you a few keystrokes and one extra level of indentation.*

>>> *No big deal.*

>>

>> *I don't agree.*

>

> *With what statement don't you agree? The part about "you used to
> be able to do that... [but] it was ambiguous ... Guido nuked it."?*

That it is no big deal.

> *Because that's a fact. It's hard not to agree with it...*

>

> *And if it's the part about "syntactic sugar", then you should
> understand that the term refers to alternate syntax which is
> seen as "cleaner" or "simpler" in some way, but which is not
> otherwise different in any way that affects the parse tree or
> resulting code. And unless you think combined finally/except
> clauses should be treated differently in the generated bytecode
> than nested try/finally/except, it's *still* a fact that it
> would just be "syntactic sugar", and arguing against facts is
> pretty hard to do...*

>

>> *It is not unusual to nest try blocks. If you then
>> need to two instead of one because you can't combine a finally*

>> *with an except clause, you almost double the level of indentations.*
>
> *In fact, you would precisely double the level of indentations.*
> *It seems that's the price we have to pay for being human, since*
> *apparently our brains aren't all wired the same way and allowing*
> *finally/except to be combined was likely to lead to subtle bugs.*

Oh come on, I though python was for consenting adults?

Personnaly I think that allowing += operators for unmutable object will lead to subtle bugs too, because if you see code like

```
b = a
a += c
```

You don't know whether a and b are still the same object or not.

This will IMO be more confusing then a combined finally, except clause, because the latter can be clearly documented in how it behaves, while the above depends on the type of the variables.

>> *IMO it is almost as bad as would not having an elif and needing*
>> *an extra indentation level for each else .. if*
>
> *Almost, perhaps. But can you see any way for similar ambiguity*
> *to arise in the case of if/elif/else blocks? I think everybody*
> *understands how they work just fine. The whole point is that*
> *not everybody understood how the try/finally/except thing worked*
> *(or they didn't agree with how it should work).*

So what? When I complained about the ambiguity that was introduced by the introduction of the += and like operators, people told me that:

- 1) Error caused by not reading the docs carefull enough were my own fault
- 2) That I could always write things in such a way that there would be no ambiguity.

So these arguments seem to be just as valid with a combined except: finaly: clause.

> *I understand that you are arguing that it would be nicer to*
> *allow try/finally/except. I agree! Unfortunately, (a) it was*
> *removed and is probably unlikely to be added back in, (b) it*
> *was ambiguous, and the ambiguity has not been resolved, so*
> *until you or someone else can come up with a better approach*
> *(and write a PEP) it will stay the way it is. And that's just*
> *another fact, so it's pointless to try to argue against it.*

Well I think it really isn't that hard to define the behaviour of

```
try:
except expr1:
finally:
except expr2:
```

to be the same as

```
try:
  try:
    except expr1:
  finally:
except expr2:
```

In fact if it wasn't for the indentations rules in python I wouldn't care because without them I could simply write something like

```
try: try: try:
except expr1:
finally:
except expr2:
```

which is how I write my if ... else blocks in C.

However I'm not going to write a PEP. The impression that there is some double standard going on is just too strong, where powerfull, flexible and/or ambiguous concepts that are in the language are defended with the argument that python is for consenting adults and powerfull, flexible and/or ambiguous concepts that are not in the language are opposed because they can lead to subtle bugs.

Maybe the python community should make up its mind and decide what it wants. A language for consenting adults or protection against subtle bugs.

- > *Hmm... Also, I think it really *is* unusual (as in, "not usual"*
- > *meaning not really common) to nest try blocks. At least, I*
- > *recall lone try/finally or try/except blocks being much much*
- > *more common in most code than a mix. Perhaps my memory is*
- > *poor on this. Perhaps someone will actually go and research*
- > *it and prove it one way or the other. If it really is*
- > *very common, it might be worth revisiting the subject...*
- > *If it's not very common, then this is once again just another*
- > *discussion about syntactic sugar for an uncommon use case,*
- > *and that kind of thing is so last year. And the year before.*
- > *And...*

comp.lang.python: Re: try: finally: question

One can argue the opposite too. Why remove it, if the subtle bugs can occur only in unusual cases. Let those who need it and know what they are doing use it. That is what I understand about a language for consenting adults.

--

Antoon Pardon