Python 3.3.3 TypeError : Can't convert 'NoneType' object to str implicitly

I've been trying to program a Mad-Libs Simulator in Python 3.3.3 and I have been receiving the error:

```
Traceback (most recent call last):
File "/Users/RuzHayes_Laptop/Desktop/Programming:Design/Programs/Python Mad Libs
Program 000.py", line 80, in <module>
templates=[("The"+" "+adj+" "+n+" "+v+" "+adv+pun),(adj+" "+pluralize(n)+' '+
(v[:len(v)-1])+" "+adv+pun)]
TypeError: Can't convert 'NoneType' object to str implicitly
```

In the following code:

```
print("Welcome!")
print("When you would like to try out ")
print("Python Mad Libs Program, Prototype000,")
begin=input("press the enter/return key.")

print()
print()
print("Initializing befuddlement...")
print()

import random
sentenceCap=35
sentenceBottom=25
numOfSentences=random.randint(sentenceBottom,sentenceCap)
caps=
{"a":"A","b":"B","c":"C","d":"D",'e':'E','f':'F','g':'G','h':'H','i':'I','j':'J','k':'

tempstore=[" "]*numOfSentences
irregplrls={'child':'children','ox':'oxen','moose':'moose'}
def testforoo(x):
    for j in range(0,len(x)):
        if j+1<=len(x) and x[j:j+1]=='oo'.lower():
            return True
    return False
def pluralize(x):
    l=len(x)
    for i in irregplrls:
        if i == x:
            return irregplrls[x]
    if x[l-1]=="y":
        return x[:l-1]+"ies"
    elif x[l-1]=="s" and x[l-2]=="u":
        return x[:l-2]+"i"
    elif x[l-1] in ('s','x'):
        return x+"es"
    elif x[-2:] in ('ch','sh'):
        return x+"es"
    elif 'f'==x[l-1] or x[l-2]:
        if 'f'==x[l-1]:
            return x[:l-1] + 'ves'
        elif 'f'==x[l-2]:
            return x[:l-2]+"ves"
    elif testforoo(x)!=False:
        return x[:testforoo(x)-2]+'ee'+x[testforoo(x):]
    else:
        return x+'s'

print()

print("Retrieving craploads of words...")
print()

verb=["moves","jumps", "hides","sniffs","gazes","sneezes","calls"]
noun=["rabbit","dog","cat","otter","seal","elephant","fox",'baby','moose','octopus']
adjec=["happy","angry","cute","enormous","elegant","annoying"]
adver=["merrily","frustratedly","incoherently","morosely","peppily",'exuberantly']
endpunct=[".","!"]
```

```
print()
print("Simulating human grammar-speak...")
print()
print()

for i000 in range(0,numOfSentences):
    v=random.choice(verb)
    n=random.choice(noun)
    adj=random.choice(adjec)
    adv=random.choice(adver)
    pun=random.choice(endpunct)
    askinput=random.randint(0,round(numOfSentences/5))
    whichinput=random.randint(0,3)
    if askinput==0:
        if whichinput==0:
            n=input("Please input a noun. ")
        elif whichinput==1:
            v=input("Please input a verb. ")
        elif whichinput==2:
            adj=input("Please input an adjective. ")
        elif whichinput==3:
            adv=input("Please input an adverb. ")
    templates=[("The"+" "+adj+" "+n+" "+v+" "+adv+pun),(adj+" "+pluralize(n)+' '+
(v[:len(v)-1])+" "+adv+pun)]
    final=templates[random.randint(0,len(templates)-1)]
    if final[:1]==final[:1].lower():
           final=caps[final[:1]]+final[1:]
    tempstore[i000]=final

print()
print()
print("Producing proof of illiteracy...")
print()
print()

for i001 in range(0,len(tempstore)):
    sent=tempstore[i001]
    print(sent)
```

I am very confused and I need help. I have now found the problem to be in the end of the pluralize definition, but otherwise I am very confused. The program worked until I changed pluralize to account for certain nouns not being correctly pluralized.

I will appreciate any help you can give me. I have only been programming for about a month now and this is the hardest program I have ever attempted.

Thank you and Merry Christmas! That is, of course, if you celebrate Christmas.

python python-3.x

edited Dec 25 '13 at 1:21

asked Dec 25 '13 at 0:15
PythonDabbler
3 ●1 ●3

Obviously one the objects you attempting to insert into the string in `None`. – ApproachingDarknessFish Dec 25 '13 at 0:17

1 `for i000 in...` and `for i0001`... Oh god, why these names? – Maxime Lorant Dec 25 '13 at 0:22

2 Answers

```
def testforoo(x):
    for j in range(0,len(x)):
        if j+1<=len(x) and x[j:j+1]=='oo'.lower():
            return True
    return False
```

is

```
def testforoo(x):

    for j in range(0,len(x)):
        if x[j]=='oo'.lower():
            return True
    return False
```

because if `len(x)` is 4, then `j` runs from 0 to 3 and `j+1` from 1 to 4, then `j+1` always <= `len(x)`.

it is also

```
def testforoo(x):
    return any(x[j]=='oo'.lower() for j in range(0,len(x)))
```

The reason of the error is when `pluralize(n)` returns `None` (I don't know in which case), the addition of `None` to a string can be processed.

