Exposure CS 2020 for CS1

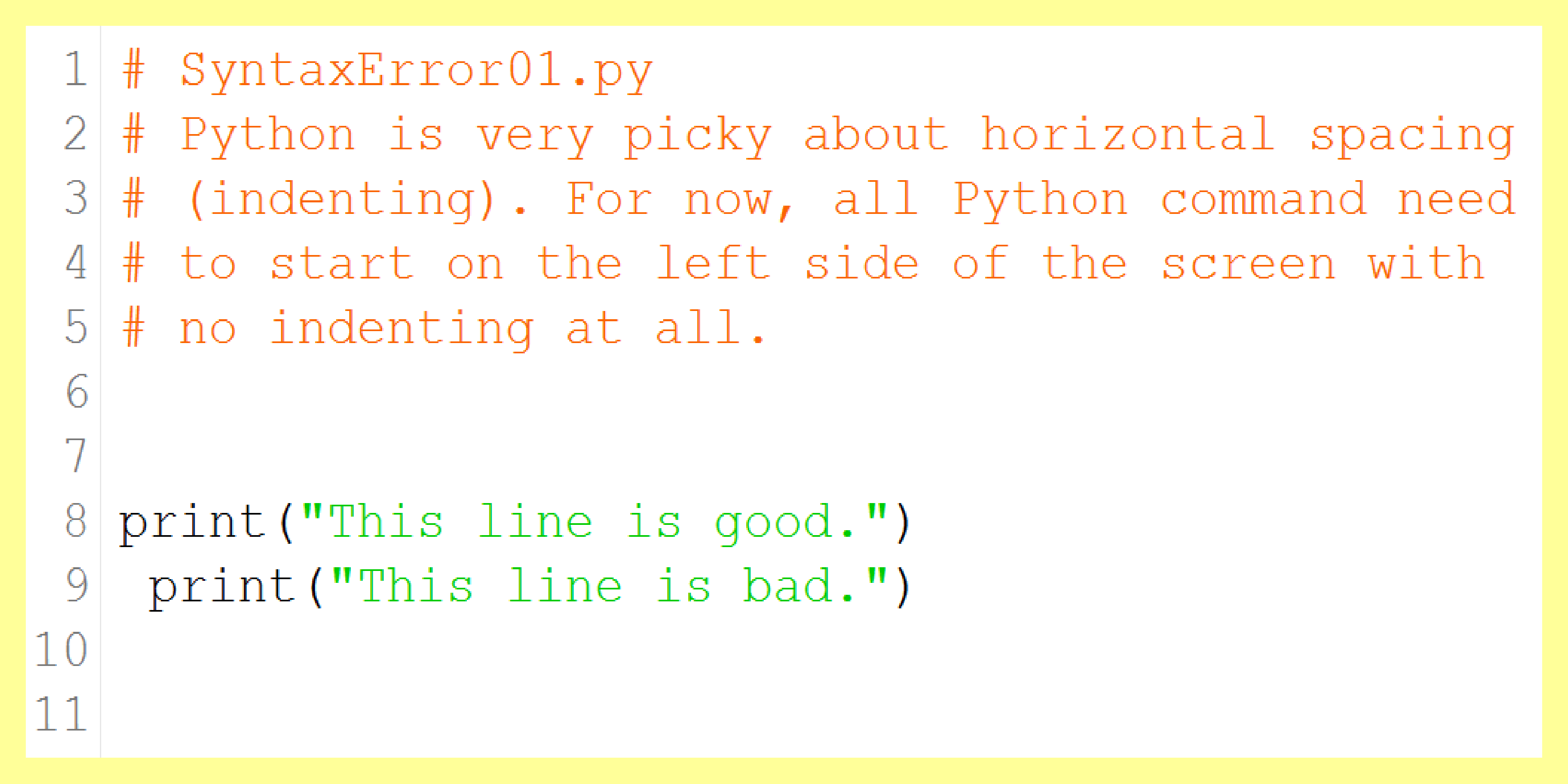
Chapter 3 Section 5-7 Slides

Syntax Errors, Issues with Quotes & The Responsible Use of Computers

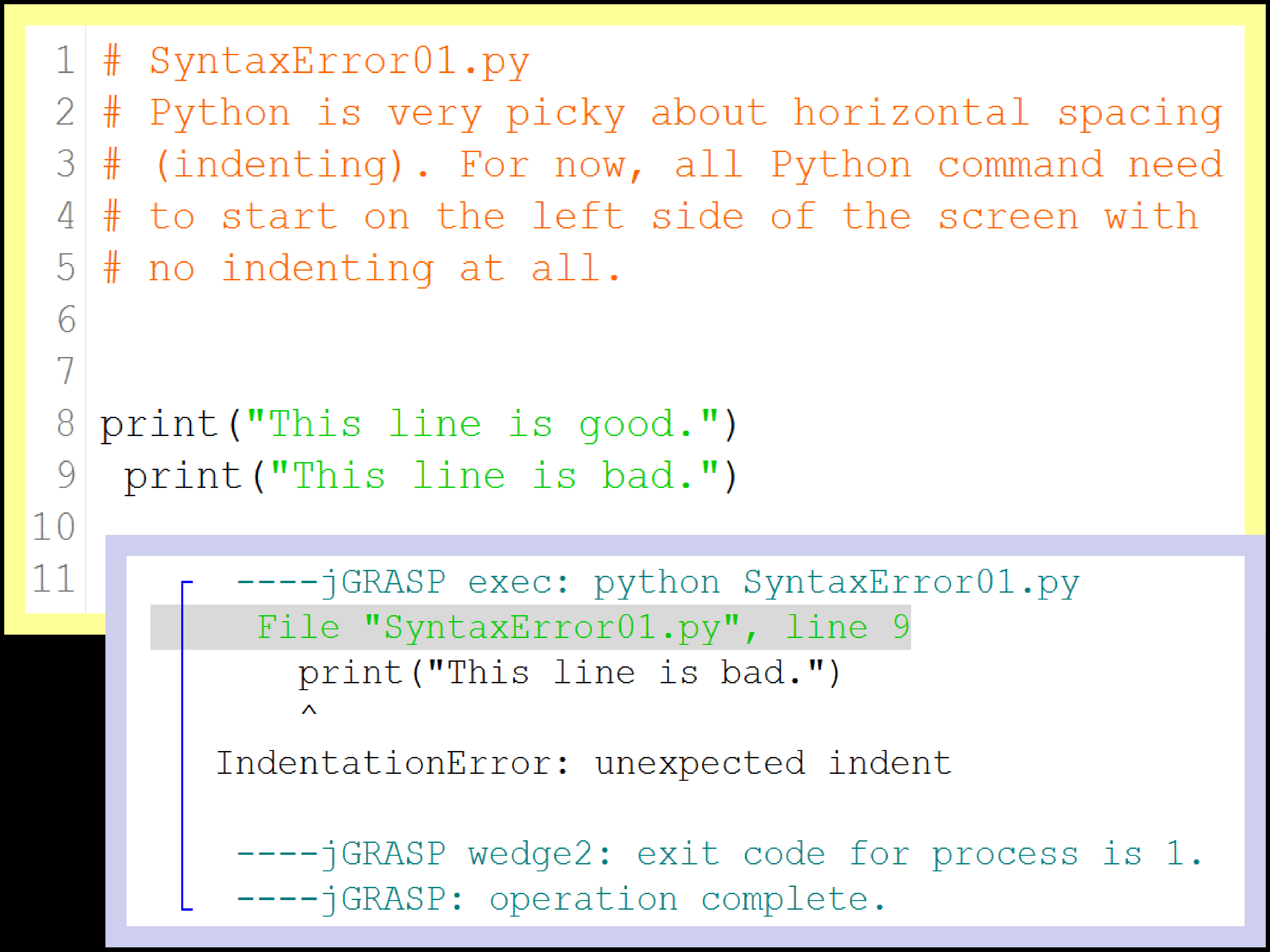
PowerPoint Presentation
created by:
Mr. John L. M. Schram
and Mr. Leon Schram
Authors of Exposure
Computer Science

Section 3.5

Syntax Errors

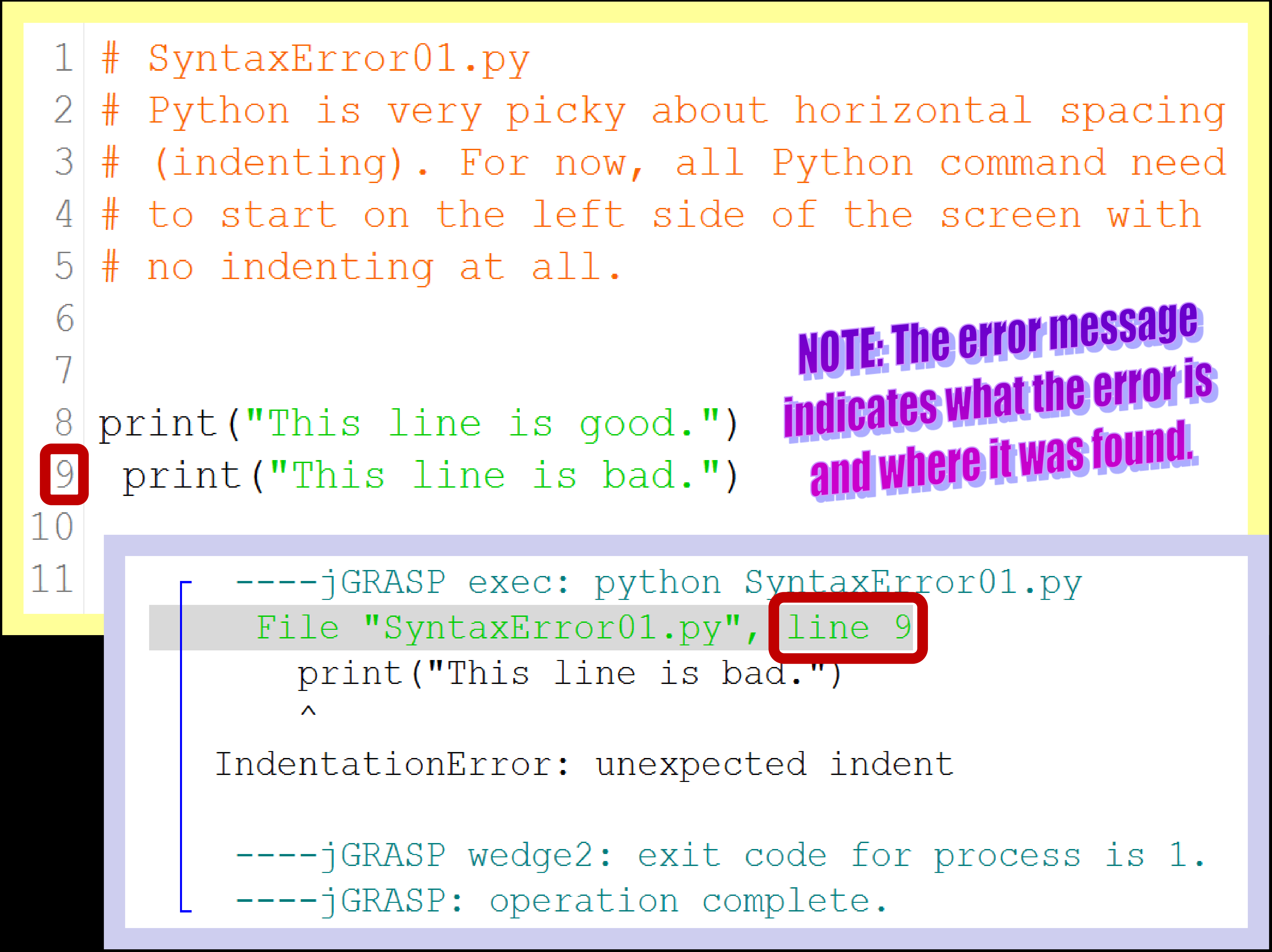


```python
# SyntaxError01.py
# Python is very picky about horizontal spacing
# (indenting). For now, all Python command need
# to start on the left side of the screen with
# no indenting at all.


print("This line is good.")
 print("This line is bad.")


```



```
 1 # SyntaxError01.py
 2 # Python is very picky about horizontal spacing
 3 # (indenting). For now, all Python command need
 4 # to start on the left side of the screen with
 5 # no indenting at all.
 6
 7
 8 print("This line is good.")
 9  print("This line is bad.")
10
11
```

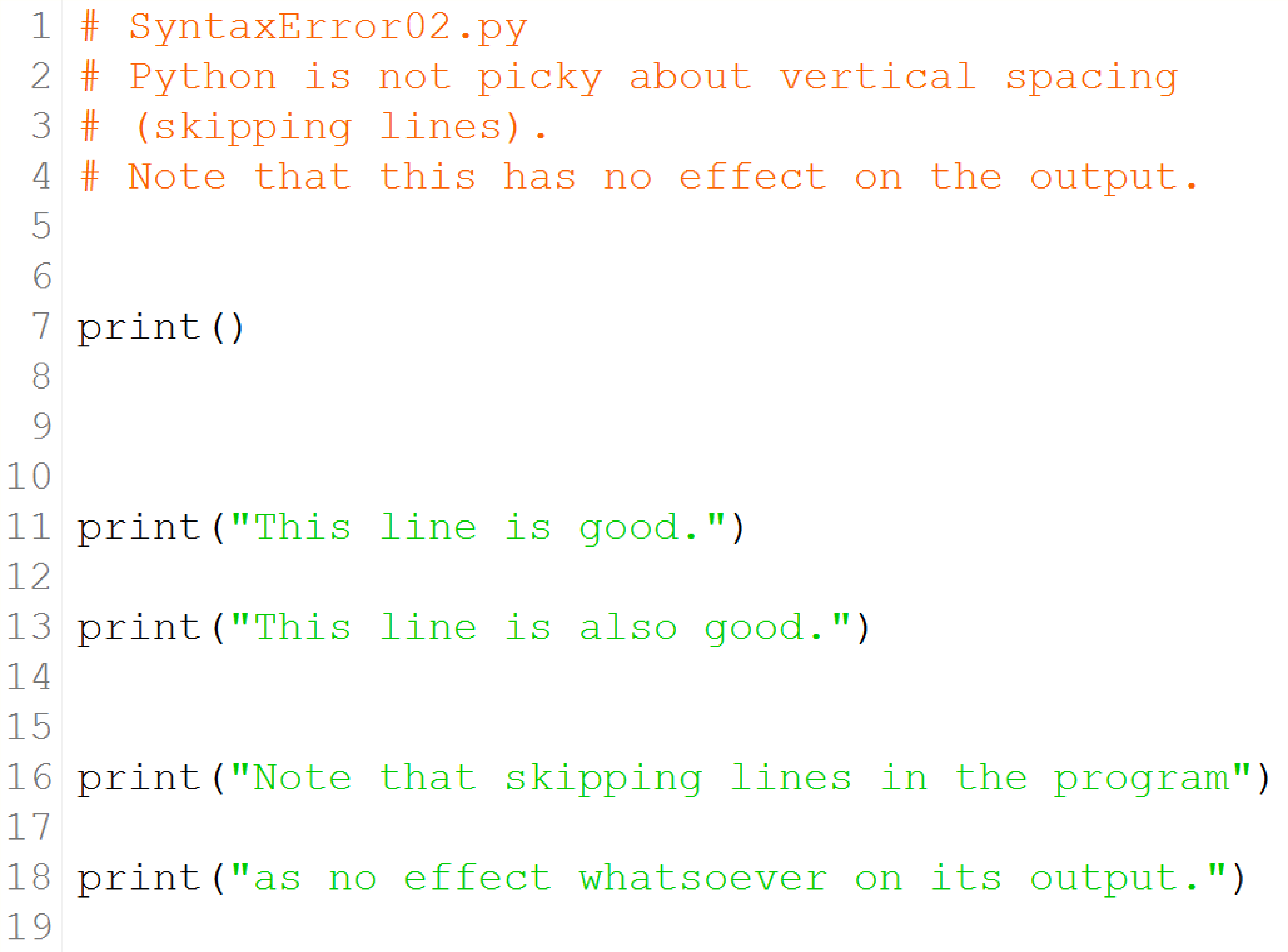
```
 ----jGRASP exec: python SyntaxError01.py
  File "SyntaxError01.py", line 9
     print("This line is bad.")
     ^
IndentationError: unexpected indent

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



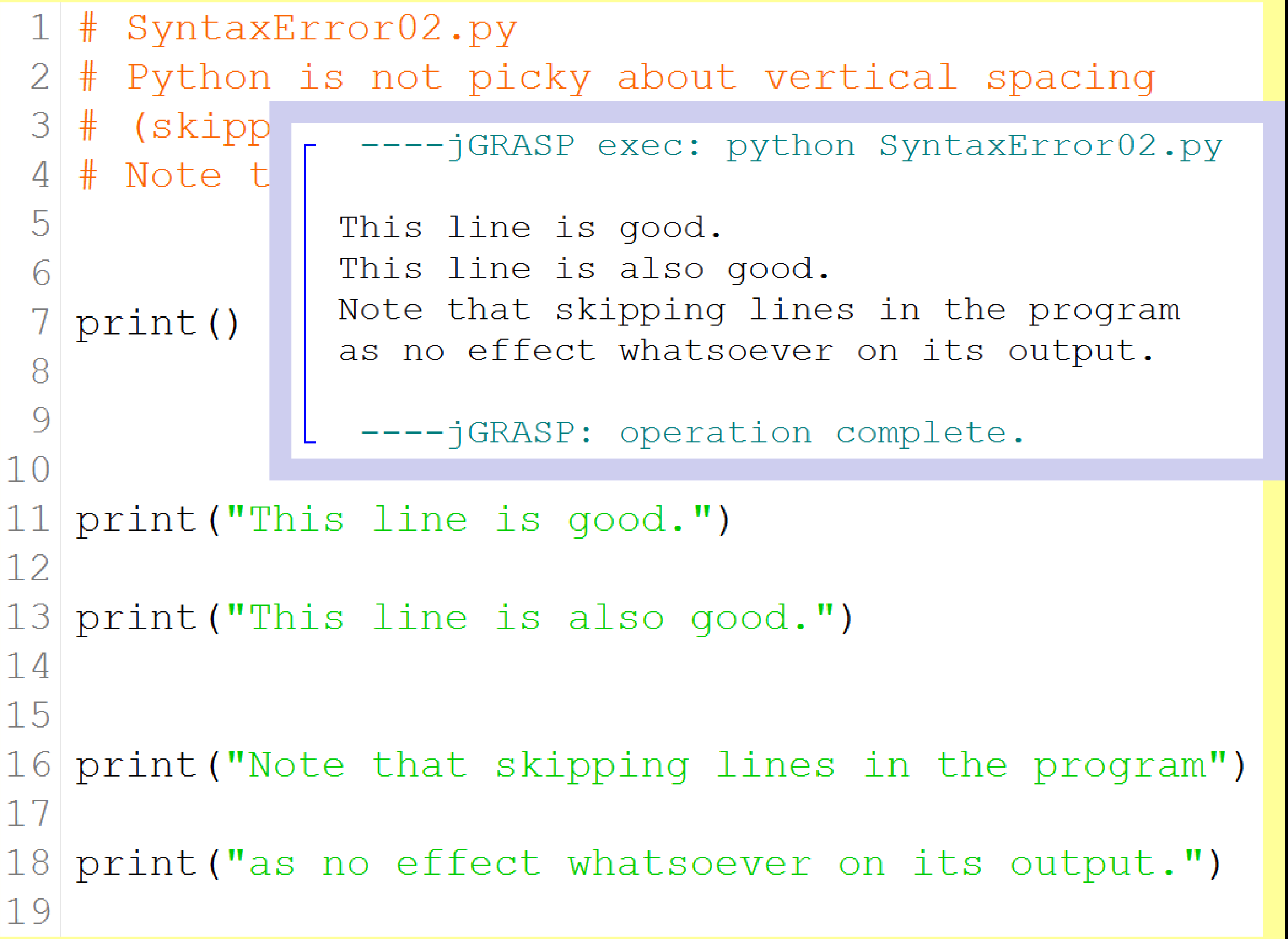
```
 1 # SyntaxError01.py
 2 # Python is very picky about horizontal spacing
 3 # (indenting). For now, all Python command need
 4 # to start on the left side of the screen with
 5 # no indenting at all.
 6
 7
 8 print("This line is good.")
 9  print("This line is bad.")
10
11
```

NOTE: The error message indicates what the error is and where it was found.

