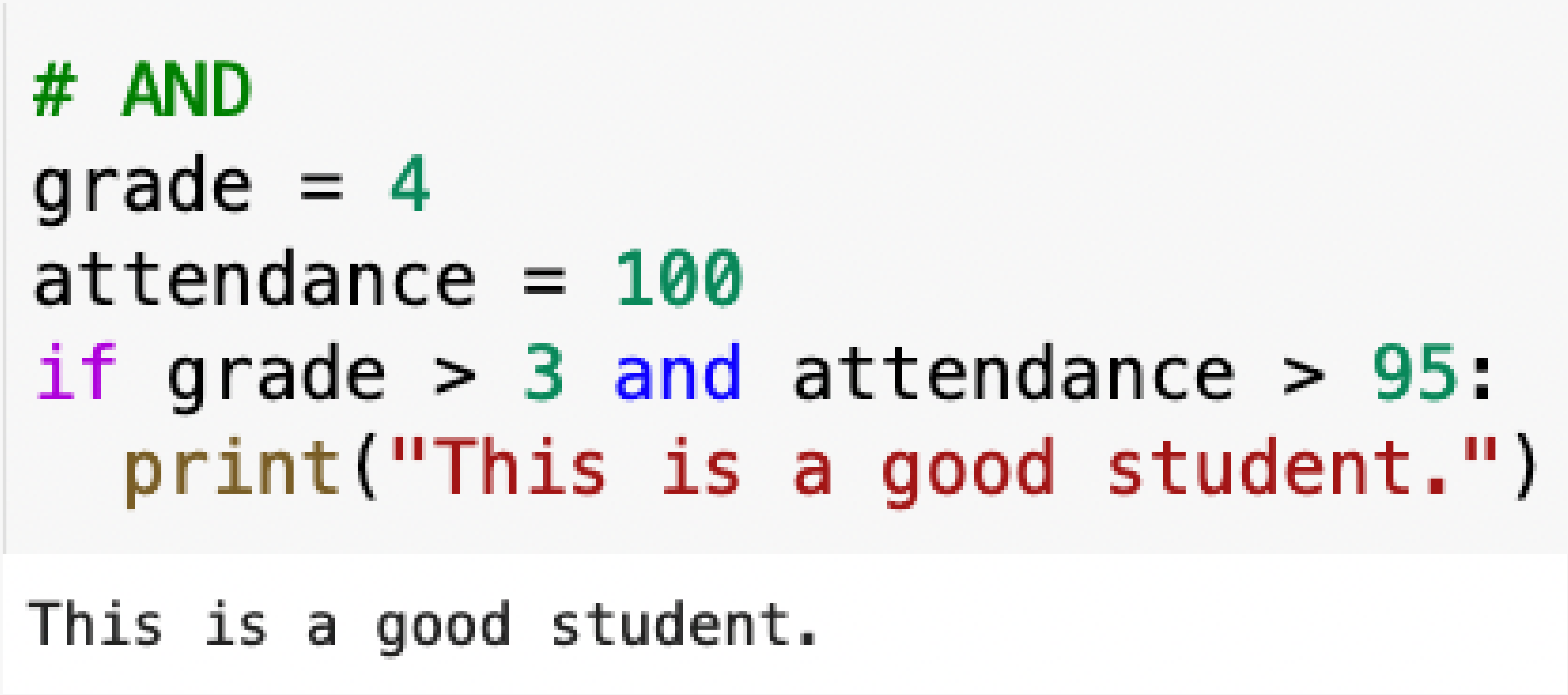
Python 3 reserved words

'False', 'None', 'True', 'and', 'as', 'assert', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
'finally','for','from', 'global', 'if', 'import', 'in', 'is',
'lambda','nonlocal', 'not', 'or', 'pass', 'raise', 'return',
'try','while', 'with', 'yield'

So many, no worries
in next 100 seconds,
you will know them all

Python 3.x reserved words

and

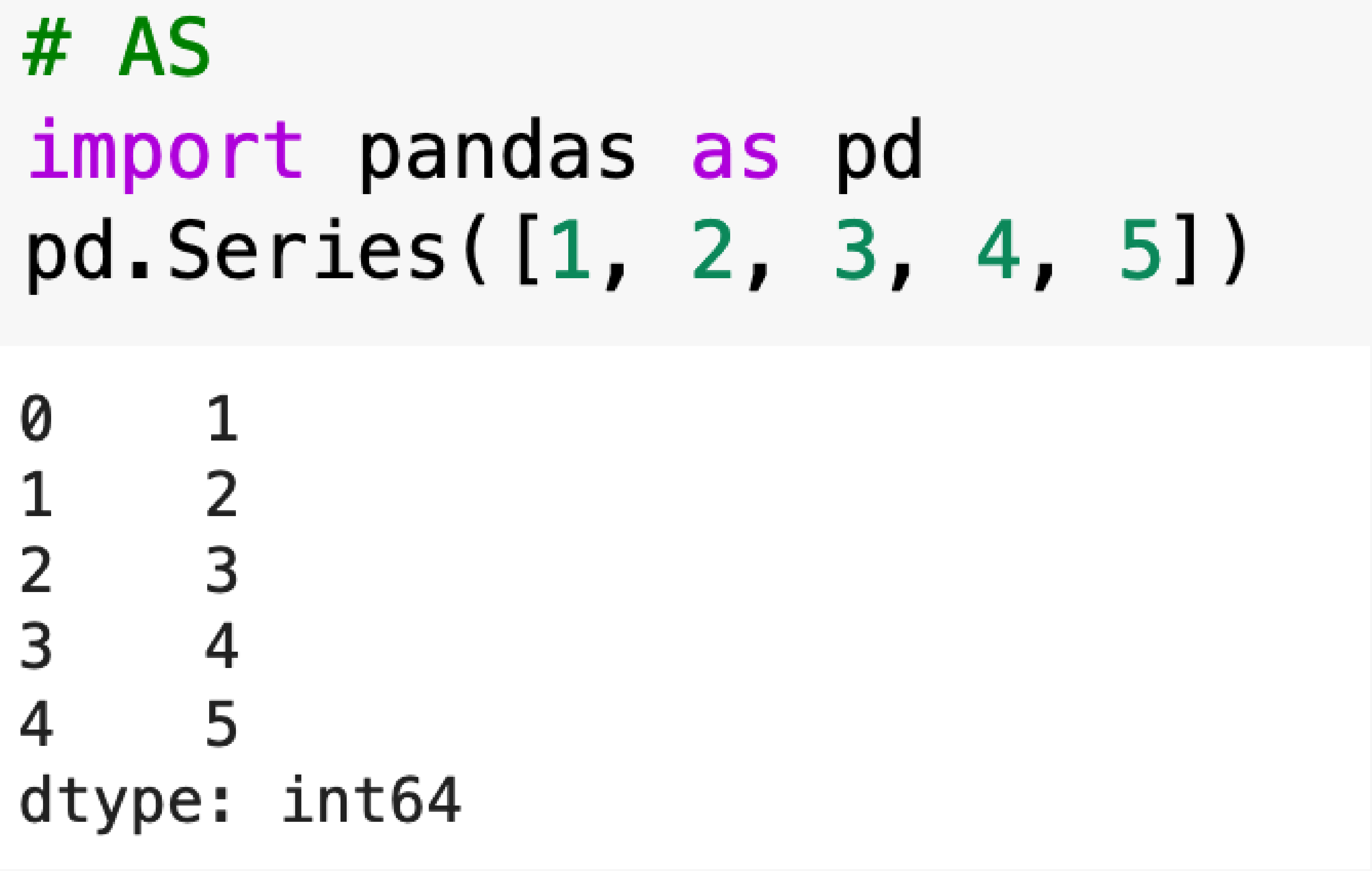


```python
# AND
grade = 4
attendance = 100
if grade > 3 and attendance > 95:
  print("This is a good student.")
```

```
This is a good student.
```

