

Myanmar Security Forum

[HOME](#)[MEMBERS](#)[HELP DOCS](#)[UPGRADE](#)[AWARDS](#)[BLACKLIST](#)[GROUPS](#)[Myanmar Security Forum - MSF](#)[Programming](#) [Python](#) [5 Error Styles in Coding](#)[TODAY'S POSTS](#)[NEW POSTS](#)[Thread Rating:](#)[New Reply](#)

5 Error Styles in Coding

[Thread Modes](#)**Hades.y2k**

Moderator



Posts: 287
Threads: 0
Thanks
Received: 65
in 31 posts
Thanks Given:
18
Joined: Feb
2014
Reputation:

103

02-01-2016, 09:30 AM (This post was last modified: 02-01-2016, 09:42 AM by Hades.y2k.)

#1

မနေ့က O'REILLY ရဲ့ How to make mistakes in Python ဆိုတဲ့စာအုပ်ဖတ်ပြီး အဲထဲကသဘောတူတဲ့အခန်းတခန်းကိုပြန်ရေးလိုက်တာပါ။ အခုပြောပြမှာကတော့ ဂျွန်ဇော်တို့တွေ code ရေးရာမှာမှားယွင်းနေတဲ့ style ၅ ခုအကြောင်းပါ။ ဒီအမှားတွေက code ကိုအလုပ်မလုပ်တာတွေဖြစ်ပါဘူး ဒါပေသိ ကိုယ်ရေးခဲ့တဲ့ code ကိုတချိန်ပြန်ဖတ်တဲ့အခါကတော့အော်ကိုဒုခပေးစေပါတယ်။

Hungarian Notation

ပထမဆုံးအမှားကတော့ Hungarian Notation လို့ခေါ်ပါတယ်။ သူကတော့ Variable Name ရှေ့မှာ variable ကဘာလဲဆိုတာကိုထည့်ရေးတာပါ။

Code:

```
strName = "PyLab"
str_First_Name = "Hades.y2k"
intYear = 2016
lstProducts = []
product_list = []
dictParams = {}
```

ဒီလိုမှိုးရေးကြတဲ့သူတွေရှိပါတယ်။ ဘာအတွက်လဲဆို ရည်ရွယ်ချက်ကတော့ နောင်တချိန် code ကိုပြန်ဖတ်တဲ့အခါဘယ်ကောင်ကဘာလဲသိဖို့ပါ။ သို့ပေမယ့် ထင်ထားသလိုဖြစ်မလာပဲ ဖတ်တဲ့အခါပိုကြည့်ရခက်သွားပါတယ်။ နောက်ရေးပါမှားလာတဲ့အခါအောက်ကလိုအမှားတွေပါဖြစ်လာပါတော့တယ်။

Code:

```
strCustomNumber = "12345"
bInFoo = "N"
```

ပြောရမယ်ဆိုပိုလွယ်လာမယ်လို့ထင်ပေမယ့် ထင်သလိုမဖြစ်ပဲ Hungarian Notation ကအနှောင့်အယှက်တစ်ခုသာဖြစ်စေပါတယ်။

PEP-8 Violation

PEP-8 ဆိုတာကတော့ Python code ရေးတဲ့အခါလိုက်တာသင့်တဲ့ standardized style တစ်ခုပါ။ အောက်ကကုဒ်ကိုတစ်ချက်ကြည့်လိုက်ပါ။

Code:

```

01) class MyGiganticUglyClass(object):
02) def iUsedToWriteJava(self,x,y = 42):
03)     blnTwoSpacesAreMoreEfficient = 1
04)     while author.tragicallyConfused():
05)         print "Three spaces FTW roflbbq!!1!"
06)     if (new_addition):
07)         four_spaces_are_best = True
08)     if (multipleAuthors \
09)         or peopleDisagree):
10)         print "tabs! spaces are so mainstream"
11)     ...