```
 ----jGRASP exec: python SyntaxError01.py
  File "SyntaxError01.py", line 9
    print("This line is bad.")
    ^
IndentationError: unexpected indent


 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

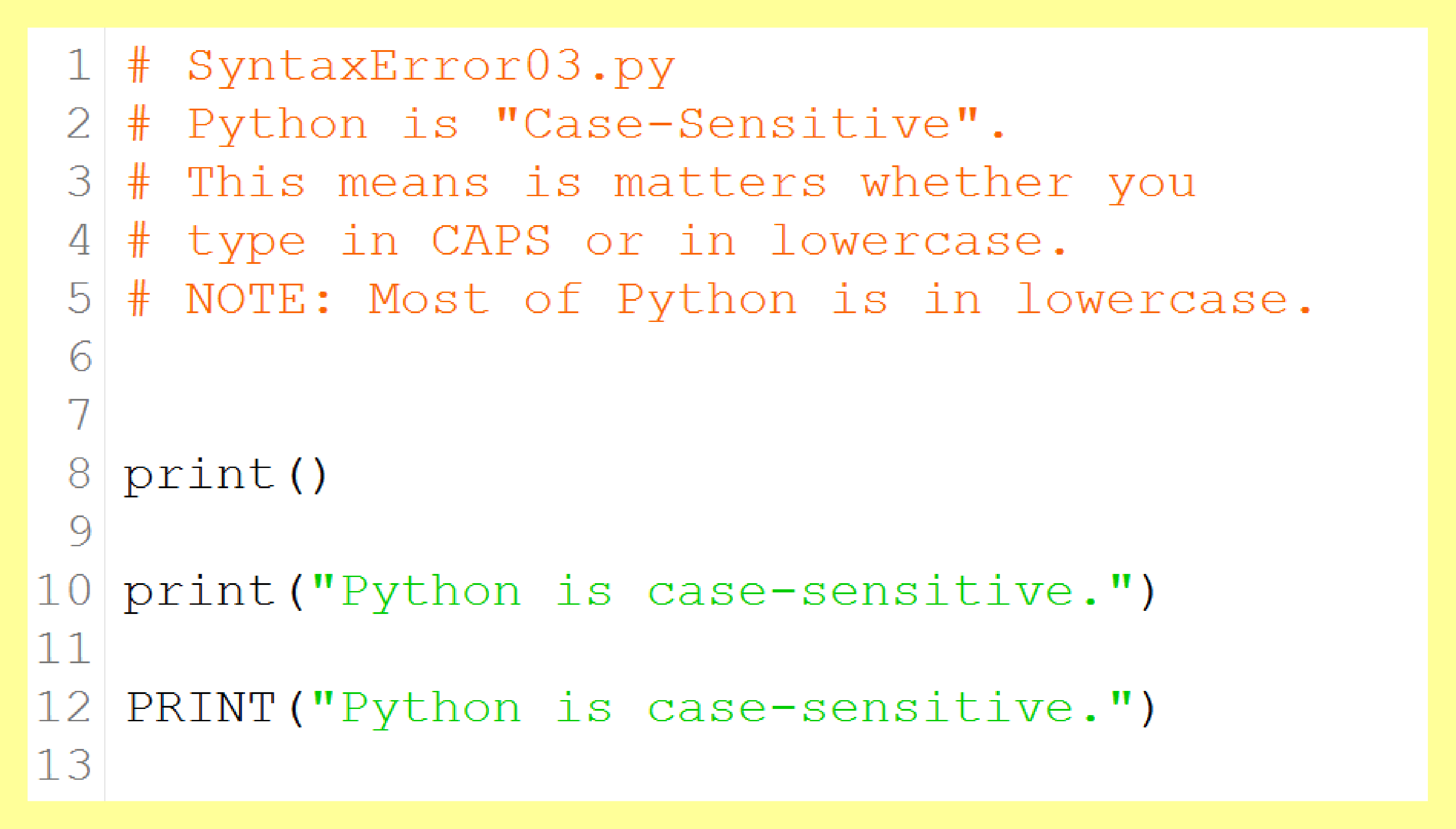


```python
# SyntaxError02.py
# Python is not picky about vertical spacing
# (skipping lines).
# Note that this has no effect on the output.


print()



print("This line is good.")

print("This line is also good.")


print("Note that skipping lines in the program")

print("as no effect whatsoever on its output.")

```

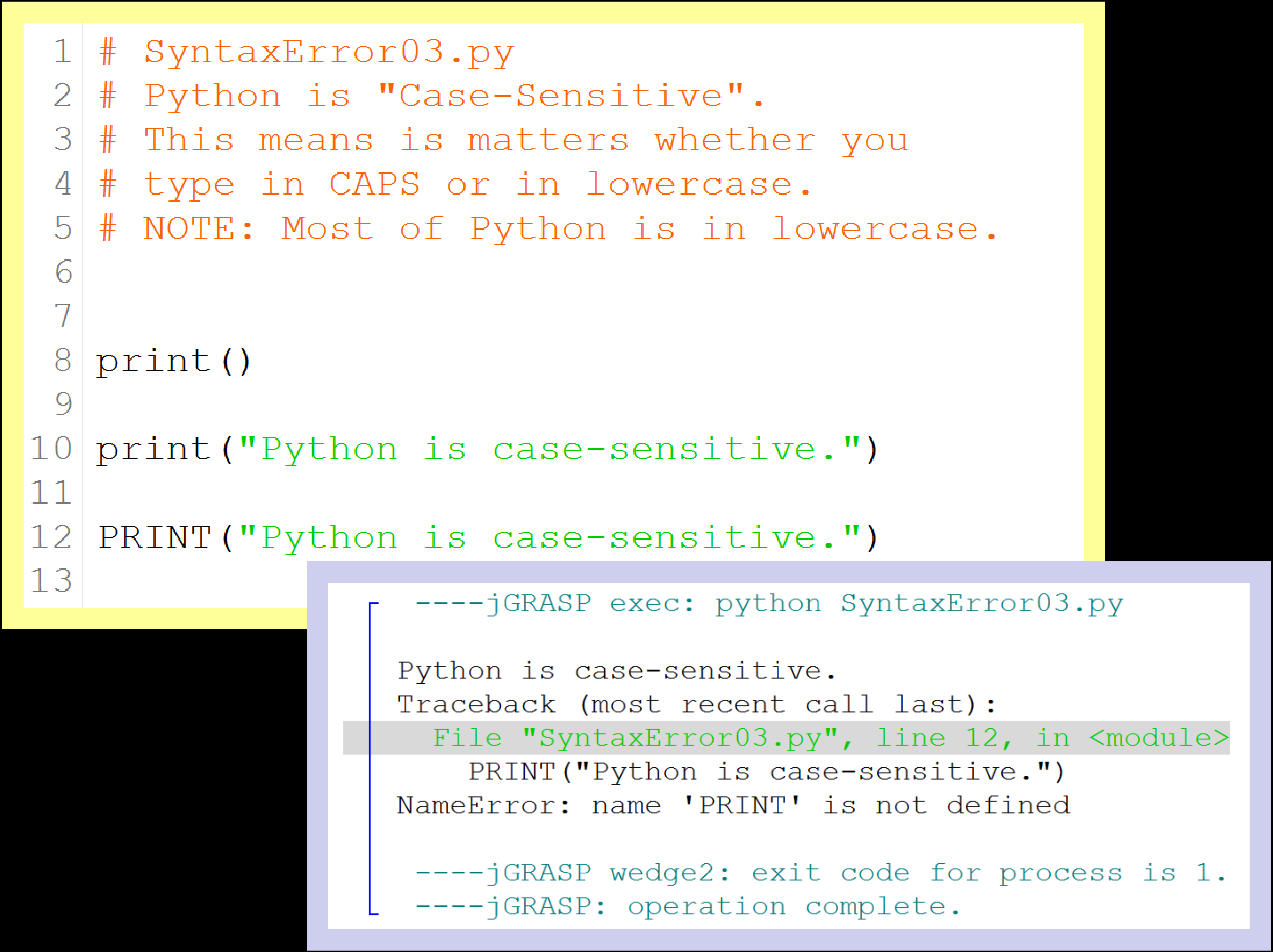


```
# SyntaxError02.py
# Python is not picky about vertical spacing
# (skipp
# Note t


print()



print("This line is good.")

print("This line is also good.")


print("Note that skipping lines in the program")

print("as no effect whatsoever on its output.")

```

```
 ----jGRASP exec: python SyntaxError02.py

This line is good.
This line is also good.
Note that skipping lines in the program
as no effect whatsoever on its output.

 ----jGRASP: operation complete.
```

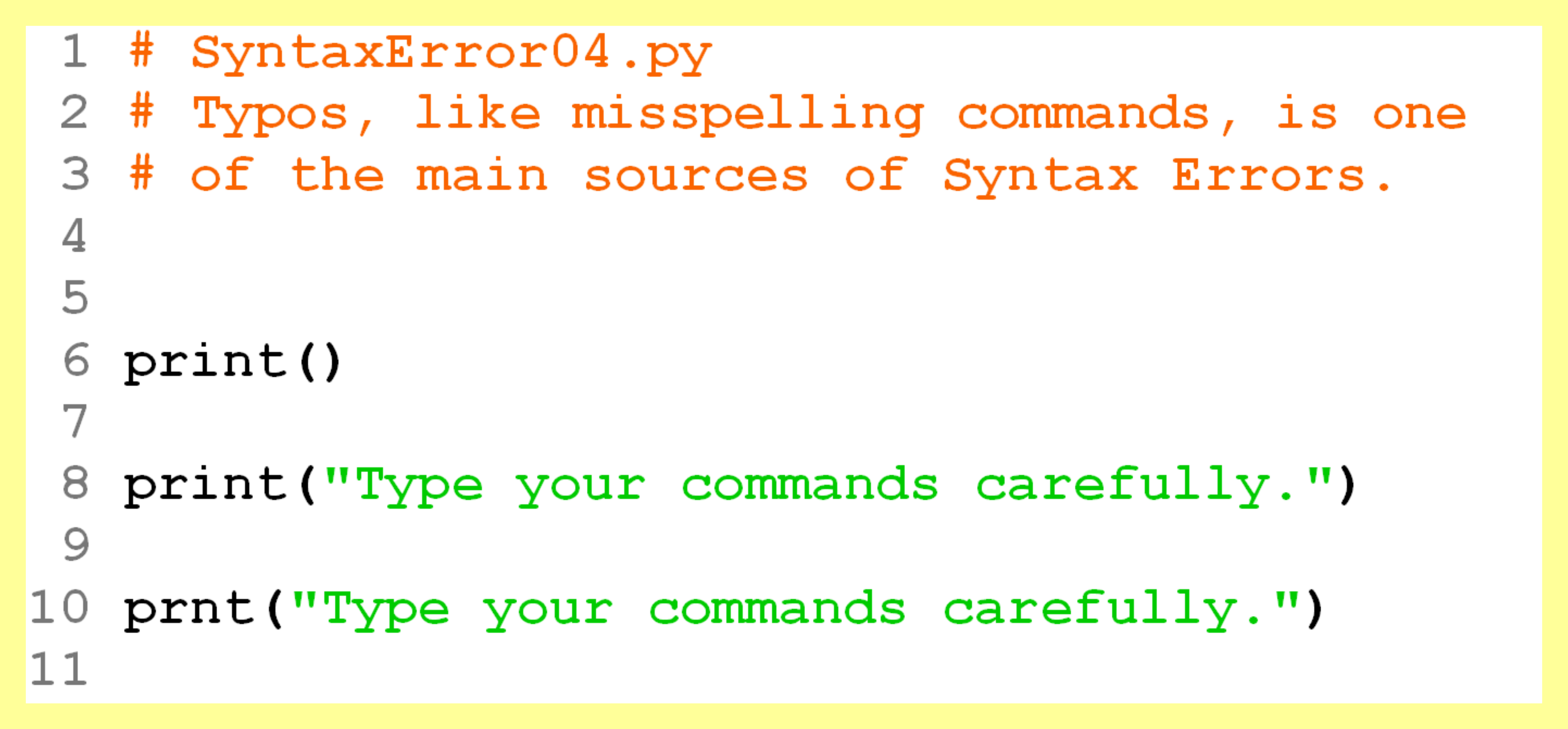


```
# SyntaxError03.py
# Python is "Case-Sensitive".
# This means is matters whether you
# type in CAPS or in lowercase.
# NOTE: Most of Python is in lowercase.


print()

print("Python is case-sensitive.")

PRINT("Python is case-sensitive.")