Python 3.x reserved words

as

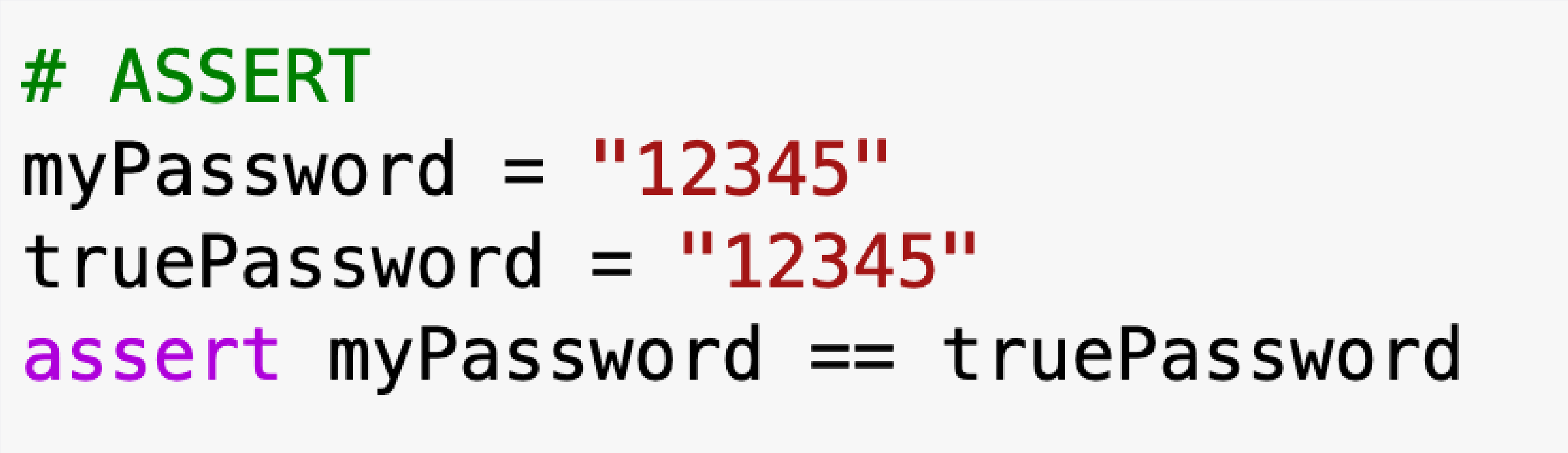


```
# AS
import pandas as pd
pd.Series([1, 2, 3, 4, 5])
```

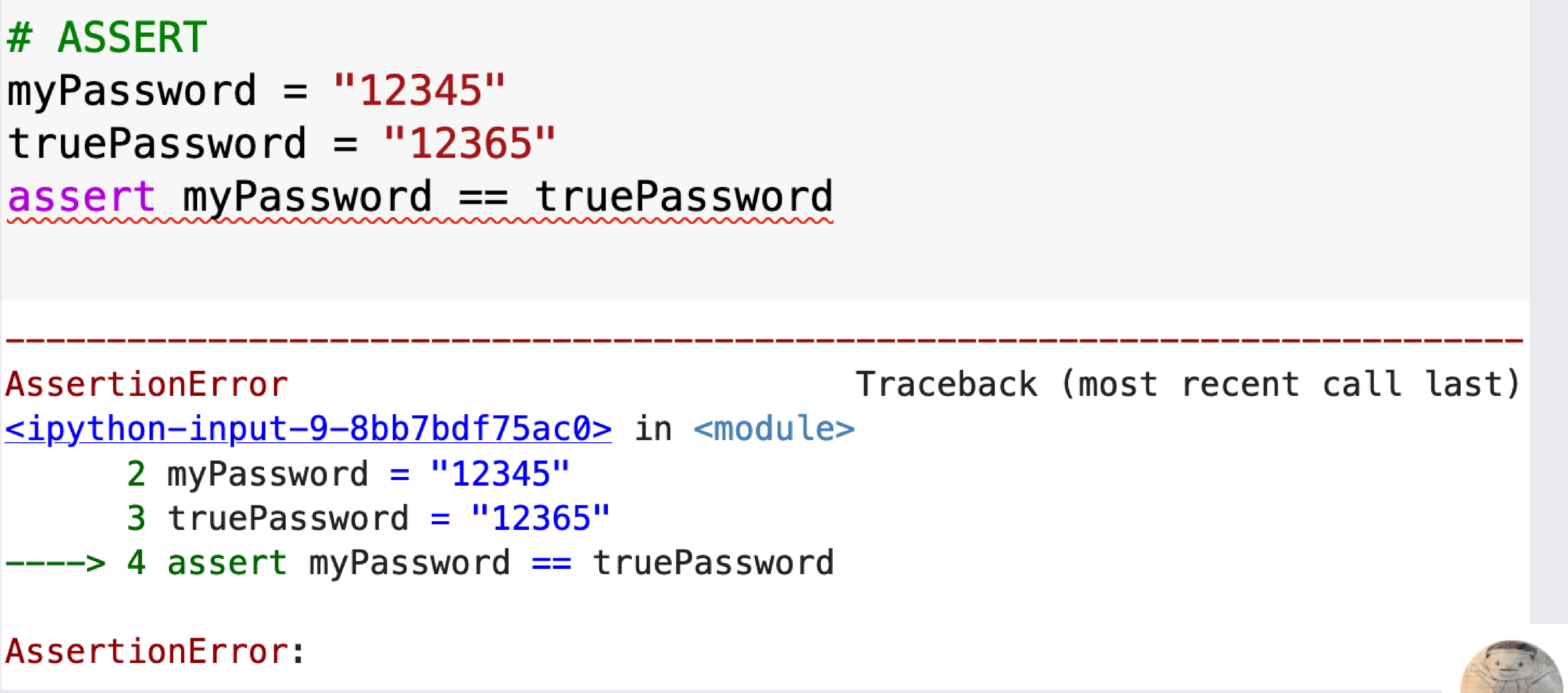
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