```

Line Number 2 မှာပြောနိုင်တာရှိပါတယ်။ ဒဲ့တာတွေကတော့ Indentation issue, Whitespace issue နဲ့ Inconsistent case ရယ်ပါ။

Indentation issue ကတော့ ဒုတိယလိုင်းမှာ class နဲ့ def က whitespace ၂ ခုသာရှိပါတယ် သို့သော် def နဲ့ bln... မှာတော့ whitespace ကုဒ်နဲ့ဖြစ်သွားပါတယ်။ နောက် tab နဲ့ whitespace ကရောသုံးတာမျိုးကိုရှောင်ပါ။
 ဂျွန်တော် rule ဖတ်ဖူးတာကိုပြောပြရရင် အကောင်းဆုံးကတော့ whitespace 4 ခုနဲ့ indentation ကိုလုပ်ပါ။
 Python မှာက program flow အတွက် whitespace/tab ကိုကြည့်တာပါ ဒီတော့ whitespace/tab တွေရောသုံးပြီး သက်ဆိုင်ရာ code line က သူနဲ့ဆိုင်တဲ့ block မှာရှိမနေခဲ့ရင် logic error ပါဖြစ်သွားနိုင်ပါတယ်။

whitespace issue ကိုပြောနိုင်တာကတော့ Line number 2 က parentheses ထဲက , ပြီးရင် whitespace မခံသလို variable name နဲ့ value ကြားမှာ whitespace တွေထည့်ထားပါတယ်။ self,x,y = 42
 ဒါကအမှားတစ်ခုပဲ တကယ်ဆို comma (,) ပြီးရင် whitespace တစ်ခုကပ်ရသလိုနောက်တစ်ခုက y=42 ဖြစ်သင့်တာပါ။
 နောက်ဒီ code တစ်ခုလုံးမှာလဲ class နဲ့ function definitions တွေကြားမှာ blank line ထည့်လိုက်ရင် code ကိုဖတ်လို့ပိုအဆင်ပြေသွားမှာပါ။

Inconsistent case ကတော့ function name ပါ။ Python community ကပြောတာကတော့ variable name တွေကို lowercase နဲ့ underscore ကိုဖတ်ရလွယ်ကူစေရန်အတွက်အသုံးပြုခိုင်းပါတယ်။ ဥပမာဒဲ့ function name ကိုပြန်ရေးပြရမယ်ဆို i_used_to_write_java ပေါ့။

ဣန္ဒြေ line တွေကိုပြောရမယ်ဆို

Line number 3 မှာကလဲ Indentation issue, Inconsistent case နဲ့ Hungarian Notation (I already explained at the beginning of the post) errors တွေဖြစ်နေပါတယ်။

Line number 4 မှာကတော့ Inconsistent case တက်နေပါတယ်။

Line number 6 မှာကတော့ **Extraneous parentheses error** တက်နေပါတယ်။ ဒဲ့တာကတော့ argument တစ်ခုပဲရှိတဲ့ကောင်ကို parentheses နဲ့ပိတ်ထားတာပါ။ တကယ်လို့ Single argument ပဲရှိတယ်ဆို parentheses နဲ့ပိတ်စရာမလိုပါဘူး။

Line number 8 မှာမှားနေတာကတော့ Inconsistent case နဲ့ Extraneous line continuations ပါ။

Extraneous line continuations ဆိုတာကိုပြောရမယ်ဆို parentheses အထဲမှာဆိုရင် statement ကိုနောက်တလိုင်းမှာဆက်ရေးတဲ့အခါဂျာရင် backslash (\) မလိုပါဘူး။

နောက်ဆုံးလိုင်းမှာဆိုရင်လဲ Extraneous parentheses error ဖြစ်နေပါတယ်။

ဒီအမှားတွေကိုအကုန်ပြင်ထားတဲ့အောက်ကကုဒ်လေးကိုတစ်ခွက်ကြည့်ကြည့်ပါ။ သေချာတာကတော့အပေါ်ကကောင်ထက်ဖတ်ရတာတော်တော်အဆင်ပြေပါတယ်။

Code:

```
class MyMorePythonicClass(object):

    def now_i_write_python(self, x, y=42):
        two_spaces_hamper_readability = True

        while author.tragically_confused():
            print "Three spaces? What was I thinking?"

        if new_addition:
            four_spaces_are_best = True
```

Bad Naming

Bad Naming ကတော့ဂျာနယ်တော်ကိုယ်တိုင်လဲရေးနေတဲ့သူထဲမှာပါပါတယ်။ ဘယ်ကနေကူးသလဲဆို Python စာအုပ်တွေကပါပဲ။ စာအုပ်တွေက example တွေမှာဆို x, y ထို့လို single character variable တွေသုံးလေ့ရှိပါတယ် for loop ပတ်တာတွေဆိုလဲ for x in ... နောက်ဂျာနယ်တော်ကိုယ်တိုင်လဲ script တွေရေးတော့ for loop ပတ်ရင် ပတ်တဲ့ကောင်ရဲ့နာမည်ကိုသေချာမစဉ်းစားတော့မိပဲ single character variable တွေနဲ့ပေးဖြစ်တော့တာပဲ။