```

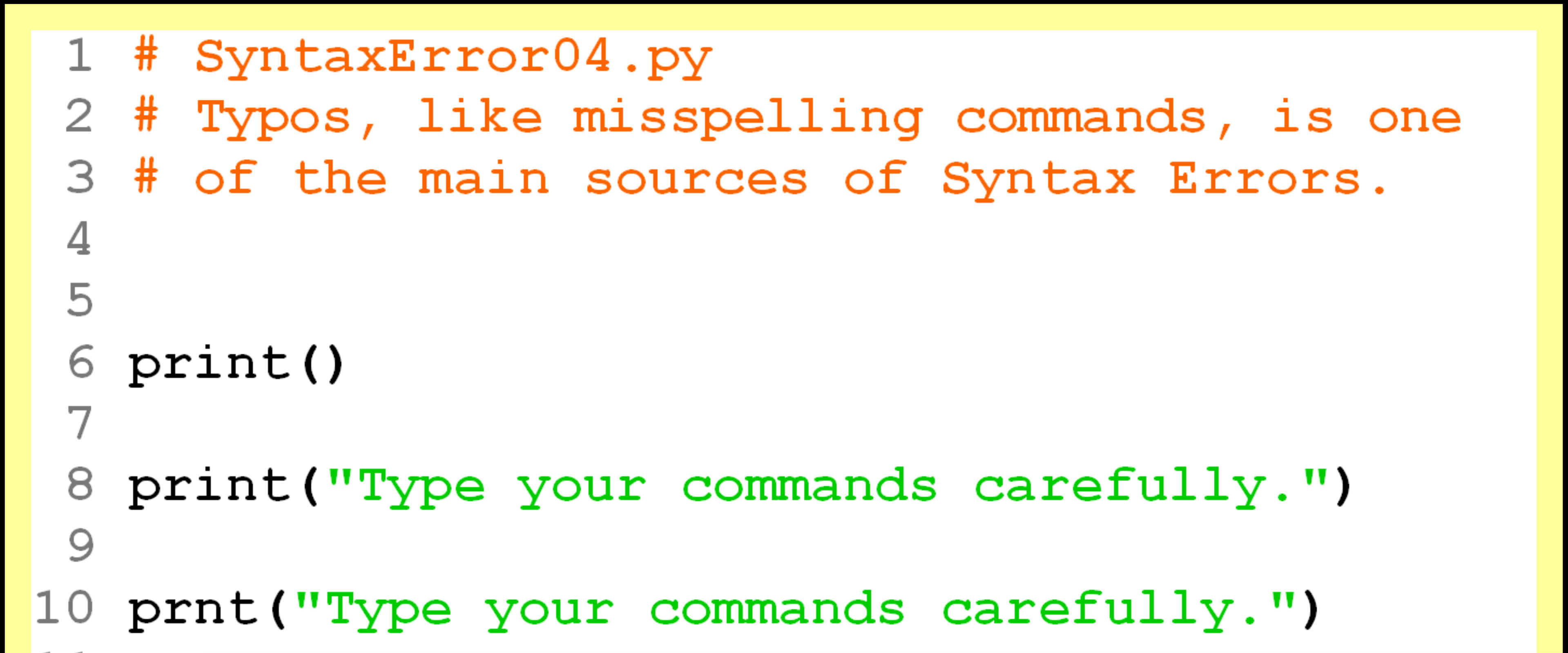


```
 1 # SyntaxError03.py
 2 # Python is "Case-Sensitive".
 3 # This means is matters whether you
 4 # type in CAPS or in lowercase.
 5 # NOTE: Most of Python is in lowercase.
 6
 7
 8 print()
 9
10 print("Python is case-sensitive.")
11
12 PRINT("Python is case-sensitive.")
13
```

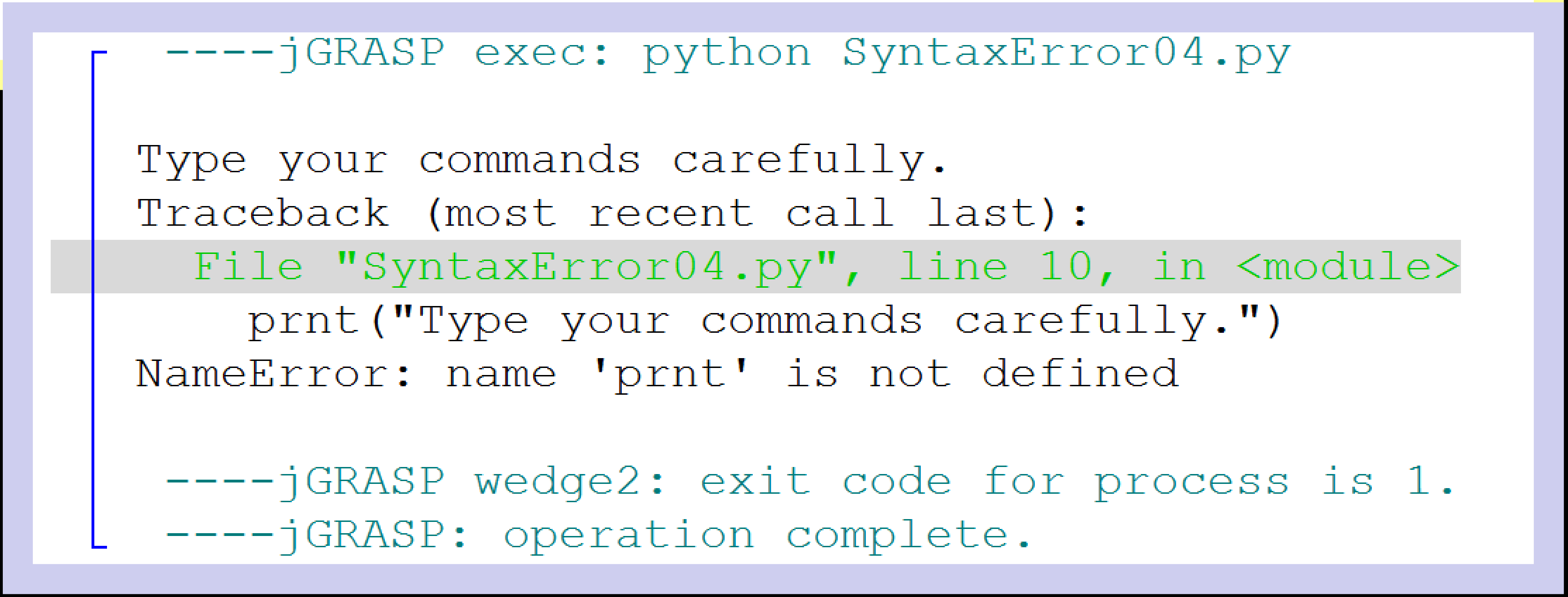
```
 ----jGRASP exec: python SyntaxError03.py

Python is case-sensitive.
Traceback (most recent call last):
  File "SyntaxError03.py", line 12, in <module>
    PRINT("Python is case-sensitive.")
NameError: name 'PRINT' is not defined

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



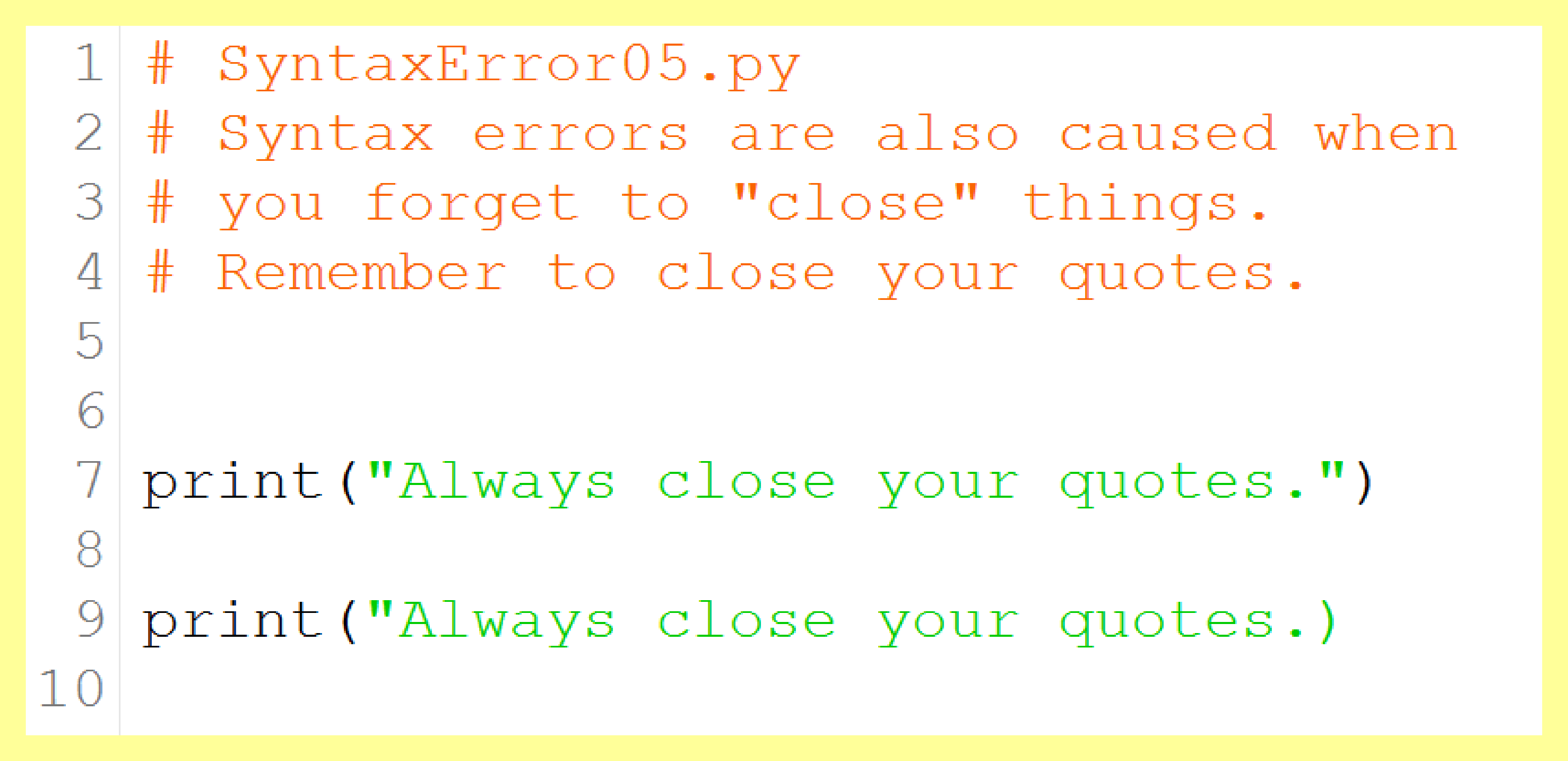
```python
# SyntaxError04.py
# Typos, like misspelling commands, is one
# of the main sources of Syntax Errors.


print()

print("Type your commands carefully.")

prnt("Type your commands carefully.")

```



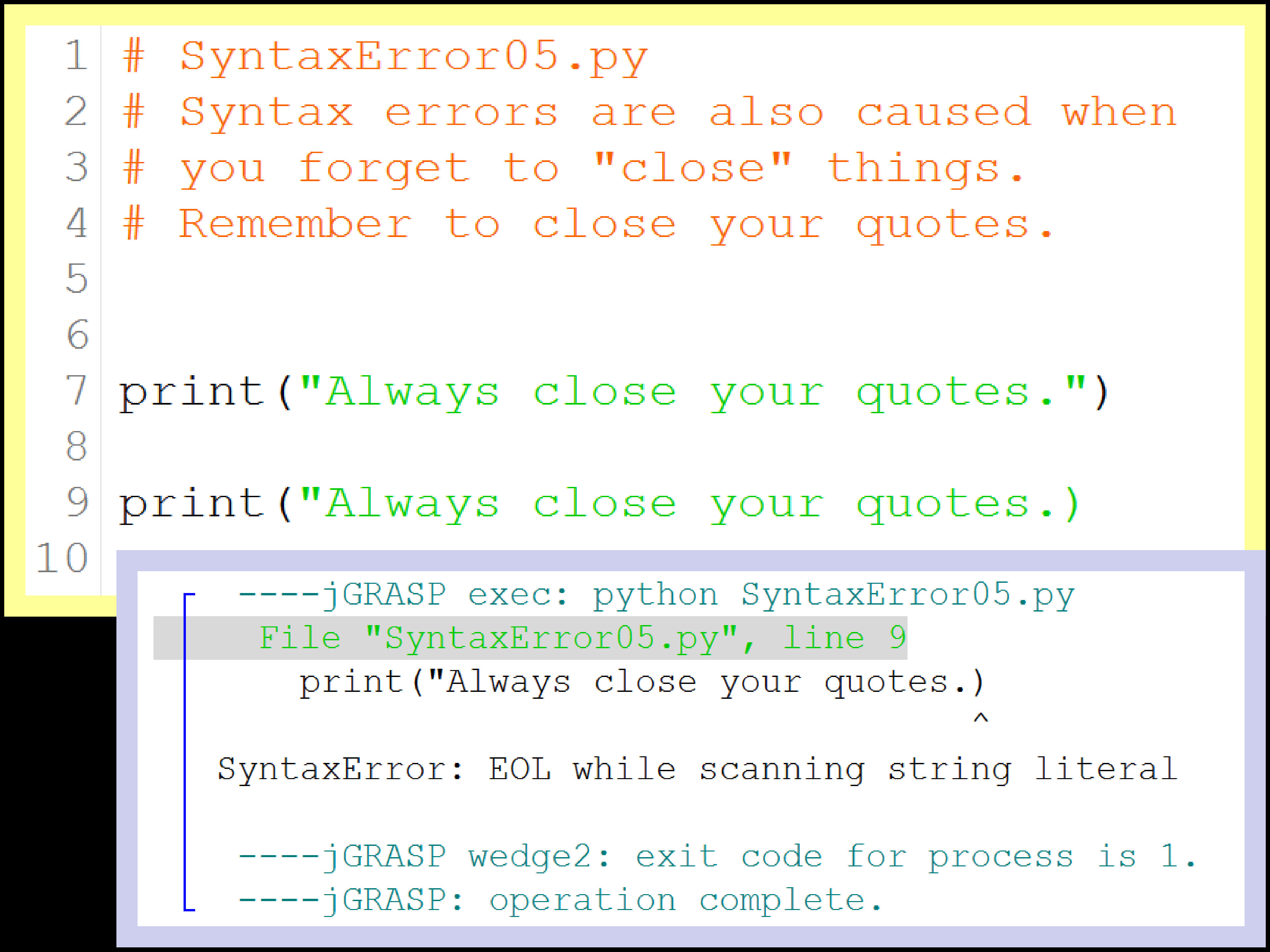
```python
# SyntaxError04.py
# Typos, like misspelling commands, is one
# of the main sources of Syntax Errors.


print()

print("Type your commands carefully.")

prnt("Type your commands carefully.")

```



```
 ----jGRASP exec: python SyntaxError04.py

Type your commands carefully.
Traceback (most recent call last):
  File "SyntaxError04.py", line 10, in <module>
    prnt("Type your commands carefully.")
NameError: name 'prnt' is not defined

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

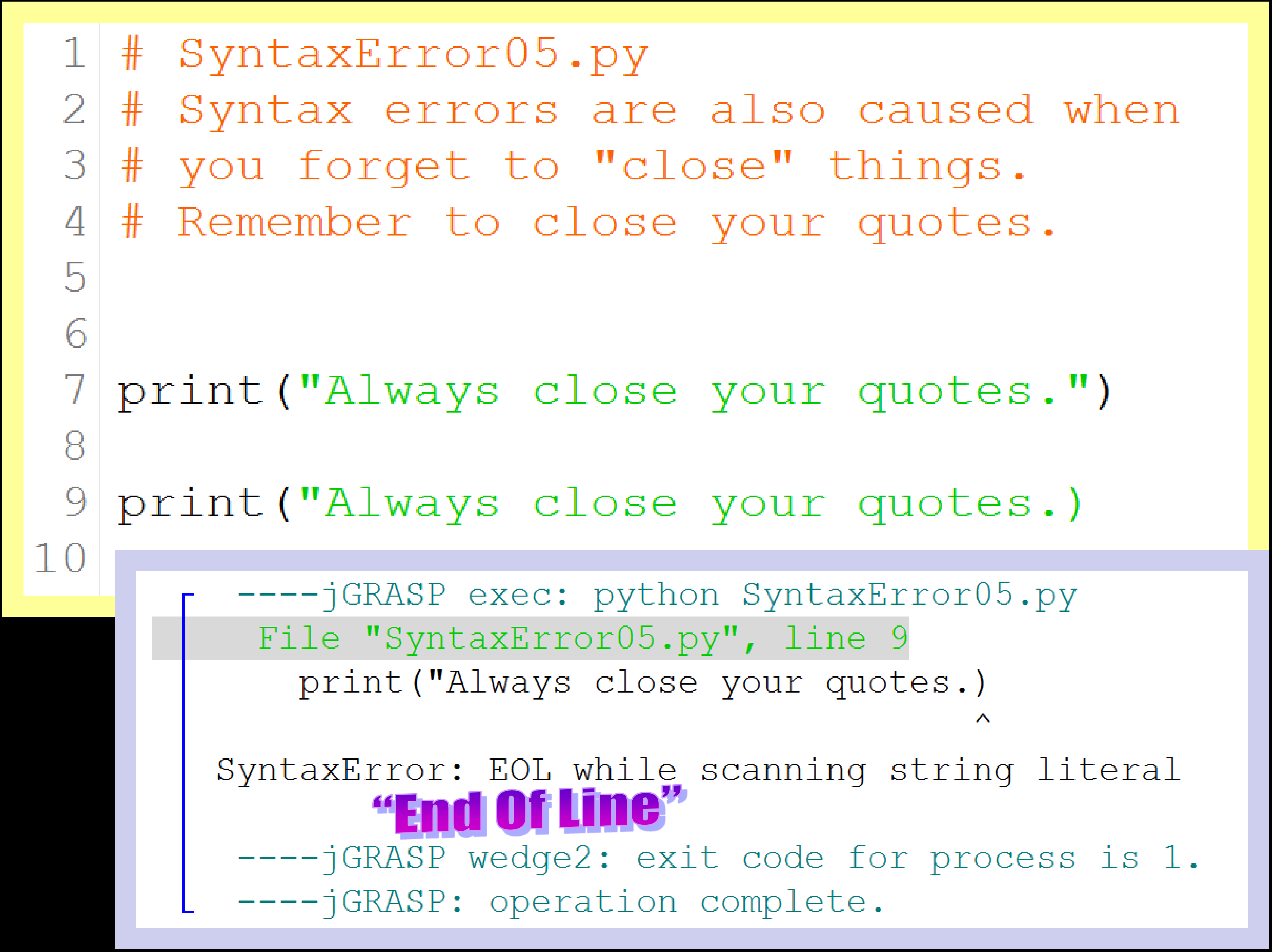


```
# SyntaxError05.py
# Syntax errors are also caused when
# you forget to "close" things.
# Remember to close your quotes.


print("Always close your quotes.")

print("Always close your quotes.)