Python 3.x reserved words

assert



```python
# ASSERT
myPassword = "12345"
truePassword = "12345"
assert myPassword == truePassword
```

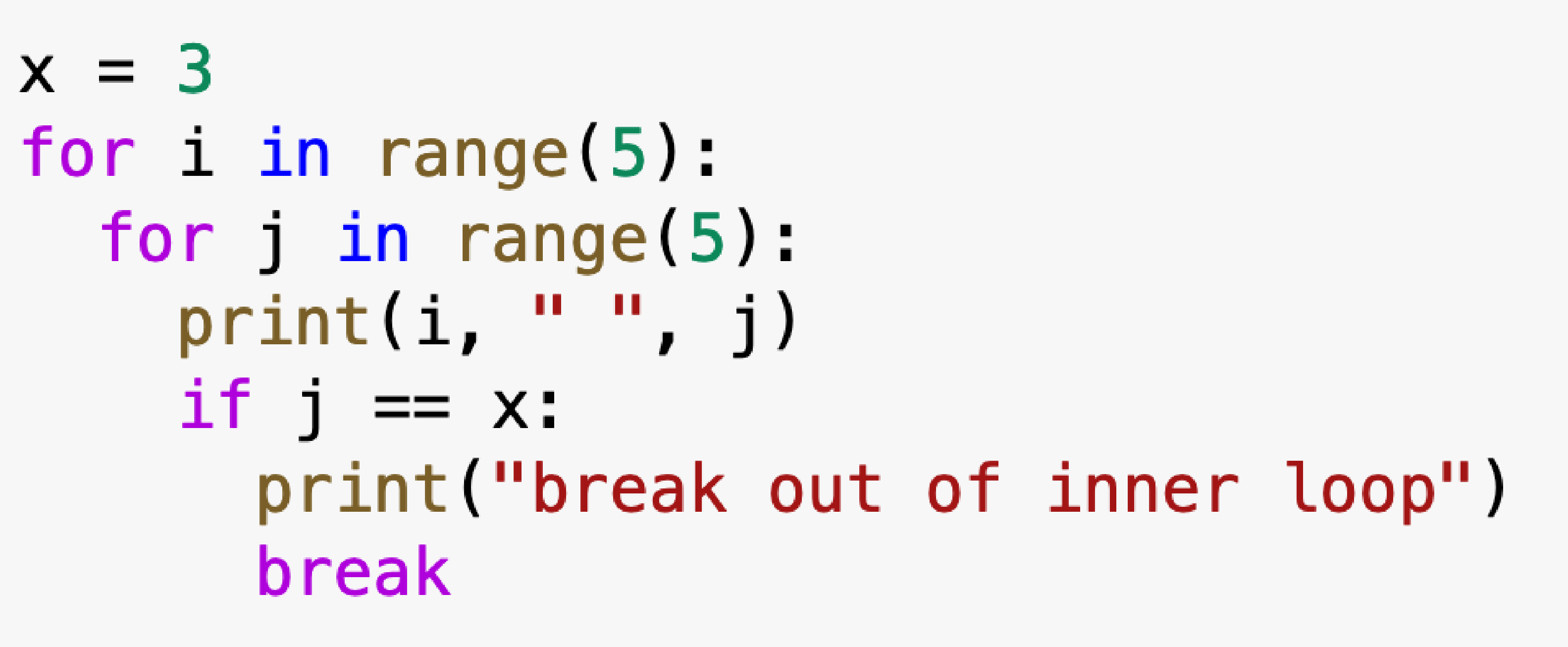


```python
# ASSERT
myPassword = "12345"
truePassword = "12365"
assert myPassword == truePassword
```

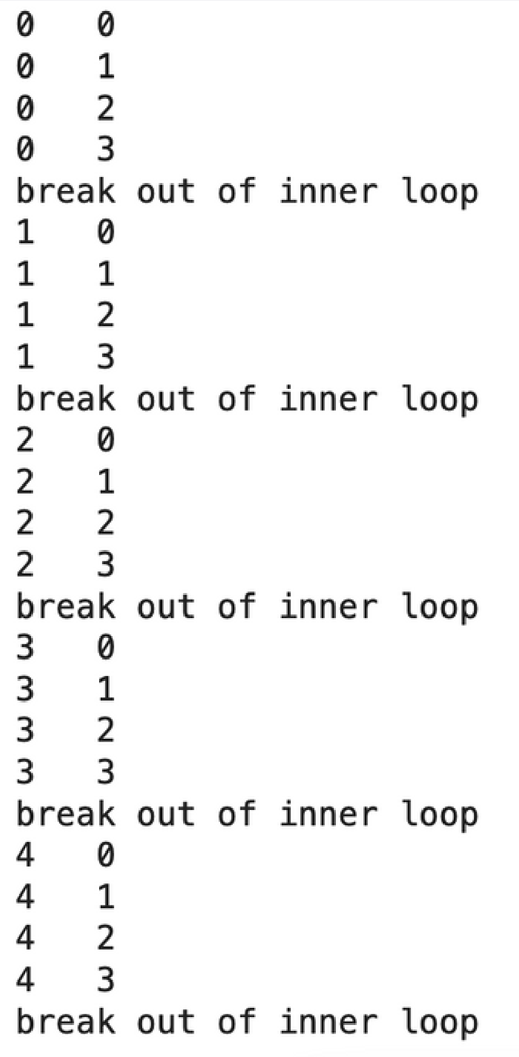
```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-9-8bb7bdf75ac0> in <module>
      2 myPassword = "12345"
      3 truePassword = "12365"
----> 4 assert myPassword == truePassword

AssertionError:
```