ဒီလိုရေးတာတွေက တကယ်ဒုခပေးမပေးအောက်ကကောင်လေးဖတ်ကြည့်ပါလား။

Code:

```
f.write(string.join(map(lambda
x,y=self.__dicProfiles,z=strPy:"%0.3s %s:
%s:(%s)" % (z,x,y[x][0],y[x]
%[1]),self.__dicProfiles.keys()),'\n')
%+ '\n')
```

ဒီလိုအကုန်လေးတွေကို တကယ် script ရေးသားတဲ့အချိန်မှာဖောက်သင့်ပါတယ် single-character variable အပြင်နောက်တခုက long names ဆိုတဲ့ name ကိုအရှည်ကြီးပေးတာတွေပါ ဥပမာအောက်ကကောင်လိုပေါ့

Code:

```
class makeSomethingFancywithThisClass():
```

ဒီလိုရေးတာတွေက function name သာဆိုခင်ဗျားပြန်ခေါ်တိုင်း characters တိုင်းအတိအကျမှတ်မိနိုင်သေးလို့လား typo error တွေဖြစ်ပြီး coding ပိုင်းမှာမလိုလားအပ်တာတွေဖြစ်စေနိုင်ပါတယ်။
Name တစ်ခုသတ်မှတ်တဲ့အခါနာမည်ကို လိုရင်းတိုရှင်းဖြစ်ရသလို သူ့ရဲ့လုပ်ဆောင်ချက်နဲ့သက်ဆိုင်တဲ့အဓိပ္ပာယ်လဲရှိရပါမယ်။

Inscrutable Lambdas

Lambda function တွေကိုရှာရင်တော့ anonymous functions တွေရေးရင်အသုံးပြုပါတယ်။ Lambda နဲ့ရေးရင် concise ဖြစ်ပေမယ့်လို့ ဖတ်တဲ့သူအတွက်အနည်းငယ်တော့ခက်ခဲပါမယ်။

Lambda function ဟုခုခုကို expression တစ်ခုထဲမှာထည့်သုံးရင် confuse ဖြစ်ဖို့လဲလွယ်သလို code ကိုပြန်ဖတ်တဲ့အခါ ကျွန်တို့တစ်ခုယူပြီးပြန်ဖတ်ရတာမျိုးတွေဖြစ်တတ်ပါတယ်။ နောက် anonymous functions တွေက normal functions တွေလို reusable မဖြစ်ပါဘူး။ အောက်က ဥပမာကိုယှဉ်ကြည့်ပါ။ Lambda နဲ့ရေးတဲ့အခါ same function ကို ခုခုရေးရပေမယ့် ပုံမှန်ရေးနေကြအတိုင်းဆို function ကိုခါခါရေးပြီးကြိုက်သလောက်ပြန်ခေါ်လို့ရပါတယ်။

Using Lambda

Code:

```
foo = map(lambda x: x[-1].replace('pylab', 'python'), foos)
example = map(lambda x: x[-1].replace('pylab', 'python'), examples)
sample = map(lambda x: x[-1].replace('pylab', 'python'), samples)
```

In Normal Way

Code:

```
def pylab_to_python(input):
    return input[-1].lower().replace('pylab', 'python')

foo = map(pylab_to_python, foos)
example = map(pylab_to_python, examples)
sample = map(pylab_to_python, samples)
```

Incomprehensible Comprehensions

List comprehensions ဆိုတာတော်တော်လေးအသုံးဝင်ပါတယ်။ Example ဆိုအောက်ကလိုကောင်မို့ပေါ့

Code:

```
noprimes = [j for i in range(2, 8) for j in range(i*2, 50, i)]
```

ပထမ example ကသိပ်မရှုပ်ပေမယ့် အောက်ကကောင်လေးထပ်ကြည့်လိုက်ပါ။

Code:

```
prop_list = [
    FilterProp(prop='P_EXCLUDED', data='.'.join([i, j, k]))
    for i in prop_data[0]
    for j in prop_data[1]
    for k in prop_data[2]]
```