```



```
 1  # SyntaxError05.py
 2  # Syntax errors are also caused when
 3  # you forget to "close" things.
 4  # Remember to close your quotes.
 5
 6
 7  print("Always close your quotes.")
 8
 9  print("Always close your quotes.)
10
```

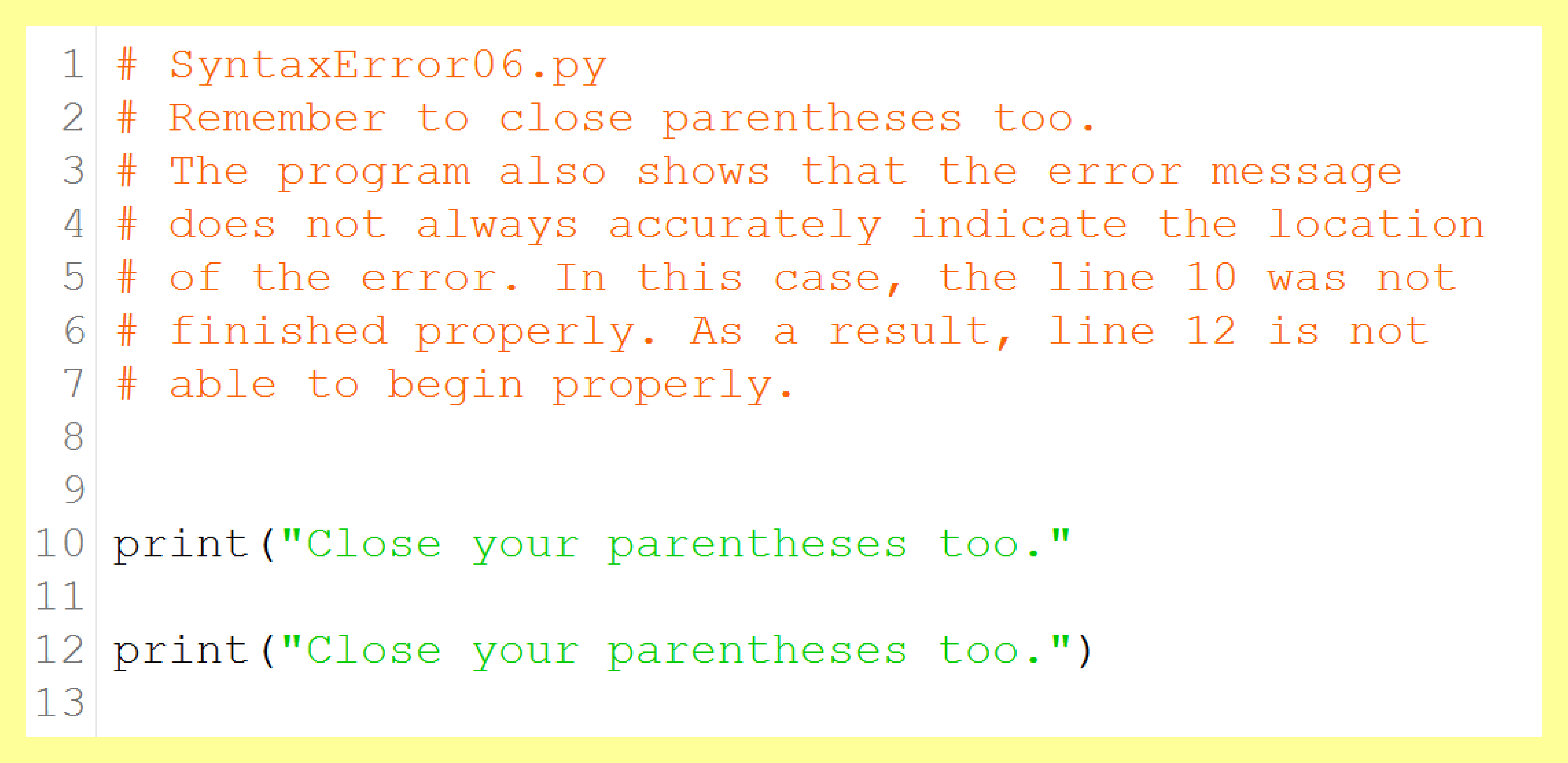
```
 ----jGRASP exec: python SyntaxError05.py
  File "SyntaxError05.py", line 9
    print("Always close your quotes.)
                                    ^
SyntaxError: EOL while scanning string literal


 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



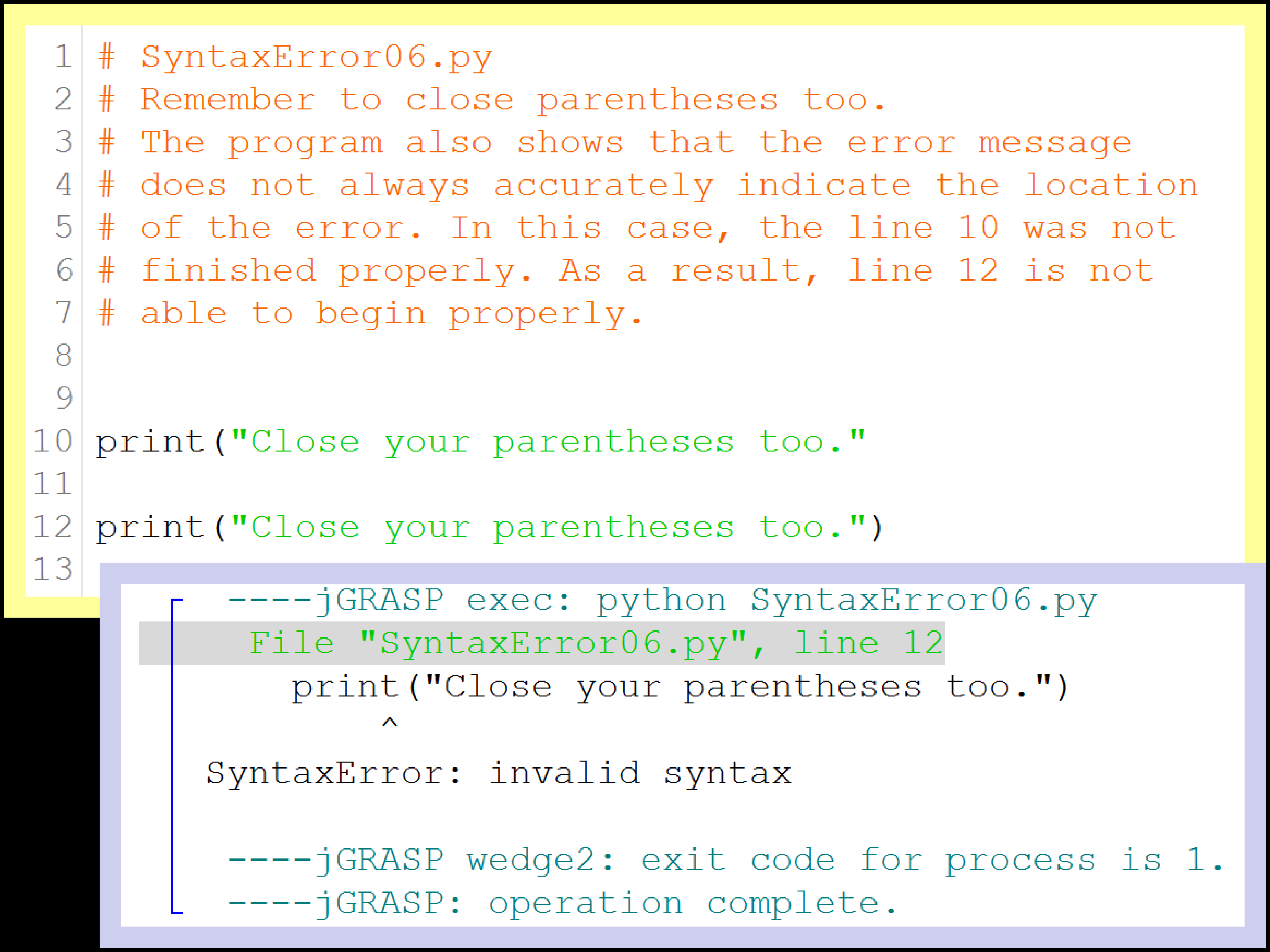
```
 1 # SyntaxError05.py
 2 # Syntax errors are also caused when
 3 # you forget to "close" things.
 4 # Remember to close your quotes.
 5
 6
 7 print("Always close your quotes.")
 8
 9 print("Always close your quotes.)
10
```

```
 ----jGRASP exec: python SyntaxError05.py
  File "SyntaxError05.py", line 9
    print("Always close your quotes.)
                                    ^
SyntaxError: EOL while scanning string literal

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

“End Of Line”

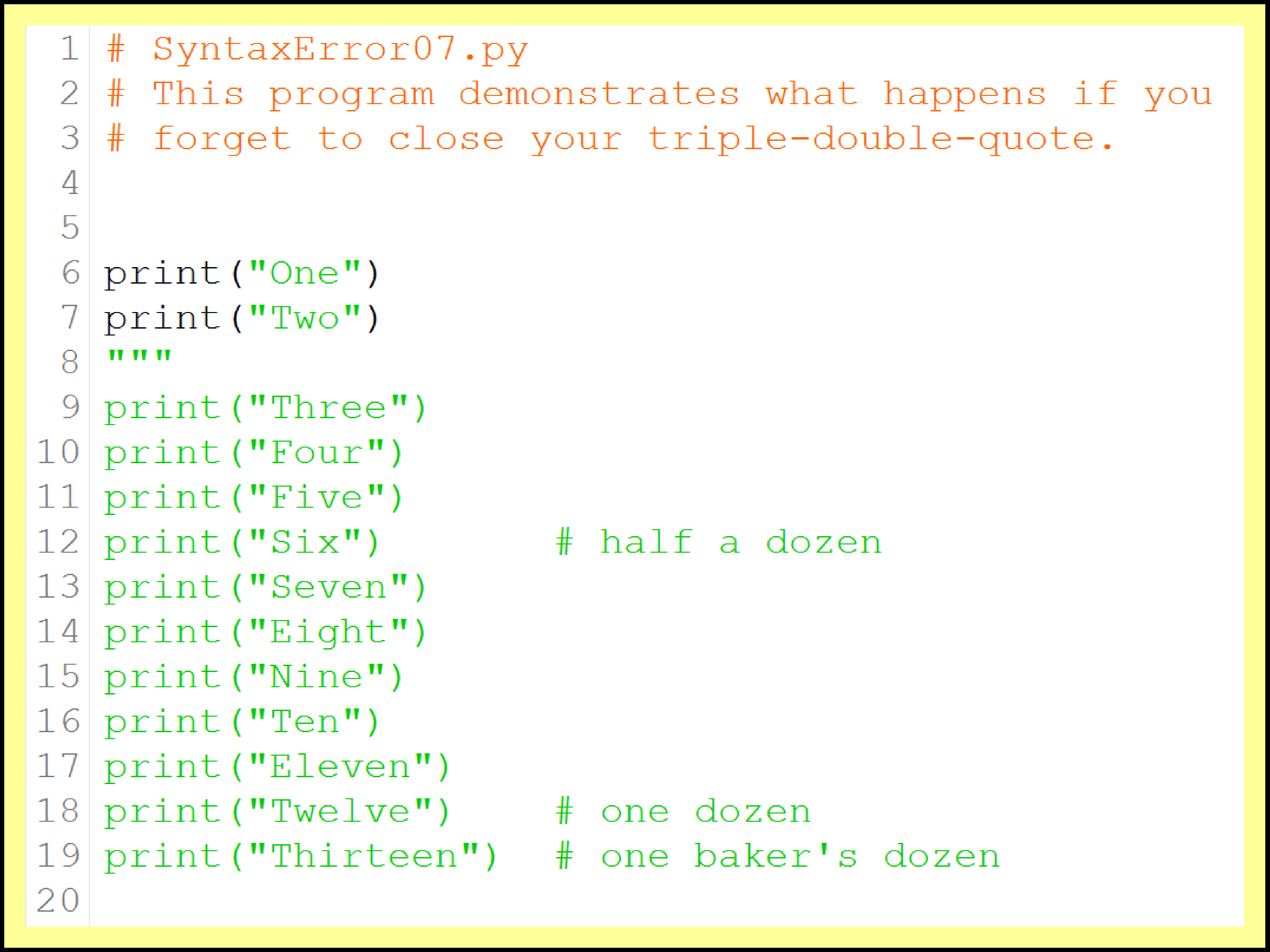


```
 1 # SyntaxError06.py
 2 # Remember to close parentheses too.
 3 # The program also shows that the error message
 4 # does not always accurately indicate the location
 5 # of the error. In this case, the line 10 was not
 6 # finished properly. As a result, line 12 is not
 7 # able to begin properly.
 8
 9
10 print("Close your parentheses too."
11
12 print("Close your parentheses too.")
13
```

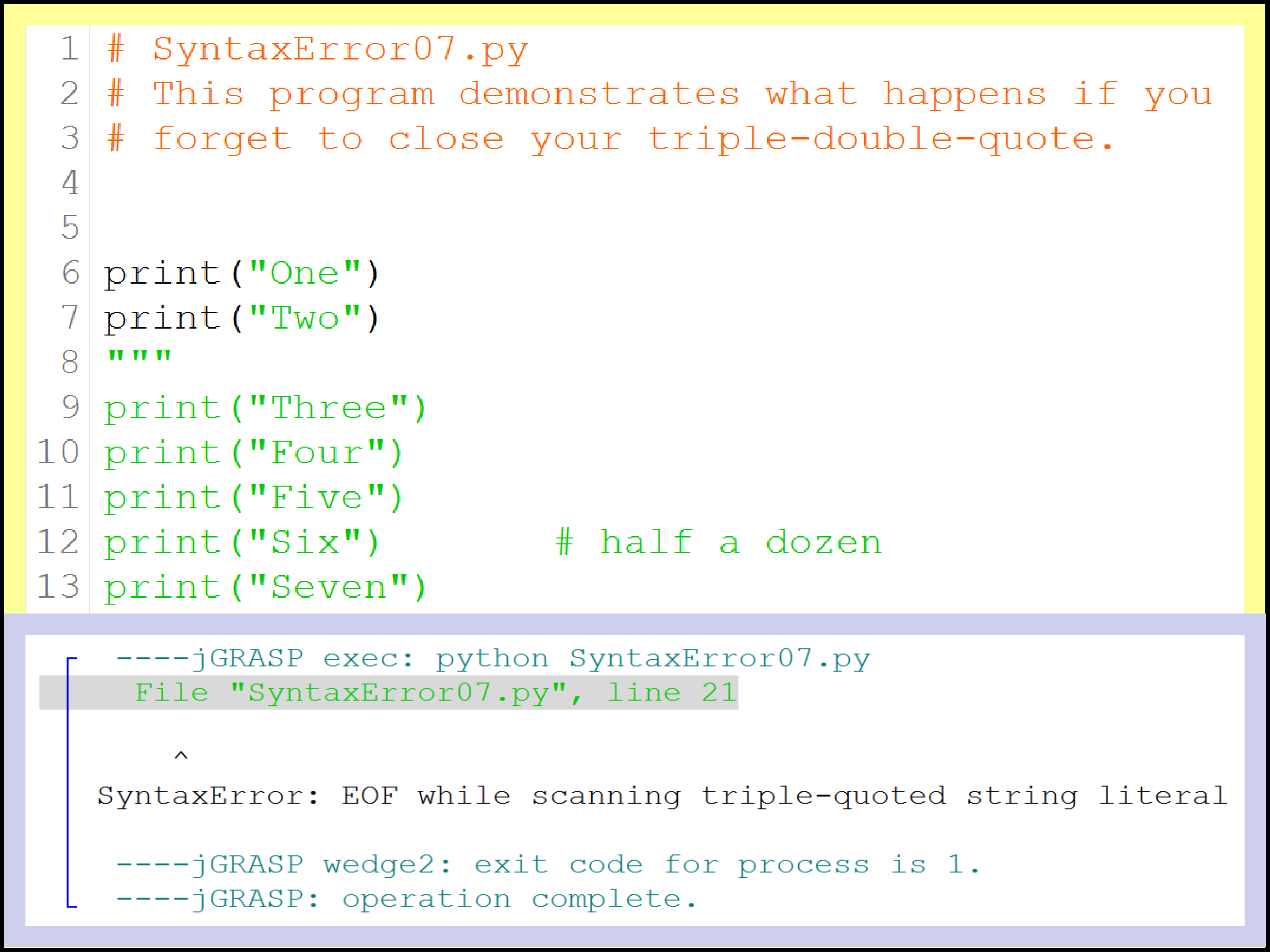


```
 1 # SyntaxError06.py
 2 # Remember to close parentheses too.
 3 # The program also shows that the error message
 4 # does not always accurately indicate the location
 5 # of the error. In this case, the line 10 was not
 6 # finished properly. As a result, line 12 is not
 7 # able to begin properly.
 8
 9
10 print("Close your parentheses too."
11
12 print("Close your parentheses too.")
13
```

```
 ----jGRASP exec: python SyntaxError06.py
  File "SyntaxError06.py", line 12
    print("Close your parentheses too.")
        ^
SyntaxError: invalid syntax

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

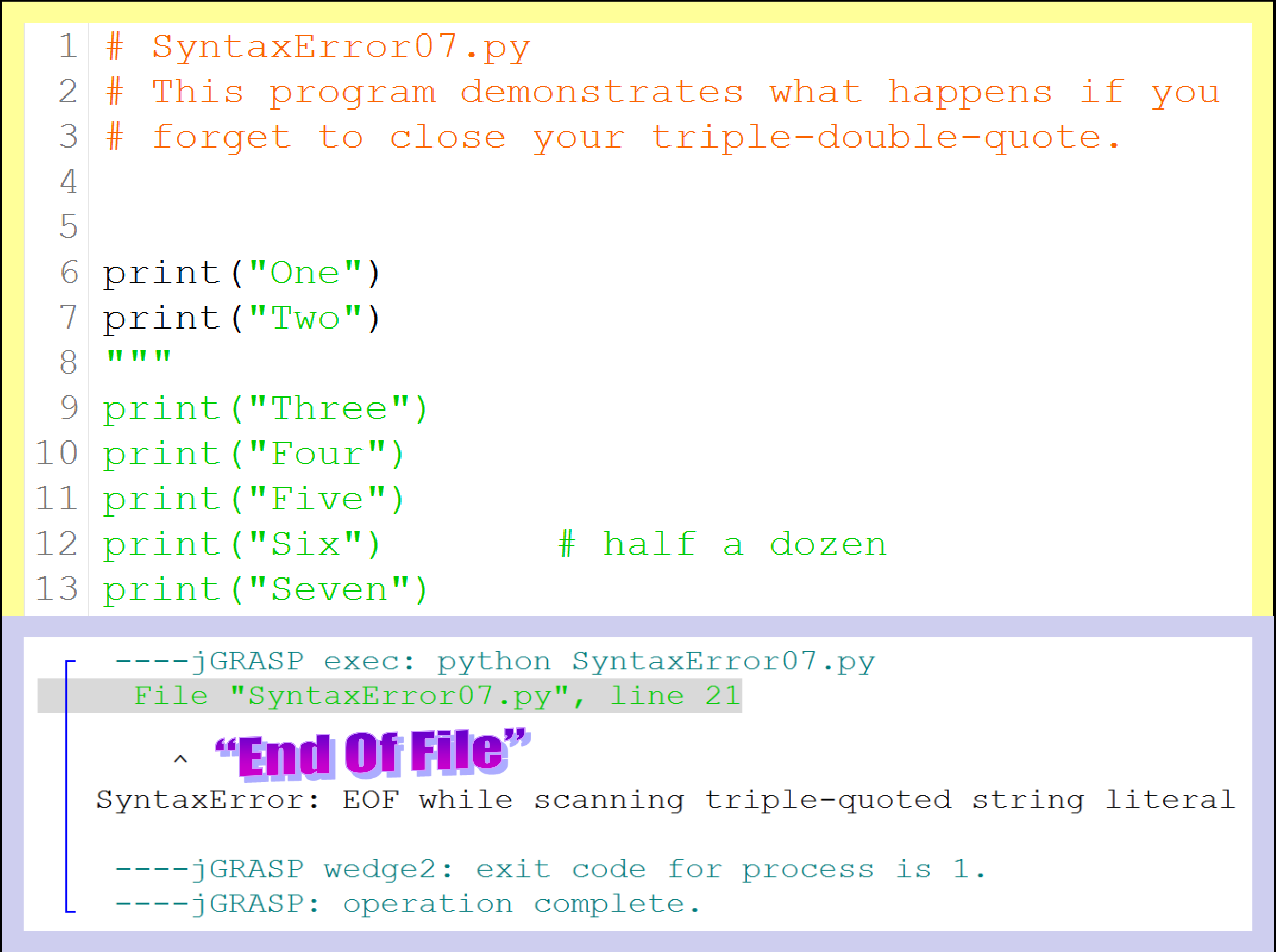