Python 3.x reserved words

break



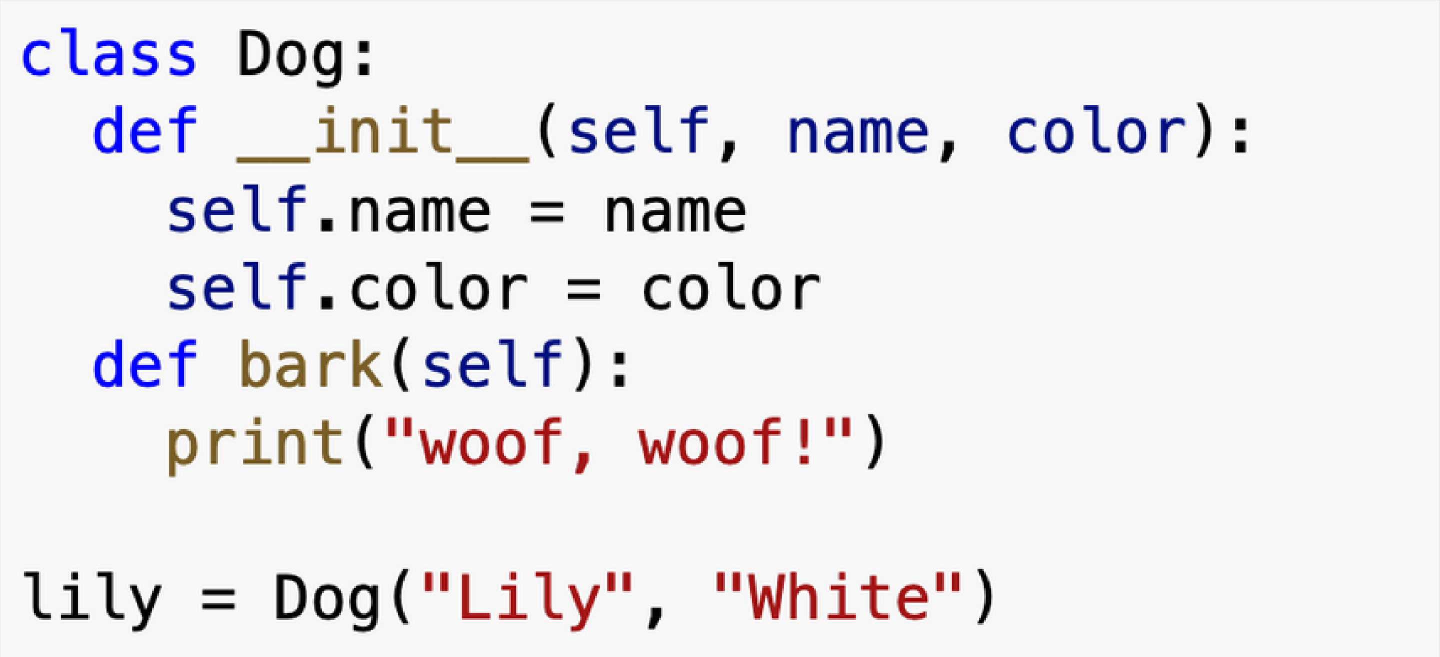
```python
x = 3
for i in range(5):
  for j in range(5):
    print(i, " ", j)
    if j == x:
      print("break out of inner loop")
      break
```



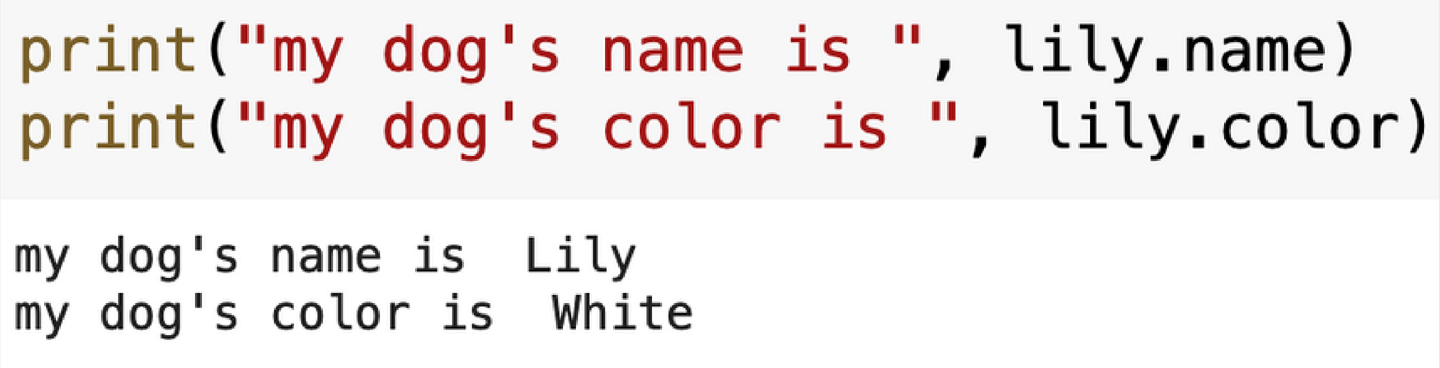
```
0   0
0   1
0   2
0   3
break out of inner loop
1   0
1   1
1   2
1   3
break out of inner loop
2   0
2   1
2   2
2   3
break out of inner loop
3   0
3   1
3   2
3   3
break out of inner loop
4   0
4   1
4   2
4   3
break out of inner loop
```