သေချာတာတော့ အနည်းငယ်တော့တွေးရမှာပါ ဘာလုပ်တယ်ဆိုတာကို

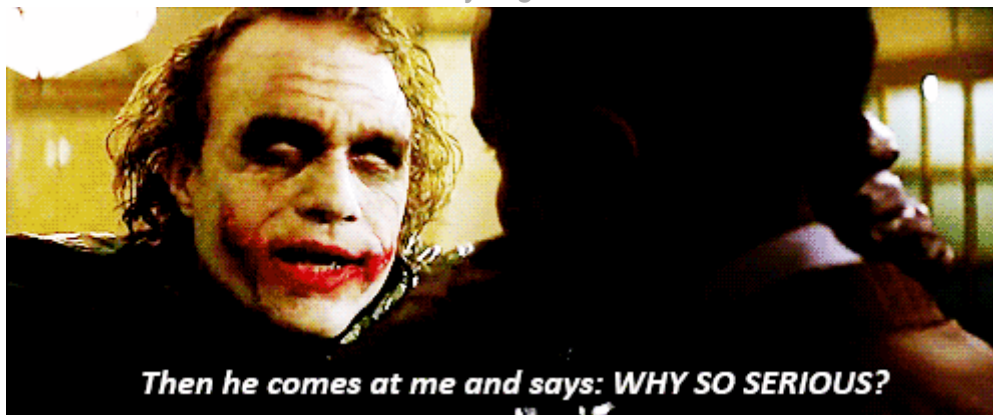
List comprehensions တွေရေးခဲ့အခါမှာ သတိပြုရမှာက comment တွေထည့်ပေးဖို့ပါ နောက်တစ်ခုကအပေါ်ကလို single-character name တွေအသုံးမပြုဖို့ပါ နောက်ဒီထက်ပိုရှင်းစေခွင့်ရင်တော့ List comprehensions အစား separate functions တွေရေးပြီး ရှုပ်နေတာကိုခွဲထုတ်လိုက်ပါ။

Error Style ၅ခုအကြောင်းကတော့ဒါပါပဲ။

အပေါ်ကပြောသွားတဲ့အခွက်လေးတွေကို တကယ်ရေးတဲ့အခါမှာလဲအကောင်အထည်ဖော်သွားသင့်ပါတယ်။ ဒါက program error မတက်ပေမယ့်လဲ ကုန်ကျတန်ဖိုးပြန်ဖတ်တာ ဒါမှမဟုတ်တစ်ခုပြင်တဲ့အခါမှာဒီလိုရေးတယ်မရေးဘူးကစကားပြောလာမှာပါ။

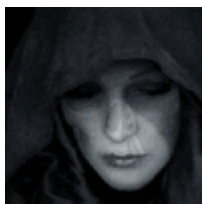
Ref: [How to make mistakes in Python \(O'Reilly\)](#)

hadesy2k.github.io



PM Find

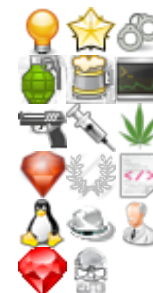
Add Thank You Reply Quote Report

**133720**

~I love Coding~ Fuck Girl.



Posts: 2,521
Threads: 0
Thanks
Received:
1,375 in 951
posts
Thanks Given:
758
Joined: Jun
2013
Reputation:
272




02-01-2016, 09:37 AM

#2

Thanks for written about the mistake of python written. 🙏 i also will notice and try to fix this problems. Rep+

Quick Reply



PMFind

Add Thank YouReplyQuoteReport

« Next Oldest | Next Newest »

Enter Keywords

Search Thread

New Reply

Quick Reply

Message

Type your reply to this message here.

☐ Disable Smilies

Post Reply

Preview Post

| Possibly Related Threads... | | | | |
|-----------------------------------|--------|---------|-------|---|
| Thread | Author | Replies | Views | Last Post |
| [Python] Error Based SQL injector | Anubis | 3 | 90 | 10-24-2013, 11:56 PM Last Post: Deep Th!nk3r |
| 08-29-2013, 04:36 AM | | | | |

[\[python\] GUI coding](#)

Anubis

7

120

Last Post: 133720

[View a Printable Version](#)[Subscribe to this thread](#)

Forum Jump:

Go

Users browsing this thread: Hacke3erDD

Myanmar Security Forum (MSF) © 2013 - 2016 - All Rights Reserved.

Powered By MyBB, © 2002-2016 MyBB Group. — Theme by FlatInk LLC.

[Contact Us](#) — [Return to Top](#) — [Lite \(Archive\) Mode](#) — [RSS Syndication](#) | [Awards](#)