```python
# SyntaxError07.py
# This program demonstrates what happens if you
# forget to close your triple-double-quote.


print("One")
print("Two")
"""
print("Three")
print("Four")
print("Five")
print("Six")       # half a dozen
print("Seven")
print("Eight")
print("Nine")
print("Ten")
print("Eleven")
print("Twelve")    # one dozen
print("Thirteen")  # one baker's dozen

```



```
# SyntaxError07.py
# This program demonstrates what happens if you
# forget to close your triple-double-quote.


print("One")
print("Two")
"""
print("Three")
print("Four")
print("Five")
print("Six")          # half a dozen
print("Seven")
```

```
 ----jGRASP exec: python SyntaxError07.py
  File "SyntaxError07.py", line 21

    ^
SyntaxError: EOF while scanning triple-quoted string literal

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



```python
# SyntaxError07.py
# This program demonstrates what happens if you
# forget to close your triple-double-quote.


print("One")
print("Two")
"""
print("Three")
print("Four")
print("Five")
print("Six")       # half a dozen
print("Seven")
```

```
 ----jGRASP exec: python SyntaxError07.py
  File "SyntaxError07.py", line 21

    ^
SyntaxError: EOF while scanning triple-quoted string literal

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

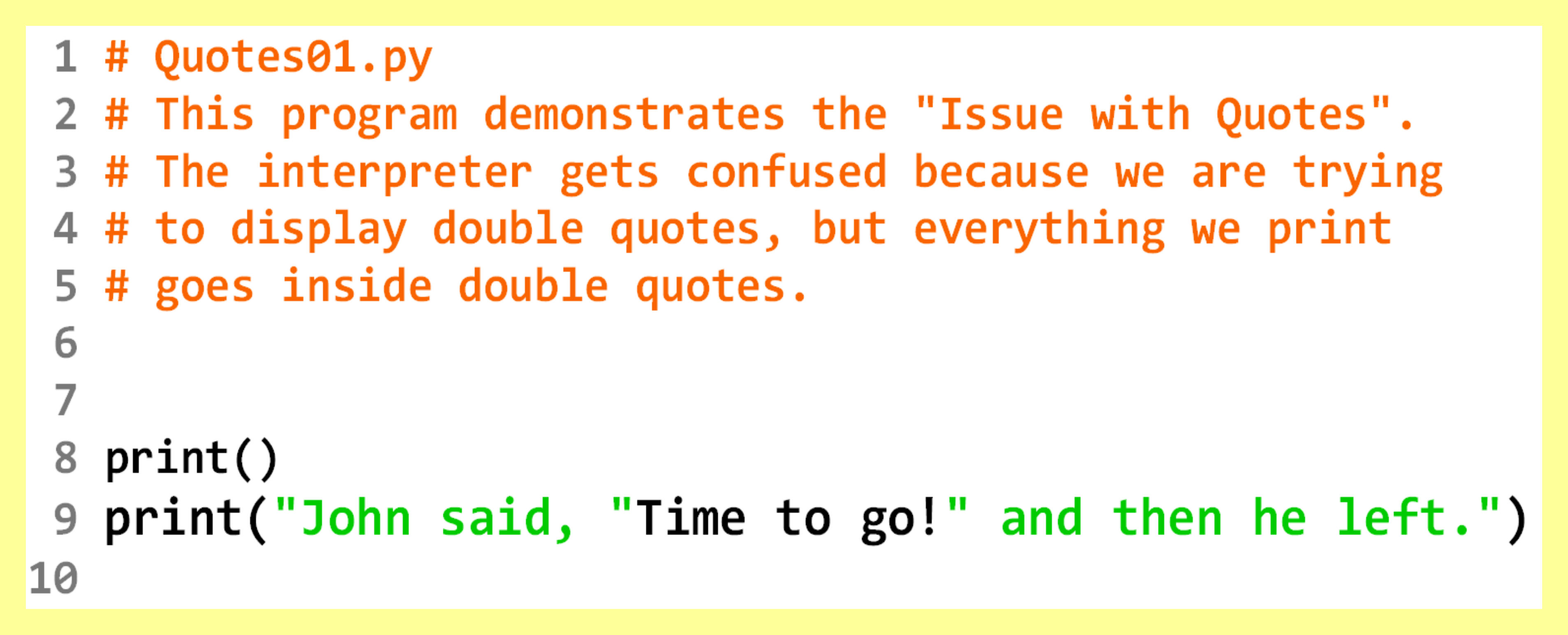
“End Of File”

Computer Science Commandment

Anything that thou open,
thou shalt also close!

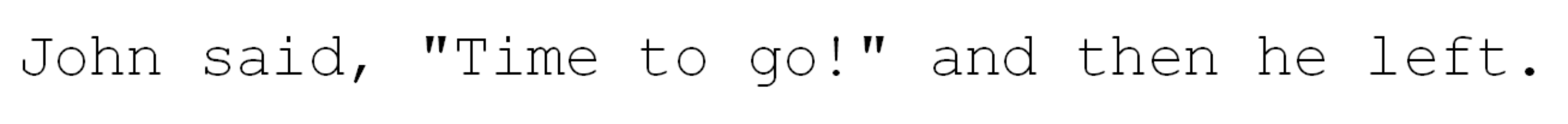
Section 3.6

The Issue with Quotes

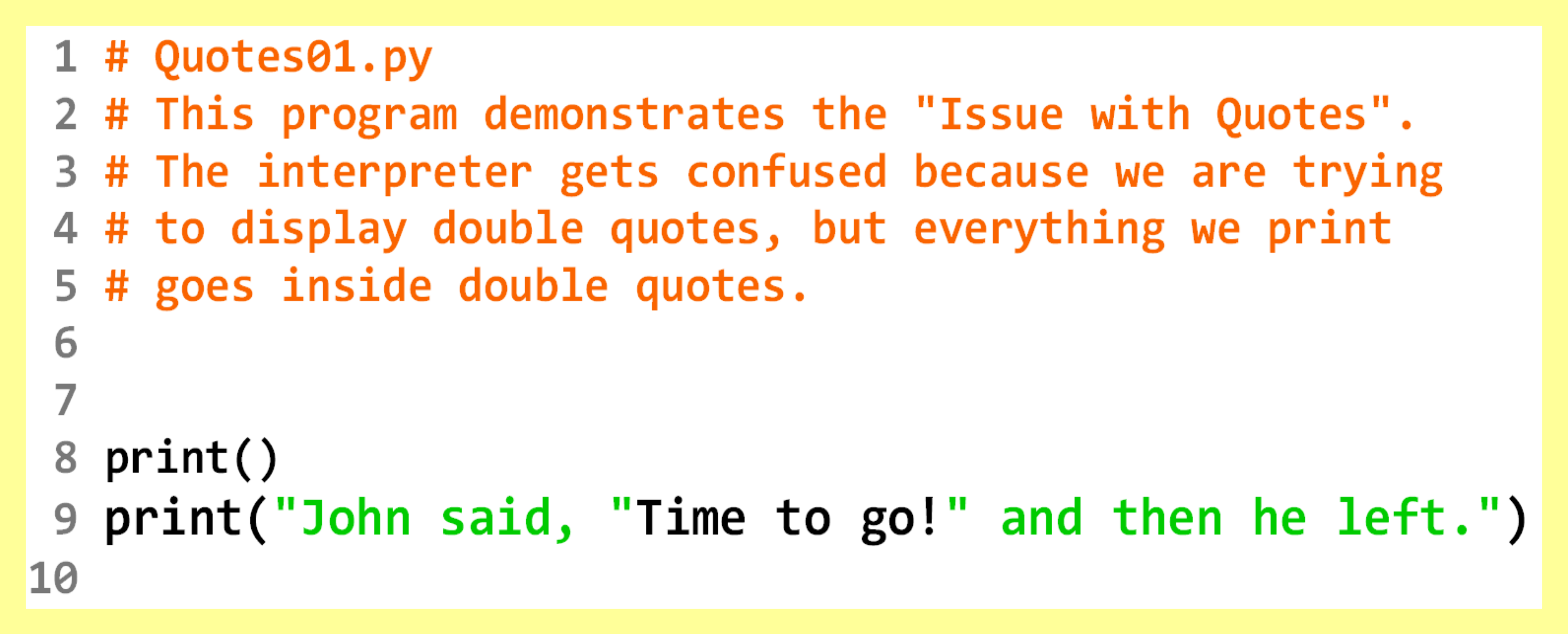


```python
# Quotes01.py
# This program demonstrates the "Issue with Quotes".
# The interpreter gets confused because we are trying
# to display double quotes, but everything we print
# goes inside double quotes.