If you use the formatting with `%s` (or with `format`), there won't be the problem anymore.

I improved your code at certain places. See:

```
begin=input("Welcome!\n"
            "When you would like to try out\n"
            "Python Mad Libs Program, Prototype000,"
            "press the enter/return key.")

print("\nInitializing befuddlement...")

import random
sentenceCap=35
sentenceBottom=25
numOfSentences=random.randint(sentenceBottom,sentenceCap)

tempstore=[" "]*numOfSentences
irregplrls={'child':'children','ox':'oxen','moose':'moose'}

def testforoo(x):
    return any(x[j]=='oo'.lower() for j in range(0,len(x)))

def pluralize(x):
    if x in irregplrls:
        return irregplrls[x]

    if x[-1]=="y":
        return x[:-1]+"ies"
    elif x[-2:]=="us":
        return x[:-2]+"i"
    elif x[-1] in 'sx':
        return x+"es"
    elif x[-2:] in ('ch','sh'):
        return x+"es"
    elif 'f'==x[-1] or x[-2]:
        if 'f'==x[-1]:
            return x[:-1] + 'ves'
        elif 'f'==x[-2]:
            return x[:-2]+"ves"
    elif any(x[j]=='oo'.lower() for j in range(0,len(x))):
        return x[:testforoo(x)-2]+'ee'+x[testforoo(x):]
    else:
        return x+'s'

print("\nRetrieving craploads of words...")

verb=["moves","jumps", "hides","sniffs","gazes","sneezes","calls"]
noun=["rabbit","dog","cat","otter","seal","elephant","fox",'baby','moose','octopus']
adjec=["happy","angry","cute","enormous","elegant","annoying"]
adver=["merrily","frustratedly","incoherently","morosely","peppily",'exuberantly']
endpunct=[".","!"]

print("\nSimulating human grammar-speak...\n")

for i000 in range(0,numOfSentences):
    v=random.choice(verb)
    n=random.choice(noun)
    adj=random.choice(adjec)
    adv=random.choice(adver)
    pun=random.choice(endpunct)
    askinput=random.randint(0,round(numOfSentences/5))
    whichinput=random.randint(0,3)
    if askinput==0:
        if whichinput==0:
            n=input("Please input a noun. ")
        elif whichinput==1:
            v=input("Please input a verb. ")
        elif whichinput==2:
            adj=input("Please input an adjective. ")
        elif whichinput==3:
            adv=input("Please input an adverb. ")
    templates=["The %s %s %s %s%s" % (adj,n,v,adv,pun),

               "%s %s %s %s%s" % (adj,pluralize(n),v[:len(v)-1],adv,pun)]
    final = random.choice(templates)
    final=final[0].upper() + final[1:]
    tempstore[i000]=final
    print('numOfSentences==',numOfSentences,i000)

print("\nProducing proof of illiteracy...\n")

print ('\n'.join(tempstore))
```

I'm not sure of what the function `pluralize()` does, I let the function `testforoo()` and didn't try to modify this part, examine it carefully, the instruction `elif 'f'==x[-1] or x[-2]:` seems suspect to me, it must be `elif 'f'==x[-1] or 'f'==x[-2]:` in my opinion.

answered Dec 25 '13 at 1:33

eyquem
13.3k ●4 ●18 ●31

Thanks, that was very helpful! However, every time the pluralize function is used, it prints None rather than a plural. It doesn't raise an error, though. How can I fix that? – PythonDabbler Dec 25 '13 at 1:46

To fix it, you should have taken in account what I wrote: the instruction `elif 'f'==x[-1] or x[-2]:` isn't good. It is equivalent to `elif ('f'==x[-1]) or (x[-2]):`. There's nothing to say concerning `'f' ==x[-1]`. But `x[-2]` is evaluated as a bool value; But in Python, every non-null object is evaluated to True. So when no otehr condition before `elif ('f'==x[-1]) or (x[-2]):` is verified, the program enters in this test and evaluates first `if 'f' ==x[-1]` and then `elif 'f'==x[-2]`. However when `'f'` is neither the last or penultiem letter, none of these two conditions... – eyquem Dec 25 '13 at 4:04

...is verified. So the programs has, in this case no `return` to execute. When a function doesn't execute any return statement, it returns None. – eyquem Dec 25 '13 at 4:06

Ok thanks! I was sure that I had tried 'f==x[-1] or 'f==x[-2] though. Hmmm... – PythonDabbler Dec 26 '13 at 23:08 ✎

I must draw your attention on an inconsistency that remaiend in my code but that came from your own code. In the `testforoo')` function, you wrote a test `x[j:j+1]=='oo'.lower()` bur `x[j:+1]` describes a string of length 1, that is to say one character only, while it is compared to the two characters string `'oo'`. You must correct this inconsistency. – eyquem Dec 27 '13 at 1:42 ✎

If one or more of `n`, `v`, `adj`, or `adv` is not being assigned correctly, then its value is `None`. The line your error is on is concatenating strings together, but if one of the variables is not a string, it throws the error you're seeing.

Try printing the values of those variables at different places until you can track down exactly where the issue with their assignment lies.

answered Dec 25 '13 at 0:18

MattDMo
44.7k ● 11 ● 86 ● 106