Python 3.x reserved words

class

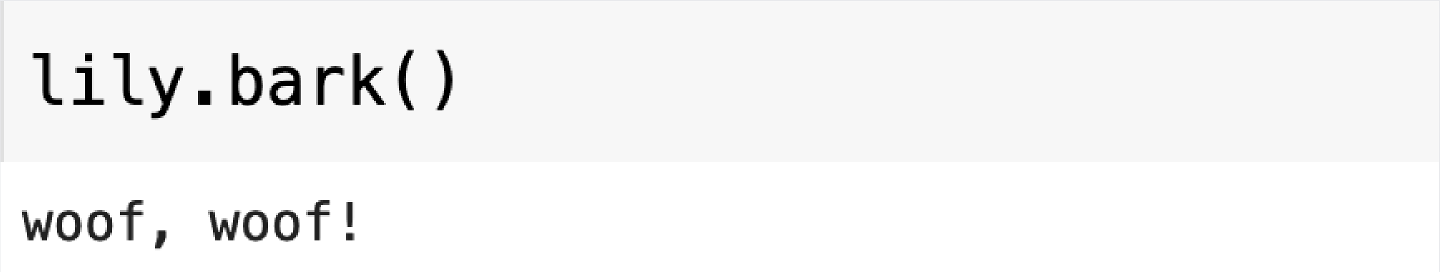


```
class Dog:
  def __init__(self, name, color):
    self.name = name
    self.color = color
  def bark(self):
    print("woof, woof!")

lily = Dog("Lily", "White")
```



```
print("my dog's name is ", lily.name)
print("my dog's color is ", lily.color)
```

```
my dog's name is  Lily
my dog's color is  White
```

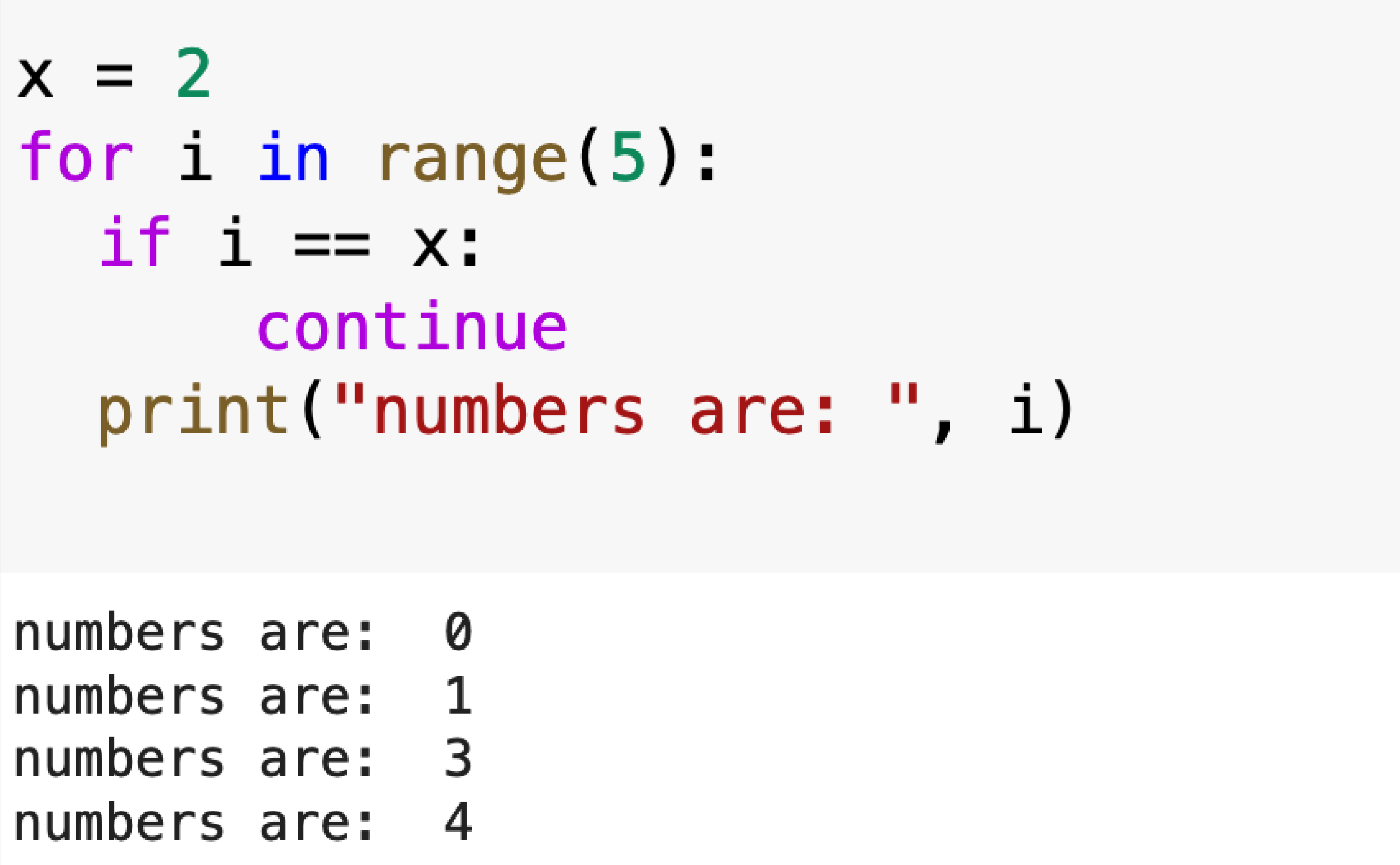


```
lily.bark()
```

```
woof, woof!
```