print()
print("John said, "Time to go!" and then he left.")

```

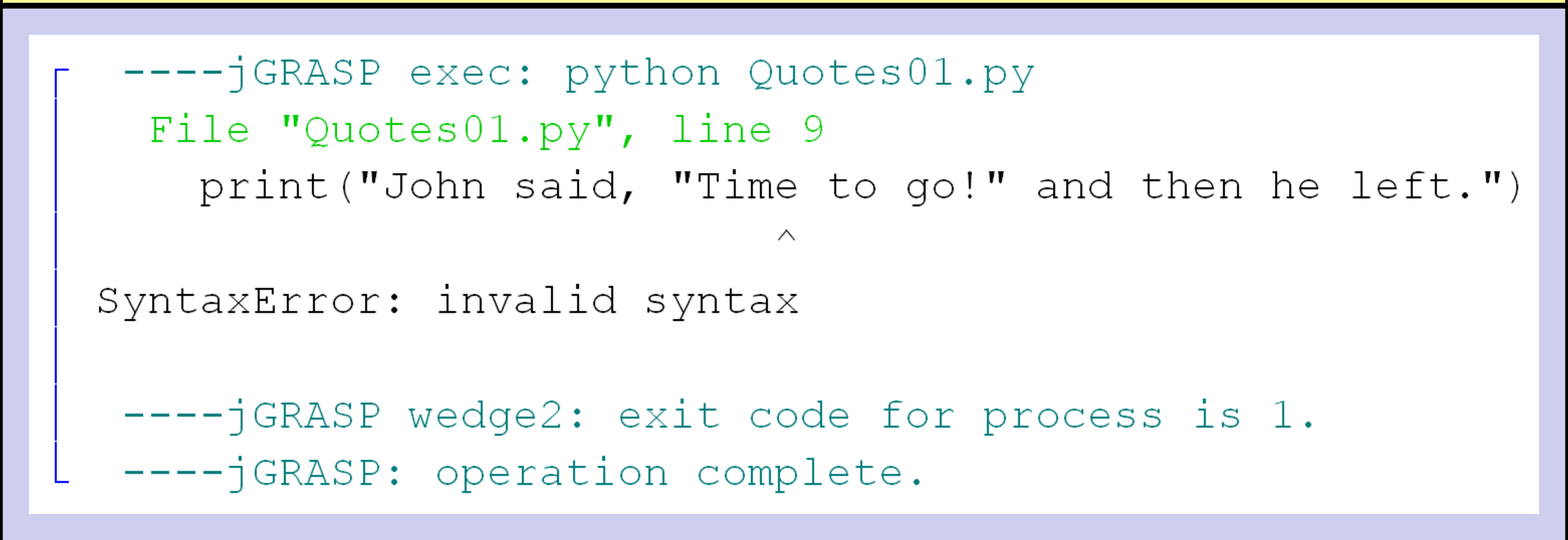
This is the sentence that I wish to display:



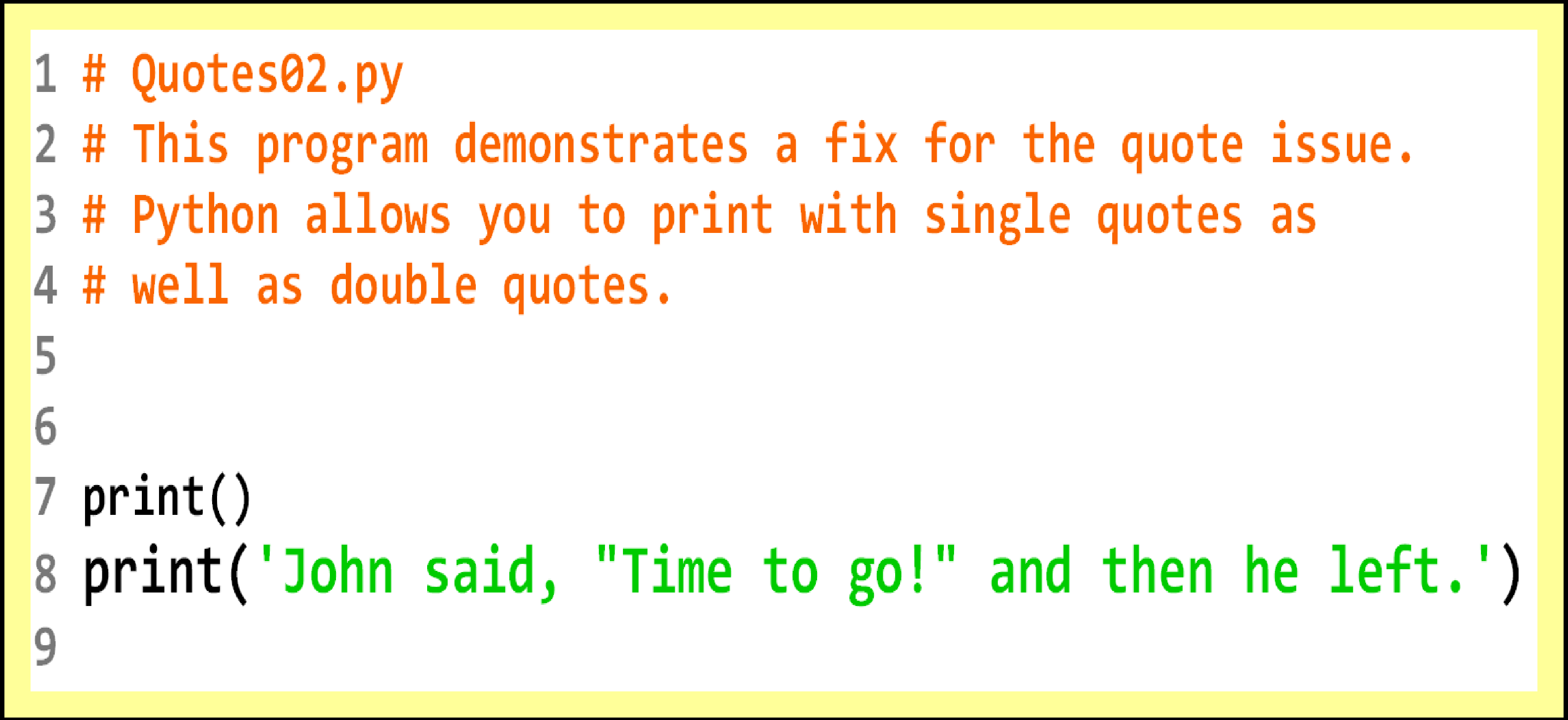
```
John said, "Time to go!" and then he left.
```



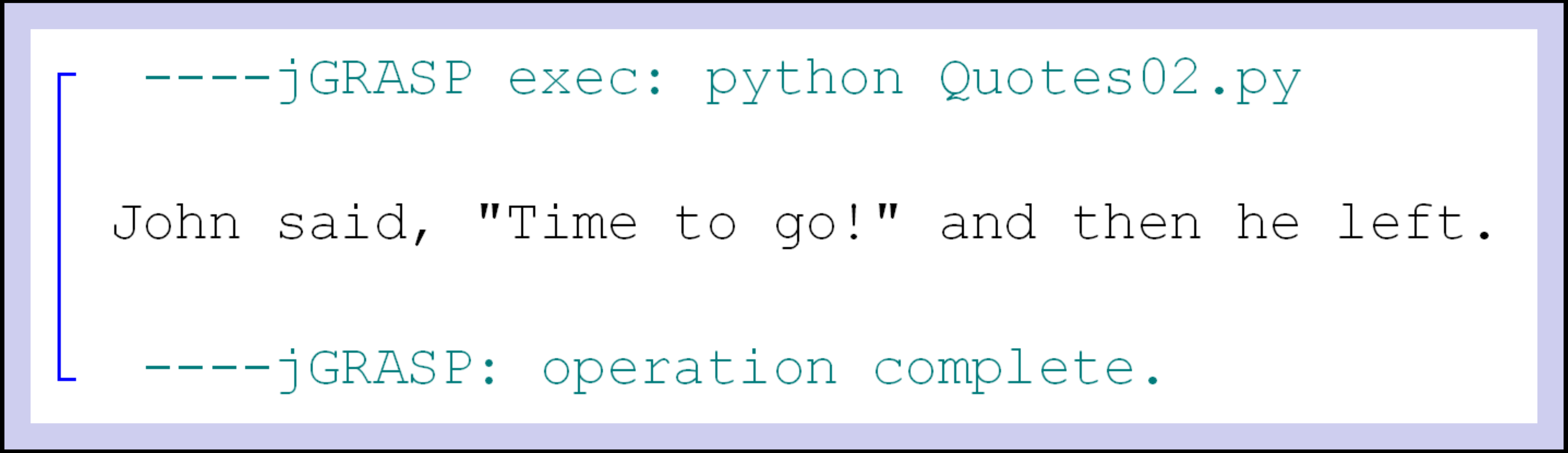
```
 1 # Quotes01.py
 2 # This program demonstrates the "Issue with Quotes".
 3 # The interpreter gets confused because we are trying
 4 # to display double quotes, but everything we print
 5 # goes inside double quotes.
 6
 7
 8 print()
 9 print("John said, "Time to go!" and then he left.")
10
```



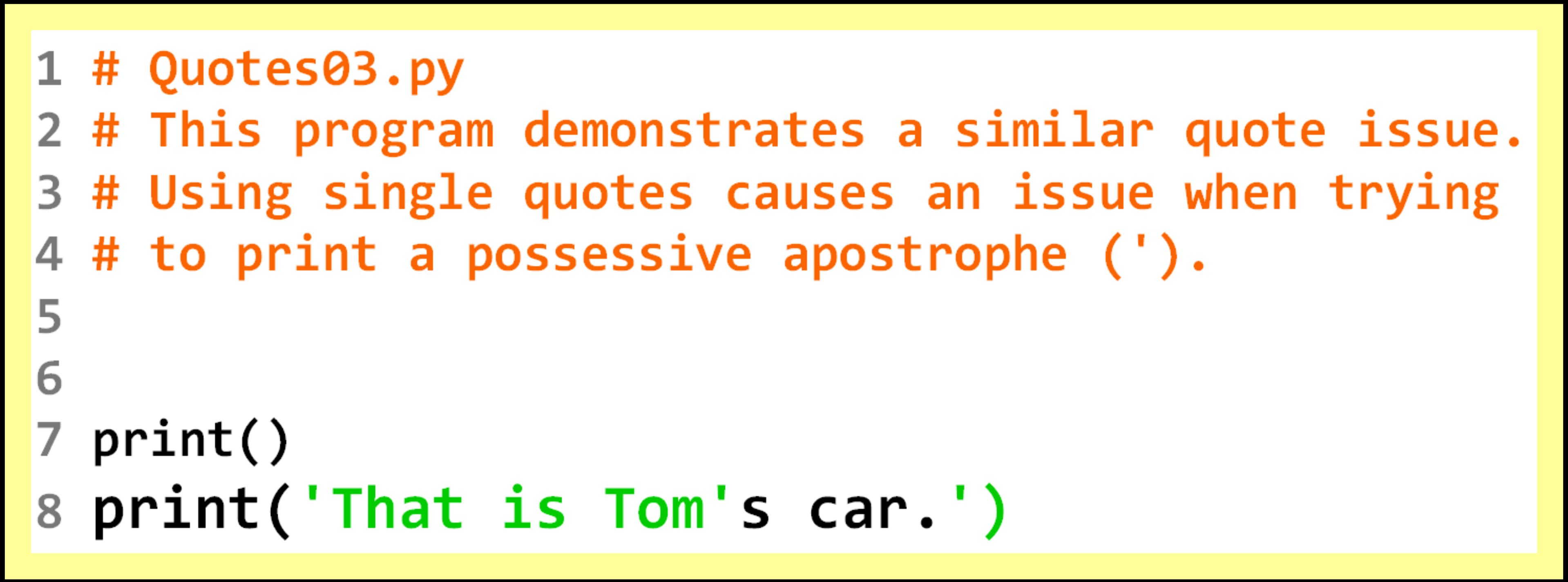
```
 ----jGRASP exec: python Quotes01.py
  File "Quotes01.py", line 9
    print("John said, "Time to go!" and then he left.")
                          ^
SyntaxError: invalid syntax


 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



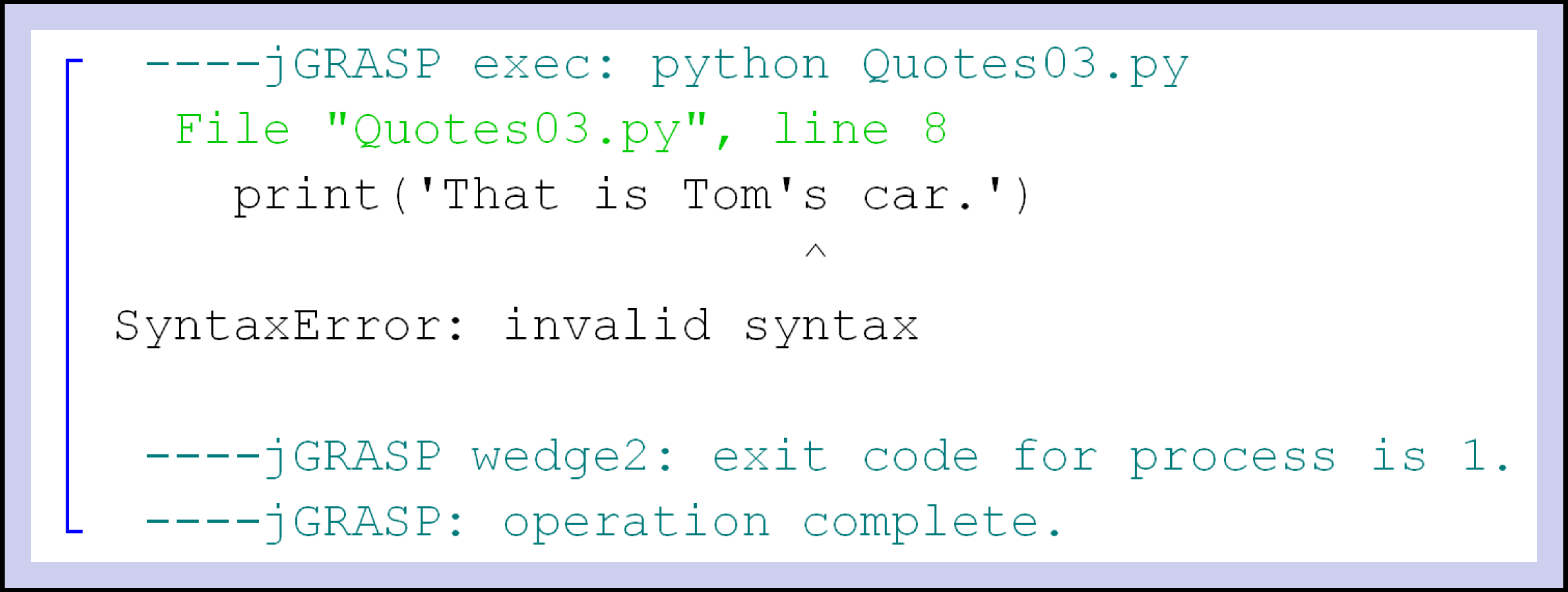
```python
# Quotes02.py
# This program demonstrates a fix for the quote issue.
# Python allows you to print with single quotes as
# well as double quotes.


print()
print('John said, "Time to go!" and then he left.')

```



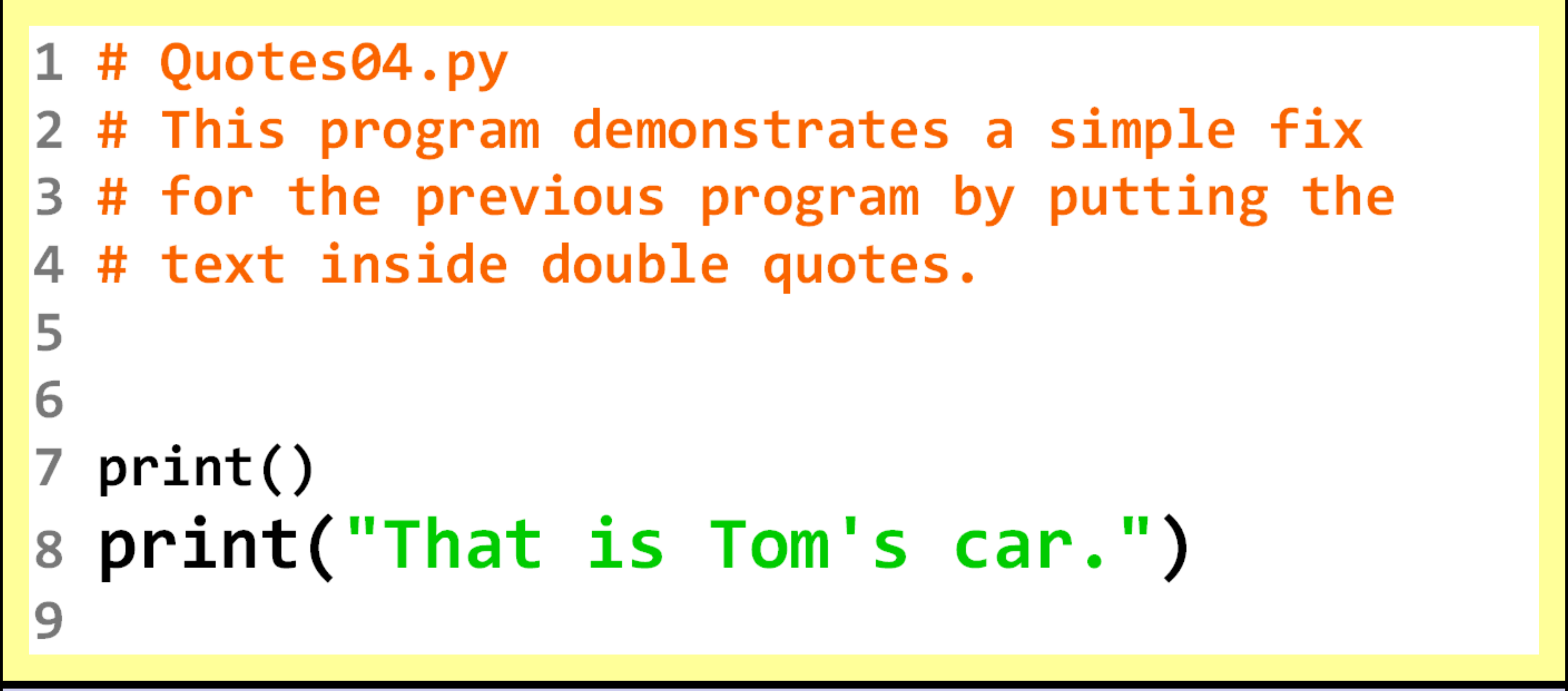
```
 ----jGRASP exec: python Quotes02.py

John said, "Time to go!" and then he left.

 ----jGRASP: operation complete.
```



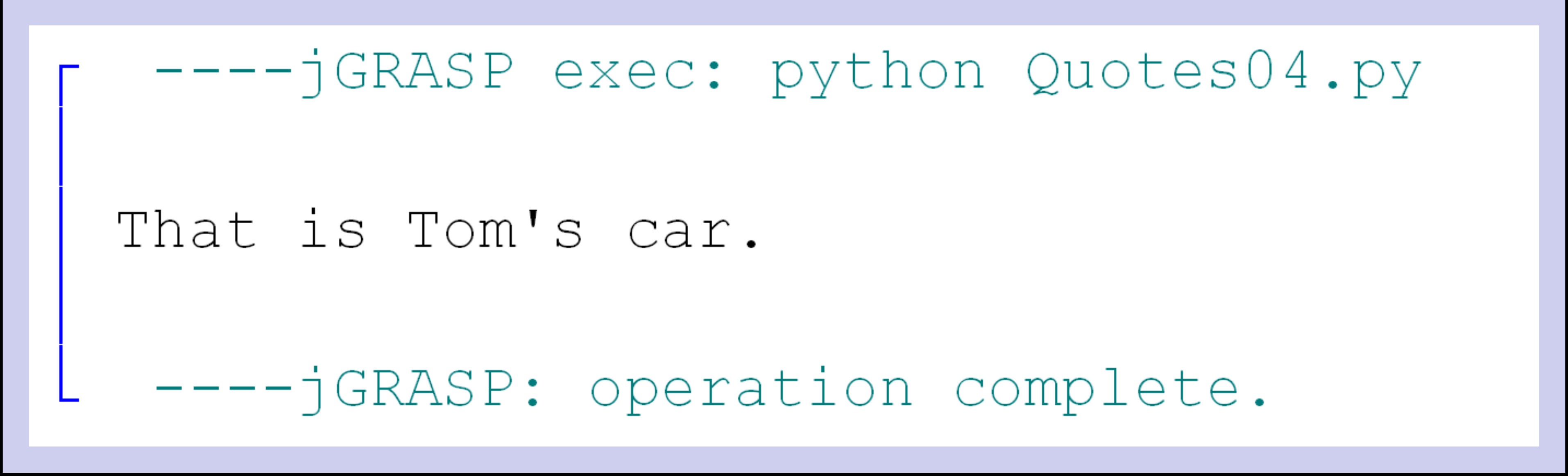
```
1 # Quotes03.py
2 # This program demonstrates a similar quote issue.
3 # Using single quotes causes an issue when trying
4 # to print a possessive apostrophe (').
5
6
7 print()
8 print('That is Tom's car.')
```