Python 3.x reserved words

continue

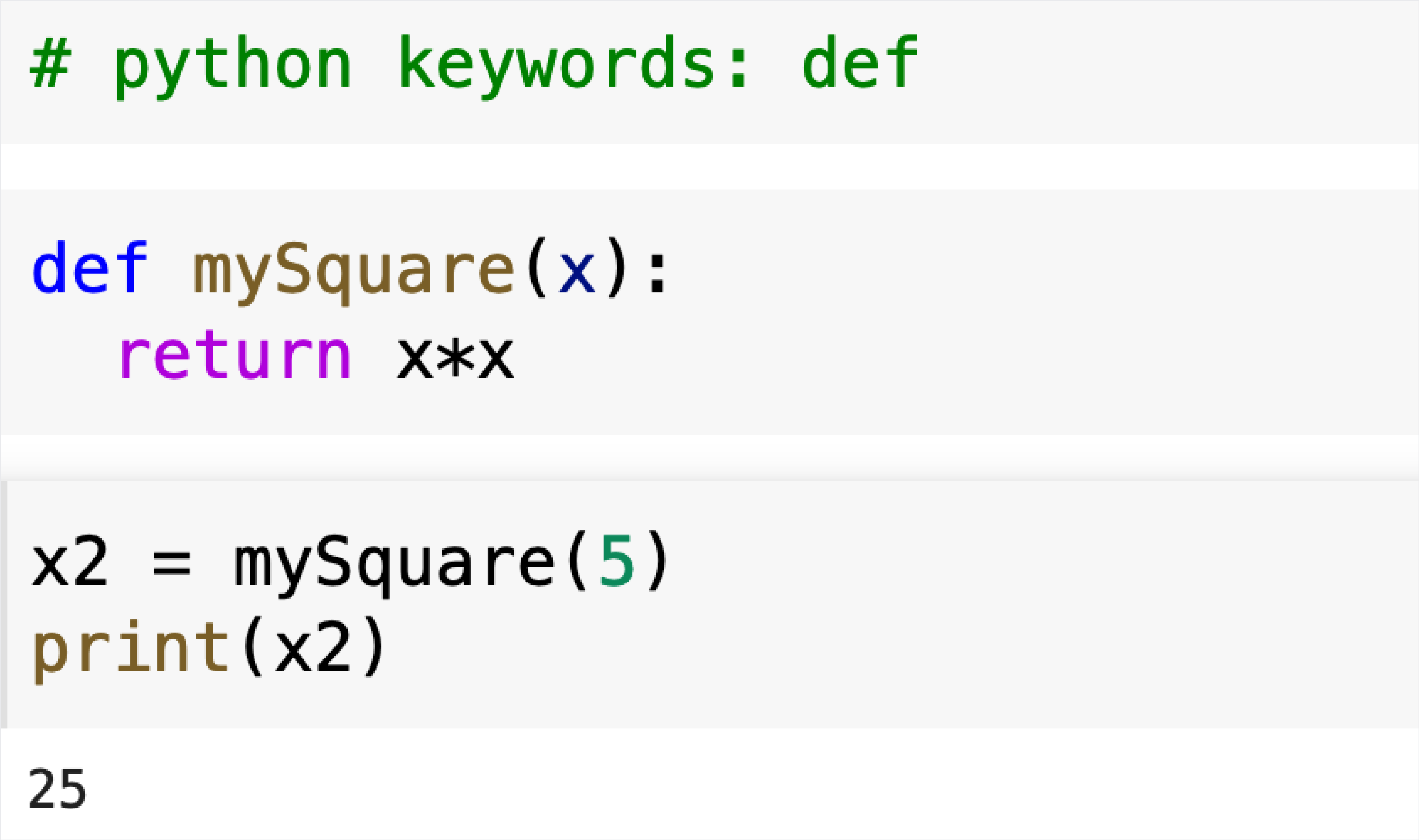


```python
x = 2
for i in range(5):
  if i == x:
      continue
  print("numbers are: ", i)
```

```
numbers are:  0
numbers are:  1
numbers are:  3
numbers are:  4
```

Python 3.x reserved words

def



```
# python keywords: def
```

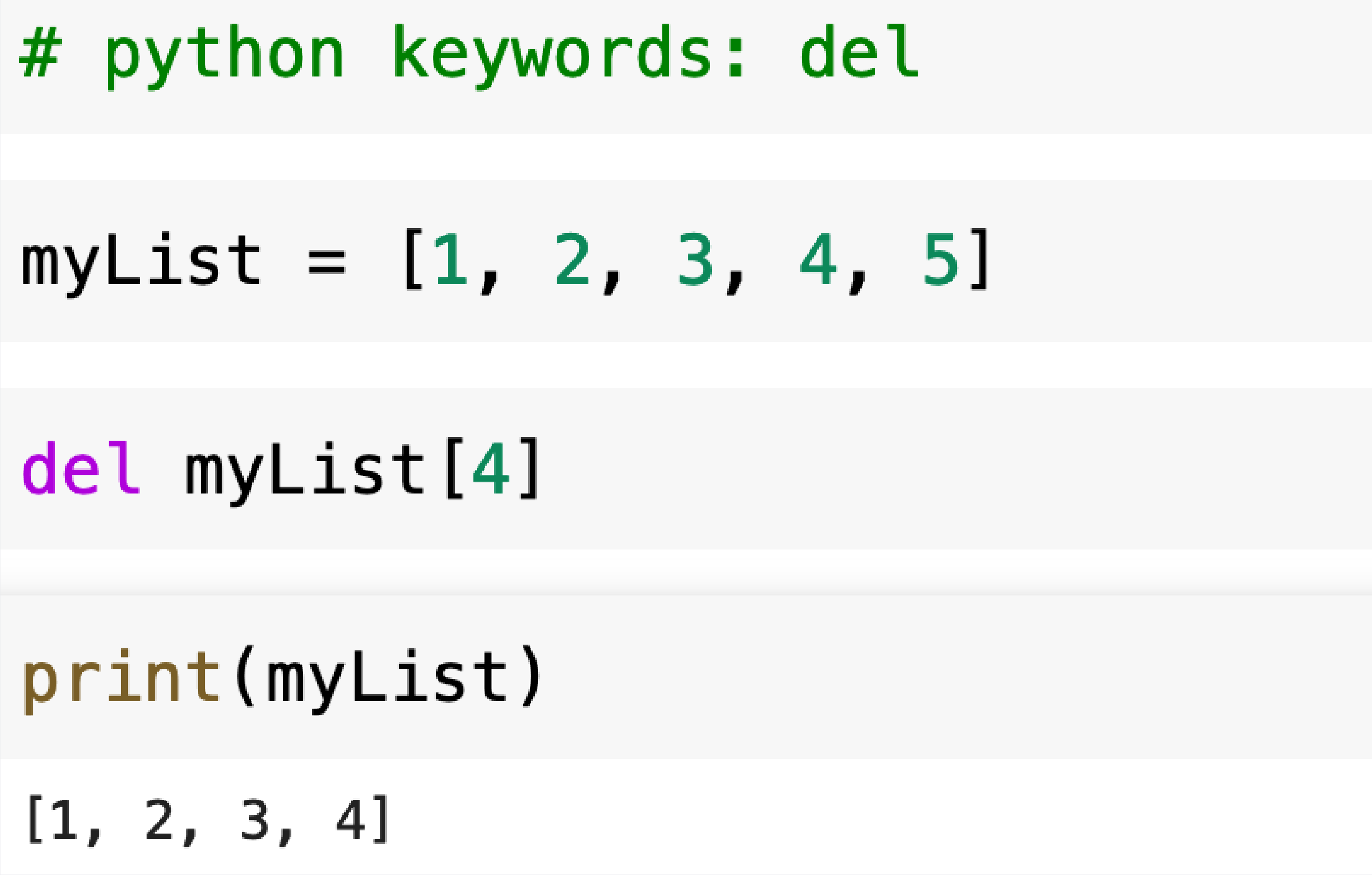
```
def mySquare(x):
  return x*x
```

```
x2 = mySquare(5)
print(x2)
```

```
25
```