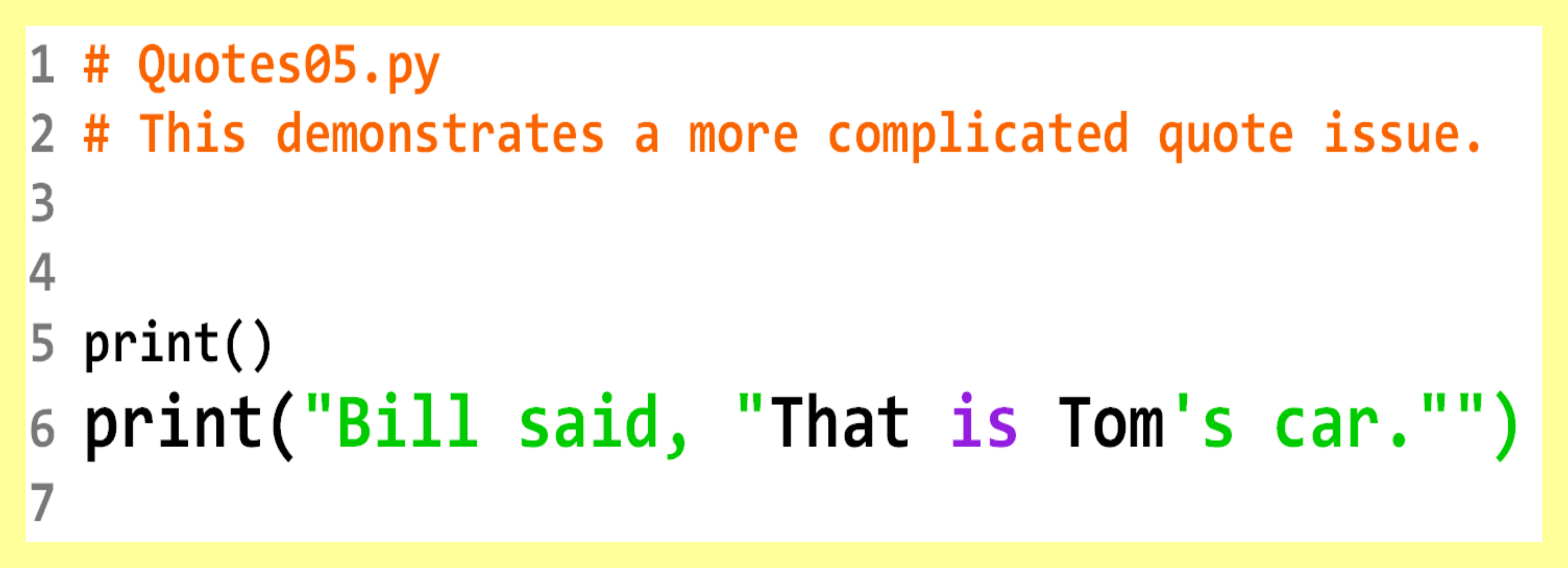
```
 ----jGRASP exec: python Quotes03.py
  File "Quotes03.py", line 8
    print('That is Tom's car.')
                       ^
SyntaxError: invalid syntax

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



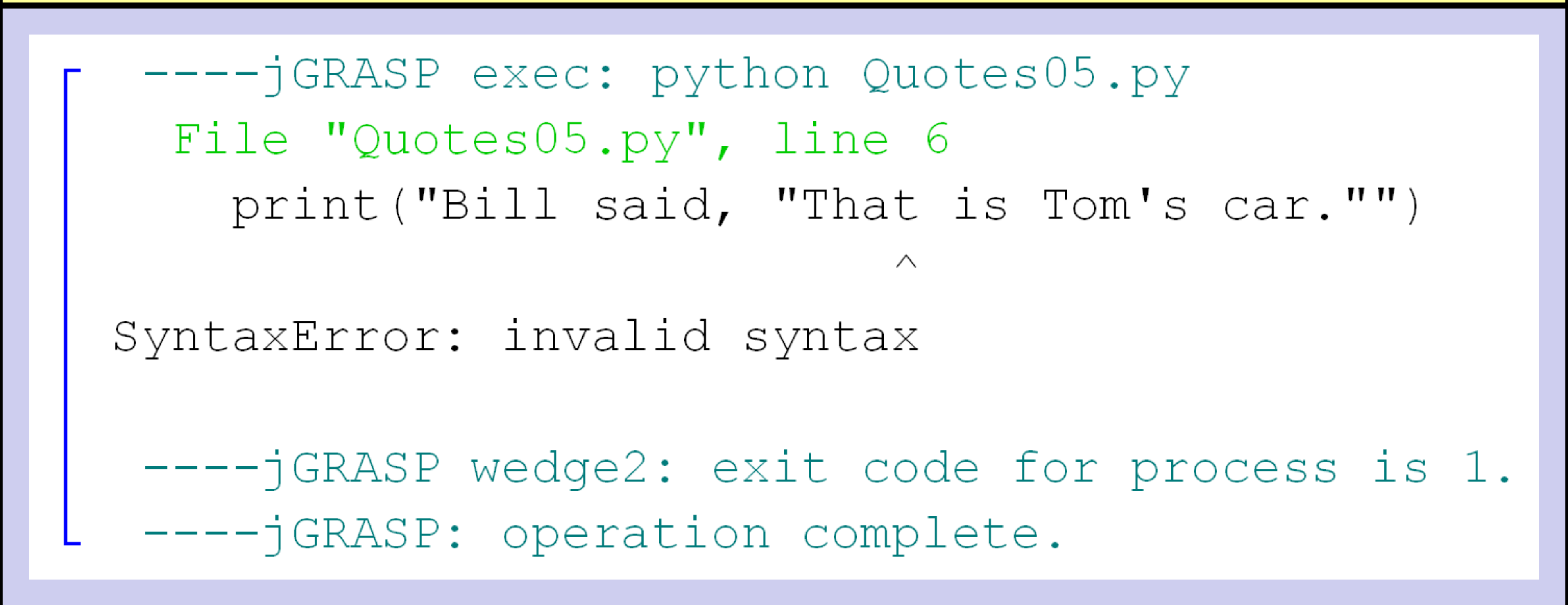
```python
# Quotes04.py
# This program demonstrates a simple fix
# for the previous program by putting the
# text inside double quotes.


print()
print("That is Tom's car.")

```



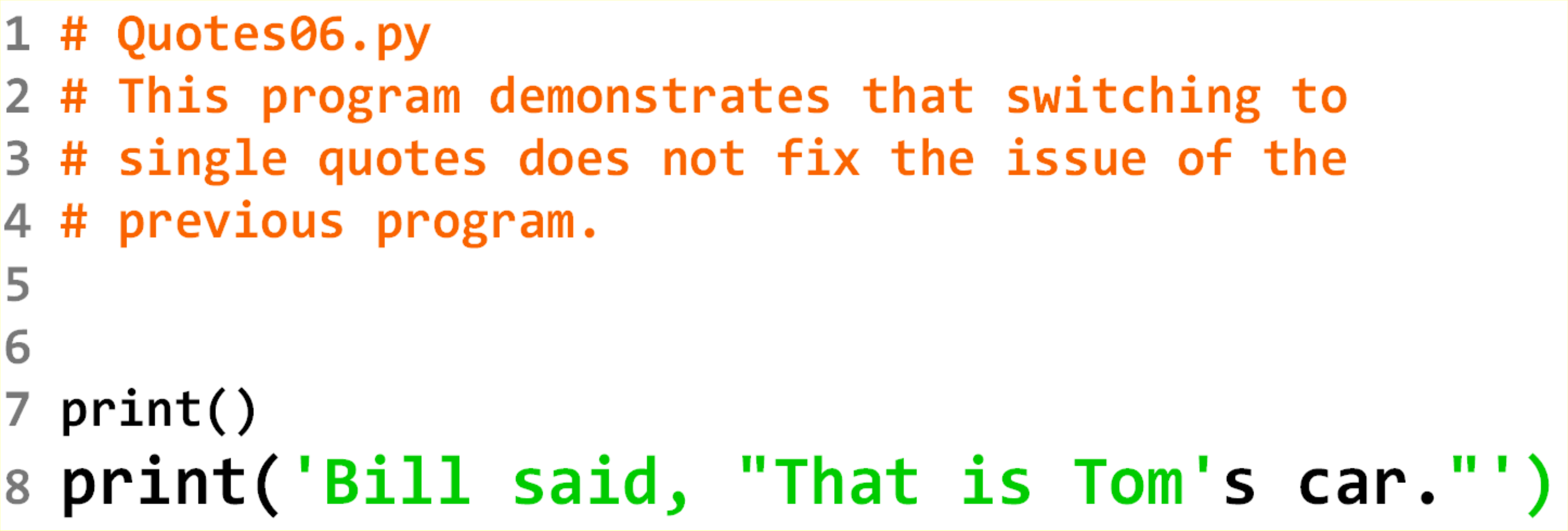
```
 ----jGRASP exec: python Quotes04.py

That is Tom's car.

 ----jGRASP: operation complete.
```



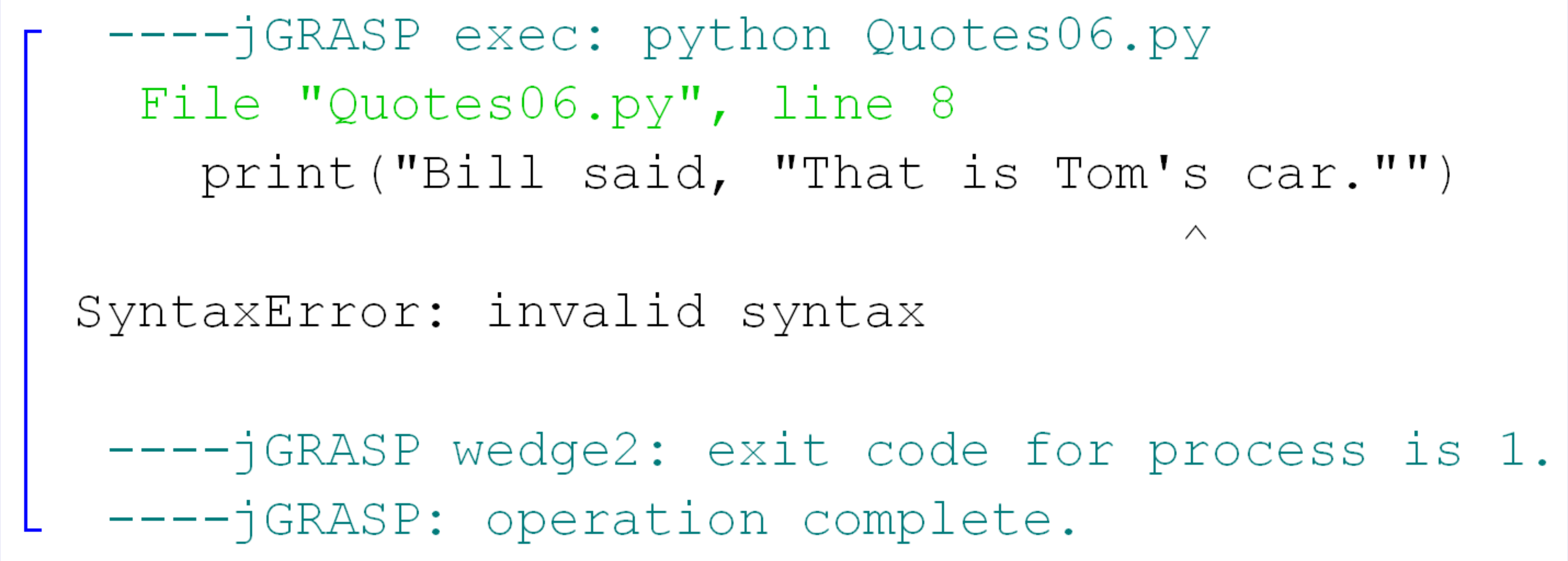
```
1 # Quotes05.py
2 # This demonstrates a more complicated quote issue.
3
4
5 print()
6 print("Bill said, "That is Tom's car."")
7
```



```
 ----jGRASP exec: python Quotes05.py
  File "Quotes05.py", line 6
    print("Bill said, "That is Tom's car."")
                          ^
SyntaxError: invalid syntax

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

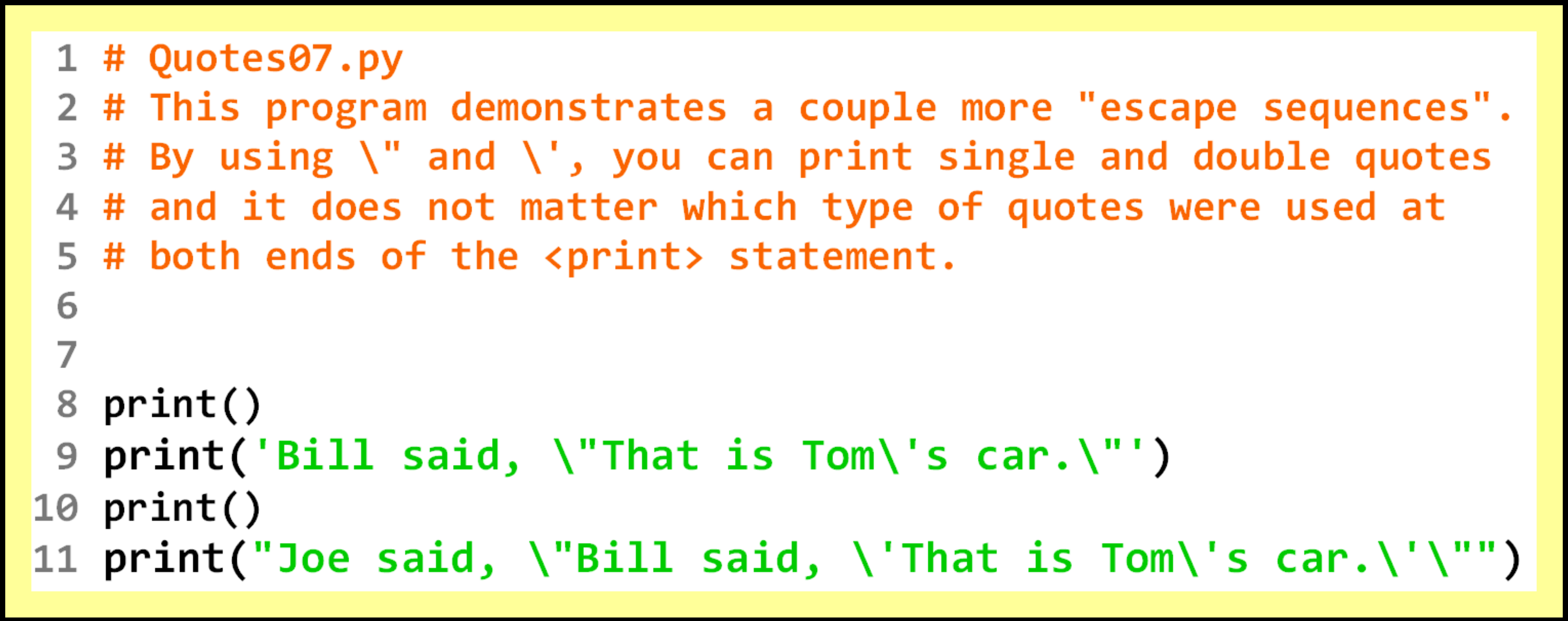


```
1 # Quotes06.py
2 # This program demonstrates that switching to
3 # single quotes does not fix the issue of the
4 # previous program.
5
6
7 print()
8 print('Bill said, "That is Tom's car."')
```

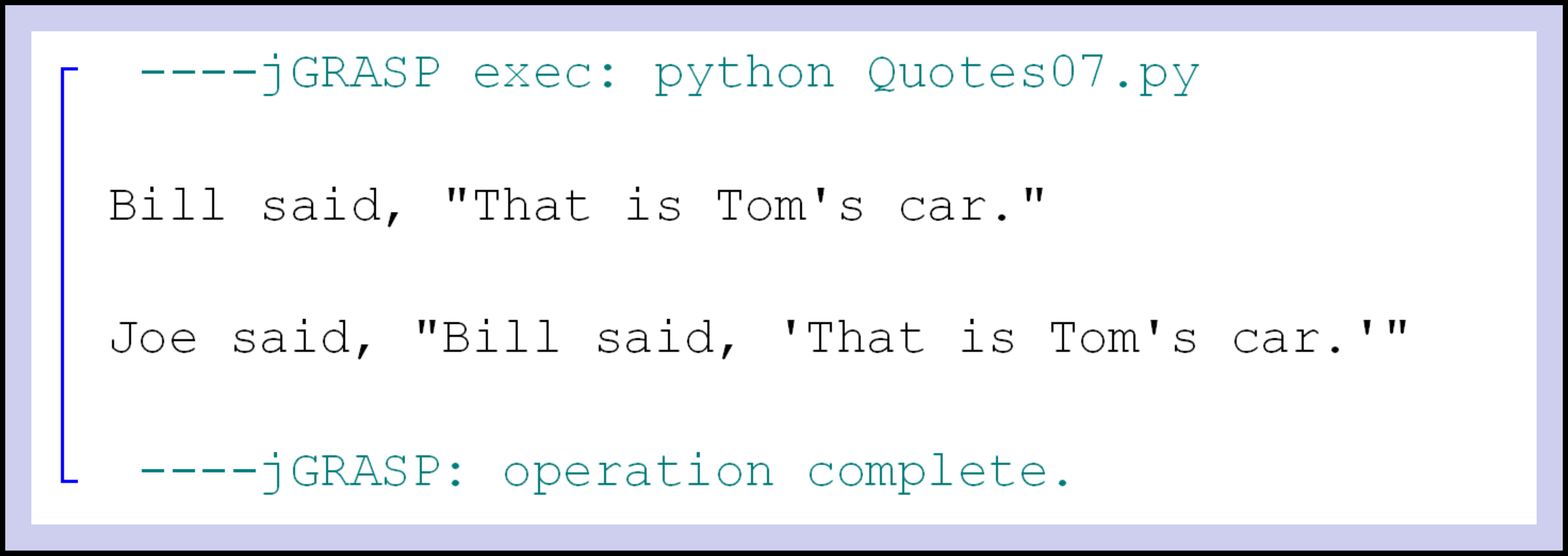


```
 ----jGRASP exec: python Quotes06.py
  File "Quotes06.py", line 8
    print("Bill said, "That is Tom's car."")
                                  ^

SyntaxError: invalid syntax


 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

More Escape Sequences

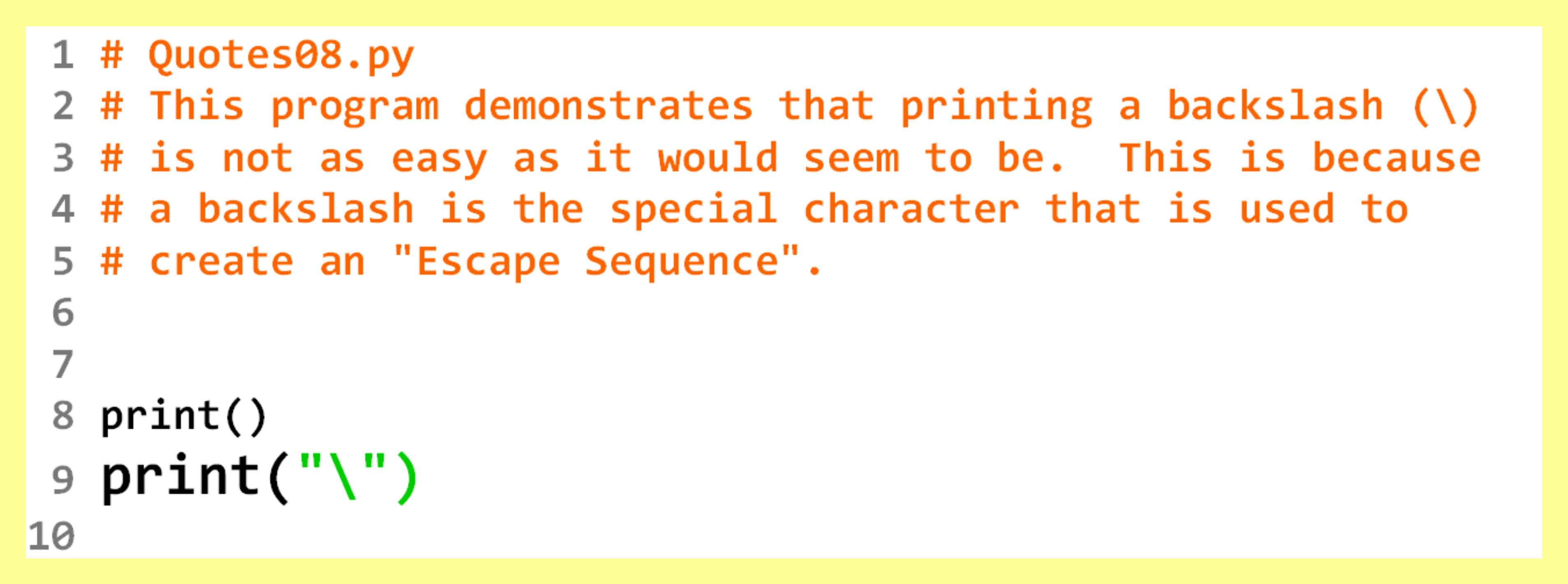


```python
# Quotes07.py
# This program demonstrates a couple more "escape sequences".
# By using \" and \', you can print single and double quotes
# and it does not matter which type of quotes were used at
# both ends of the <print> statement.


print()
print('Bill said, \"That is Tom\'s car.\"')
print()
print("Joe said, \"Bill said, \'That is Tom\'s car.\'\"")
```

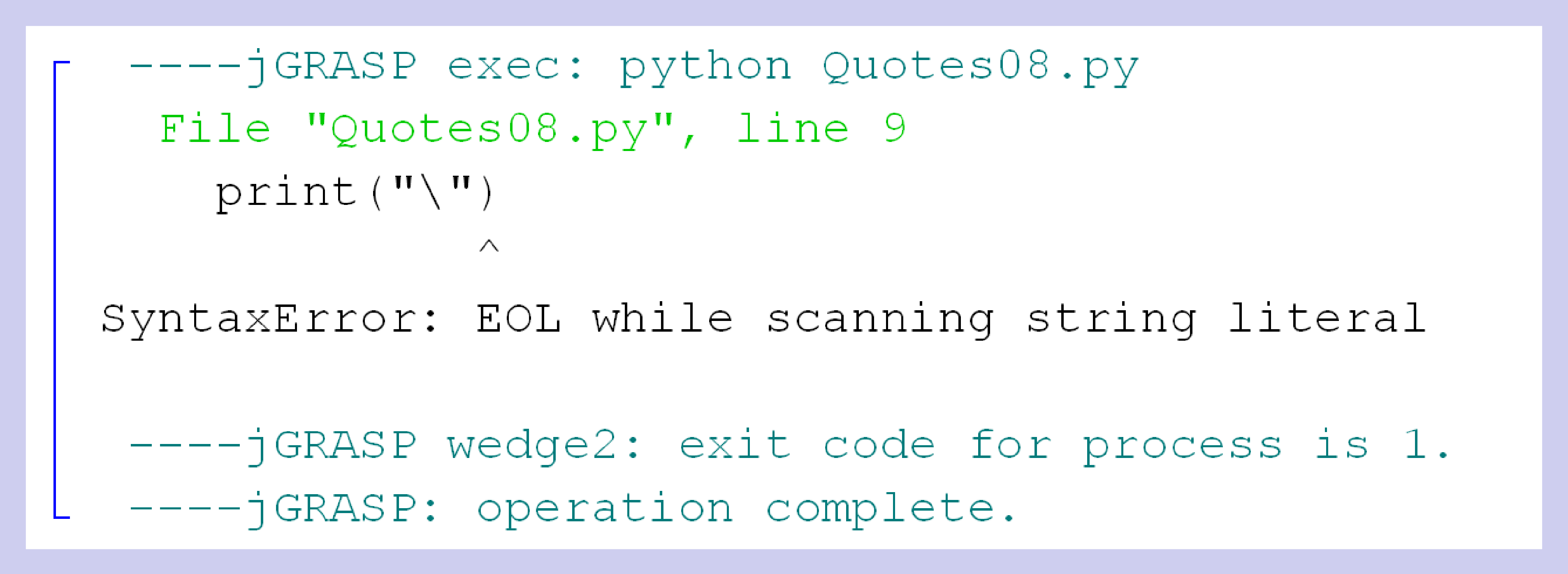


```
 ----jGRASP exec: python Quotes07.py

Bill said, "That is Tom's car."

Joe said, "Bill said, 'That is Tom's car.'"

 ----jGRASP: operation complete.
```

```python
# Quotes08.py
# This program demonstrates that printing a backslash (\)
# is not as easy as it would seem to be.  This is because
# a backslash is the special character that is used to
# create an "Escape Sequence".


print()
print("\")

```

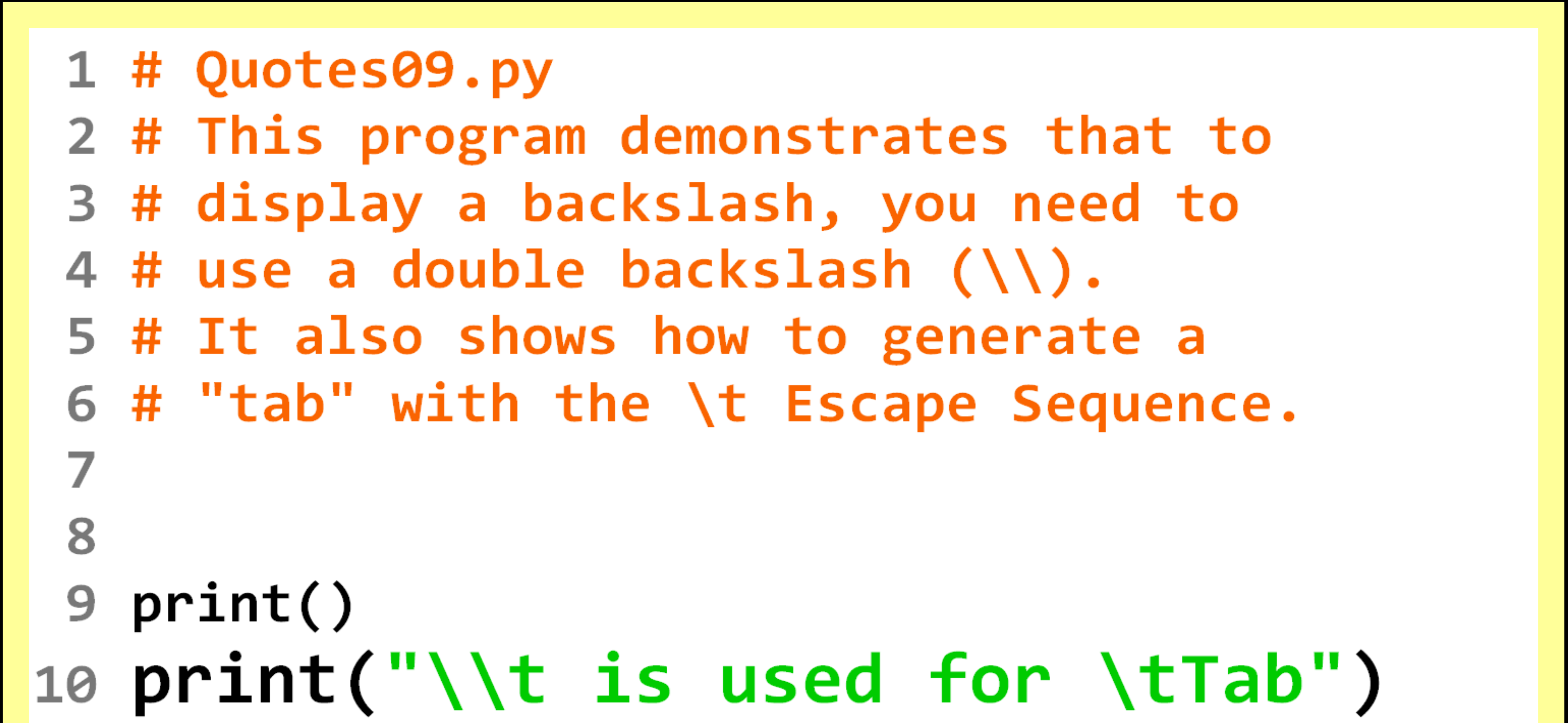


```
1 # Quotes08.py
2 # This program demonstrates that printing a backslash (\)
3 # is not as easy as it would seem to be.  This is because
4 # a backslash is the special character that is used to
5 # create an "Escape Sequence".
6
7
8 print()
9 print("\")
10
```

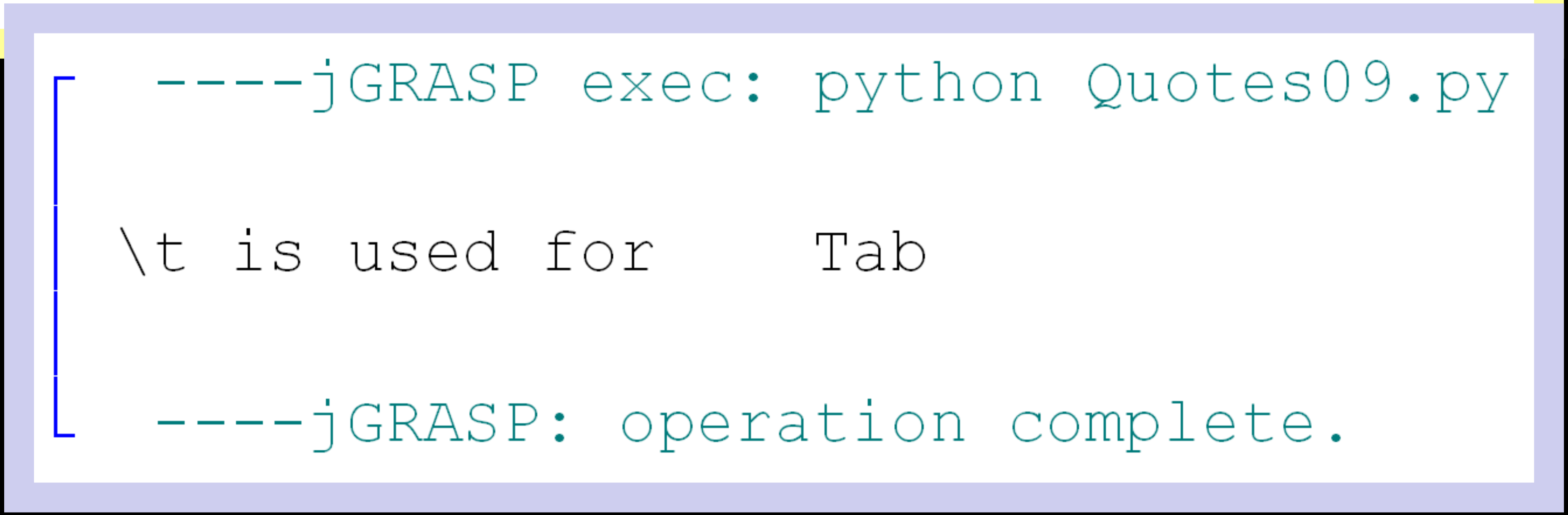


```
 ----jGRASP exec: python Quotes08.py
  File "Quotes08.py", line 9
    print("\")
             ^
SyntaxError: EOL while scanning string literal


 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```

```python
# Quotes09.py
# This program demonstrates that to
# display a backslash, you need to
# use a double backslash (\\).
# It also shows how to generate a
# "tab" with the \t Escape Sequence.


print()
print("\\t is used for \tTab")
```

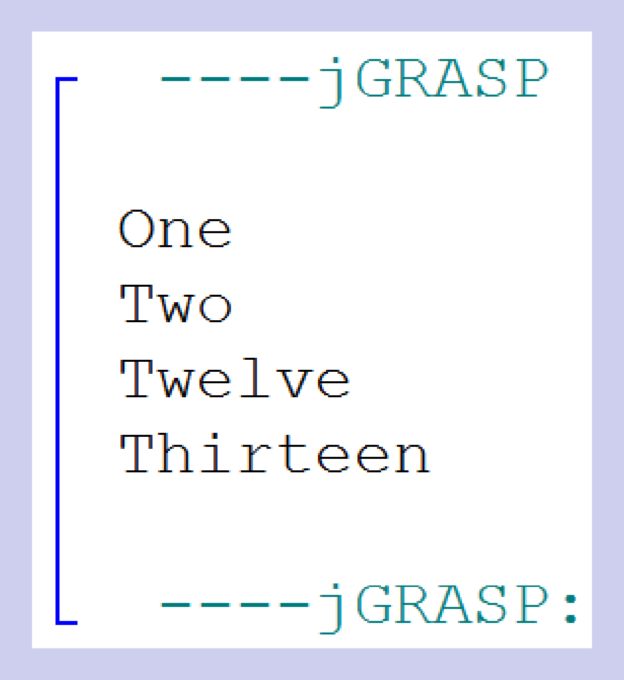


```python
# Quotes09.py
# This program demonstrates that to
# display a backslash, you need to
# use a double backslash (\\).
# It also shows how to generate a
# "tab" with the \t Escape Sequence.


print()
print("\\t is used for \tTab")
```



```
 ----jGRASP exec: python Quotes09.py

\t is used for     Tab

 ----jGRASP: operation complete.
```

```python
# Quotes10.py
# This program is almost identical to Comments02.py
# and demonstrates that a multi-line comment can
# also be created with triple-single-quotes.


print()
print("One")
print("Two")
'''
print("Three")
print("Four")
print("Five")
print("Six")         # half a dozen
print("Seven")
print("Eight")
print("Nine")
print("Ten")
print("Eleven")
'''
print("Twelve")      # one dozen
print("Thirteen")    # one baker's dozen
```



```
 ----jGRASP

One
Two
Twelve
Thirteen

 ----jGRASP:
```

Section 3.7

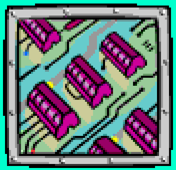
The Responsible Use of Computers

Protect Your Computer From the Environment

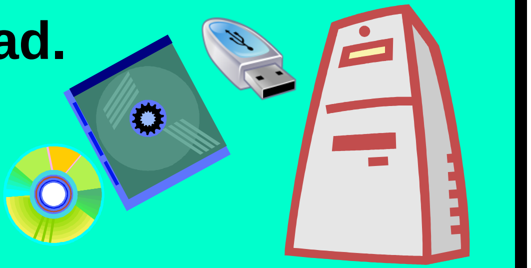
Computers and computer information are vulnerable.

Computers can be <u>physically</u> damaged.

RAM is temporary, save often!

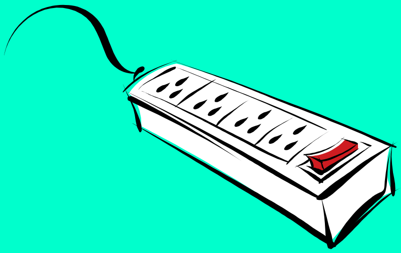


CDs, USB Drives and even hard drives can go bad.



<u>Back up your stuff!</u>

Blackouts and Power Surges
are major problems.
Backup Batteries and
Surge Protectors are useful.



Protect Your Computer From Viruses

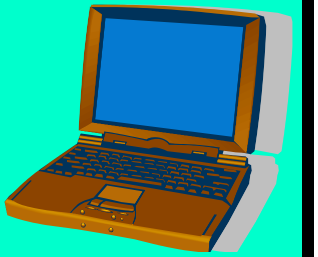
A *virus* is a special program that has these two qualities:

- **The ability to duplicate itself to spread to other systems.**

- **The *payload* it carries, which is a program that will do some type harm to the computer.**

Protect Your Computer From Improper Access

- **Don't leave your computer unattended and logged in.**
- **Information can easily be copied or erased from an unattended computer.**
- **Unattended laptops are easily stolen.**
- **Label and guard your stuff!**



The Ethical Use of Computer Software

Copying copyrighted software is illegal.

Companies have been sued and people have been arrested for not taking this seriously.

Hacking

Some confused people think that if you hack into a network, it is fine as long as you don't steal any money or damage any information.

WRONG!

Just attempting to hack into a computer or network is a misdemeanor.

If you actually succeed at getting in, you have already committed a felony.

Yes, high school students have been prosecuted for this!

Vandalism

Physical computer vandalism is bad, but typically not a problem in high school.

You also need to make sure you do not alter ANY settings on the computer or install ANY software unless directed to do so by your teacher.

Altering settings can prevent the computer from working properly.

Downloading software has legal issues and can also cause the spread of viruses.