Python 3.x reserved words

del



```
# python keywords: del
```

```
myList = [1, 2, 3, 4, 5]
```

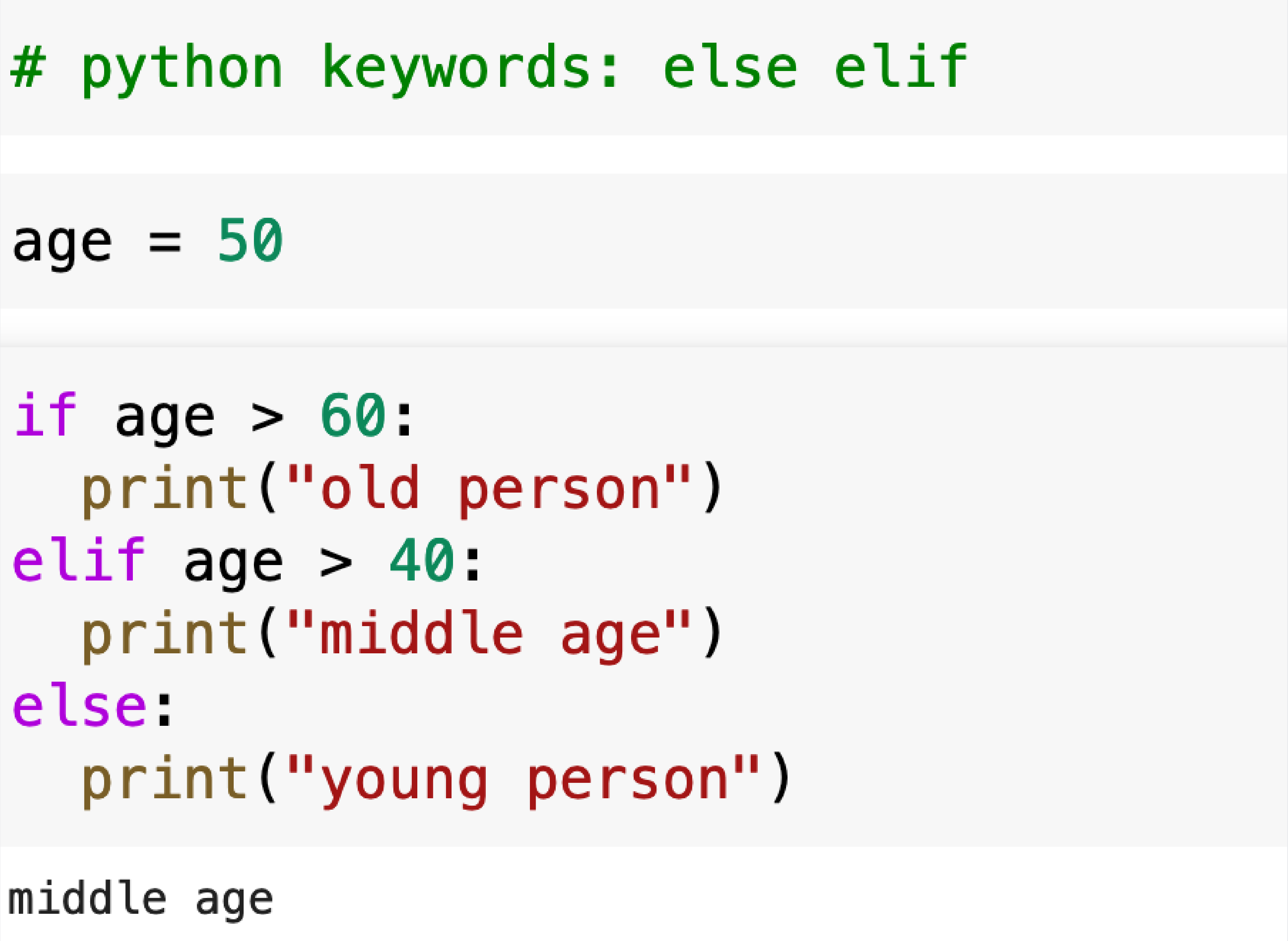
```
del myList[4]
```

```
print(myList)
```

```
[1, 2, 3, 4]
```

Python 3.x reserved words

else, elif



```
# python keywords: else elif
```

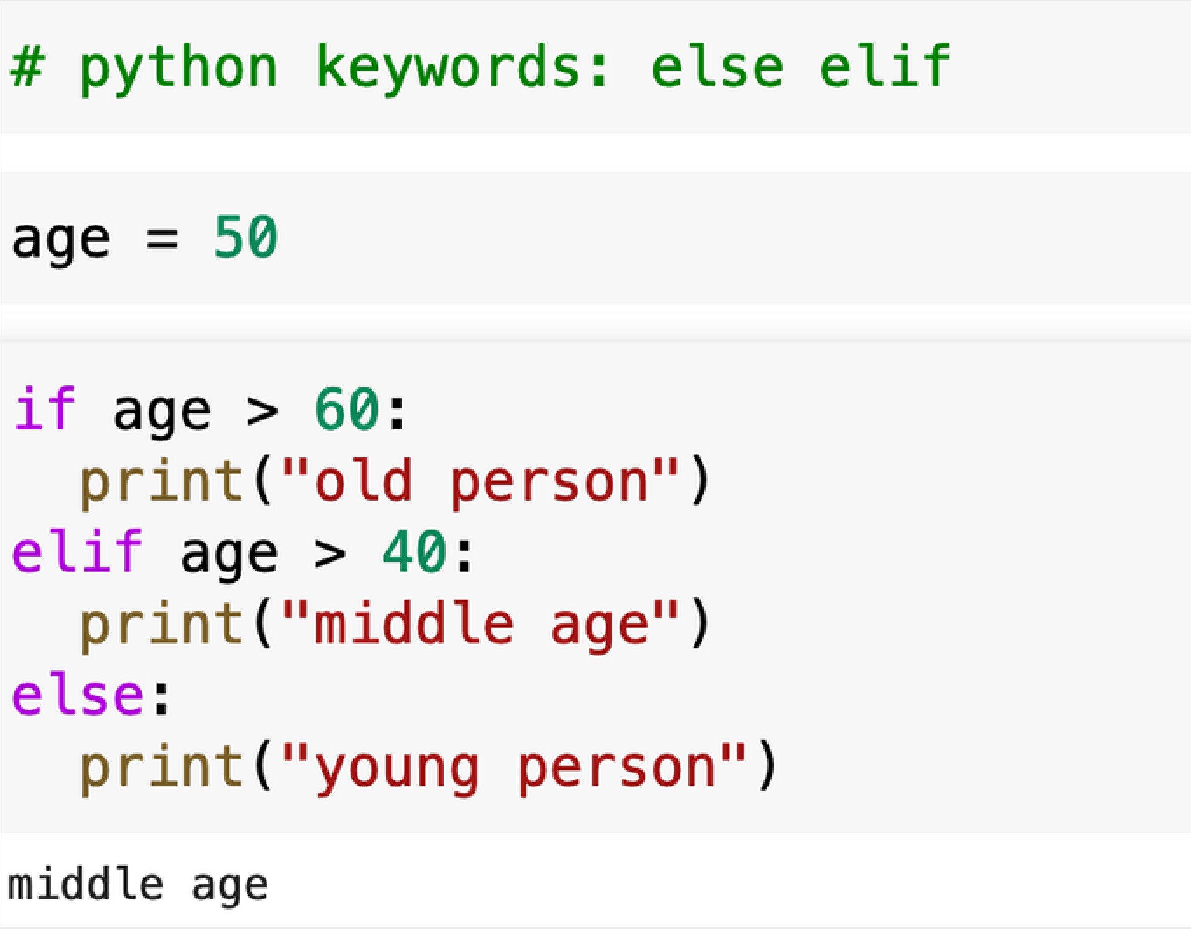
```
age = 50
```

```
if age > 60:
  print("old person")
elif age > 40:
  print("middle age")
else:
  print("young person")
```

```
middle age
```

Python 3.x reserved words

except

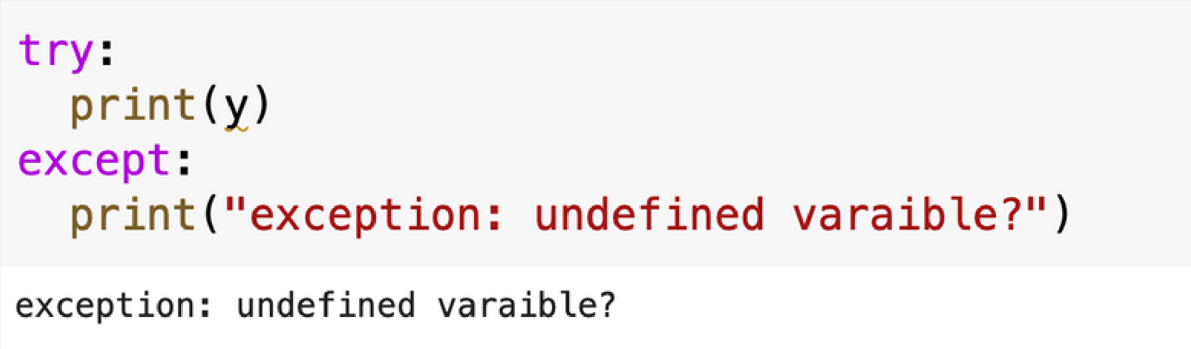


```
# python keywords: else elif
```

```
age = 50
```

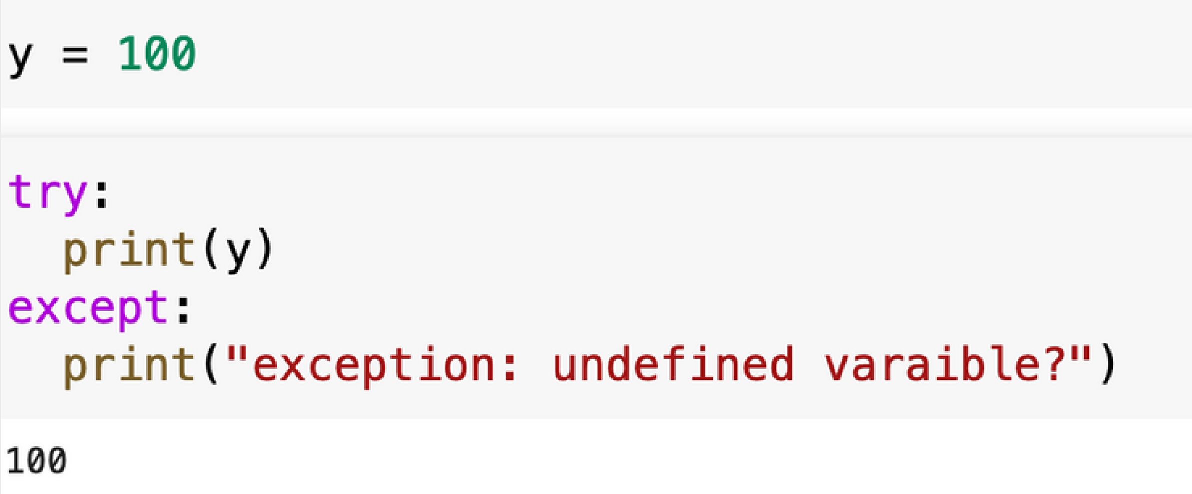
```
if age > 60:
  print("old person")
elif age > 40:
  print("middle age")
else:
  print("young person")
```

```
middle age
```



```
try:
  print(y)
except:
  print("exception: undefined varaible?")
```

```
exception: undefined varaible?
```



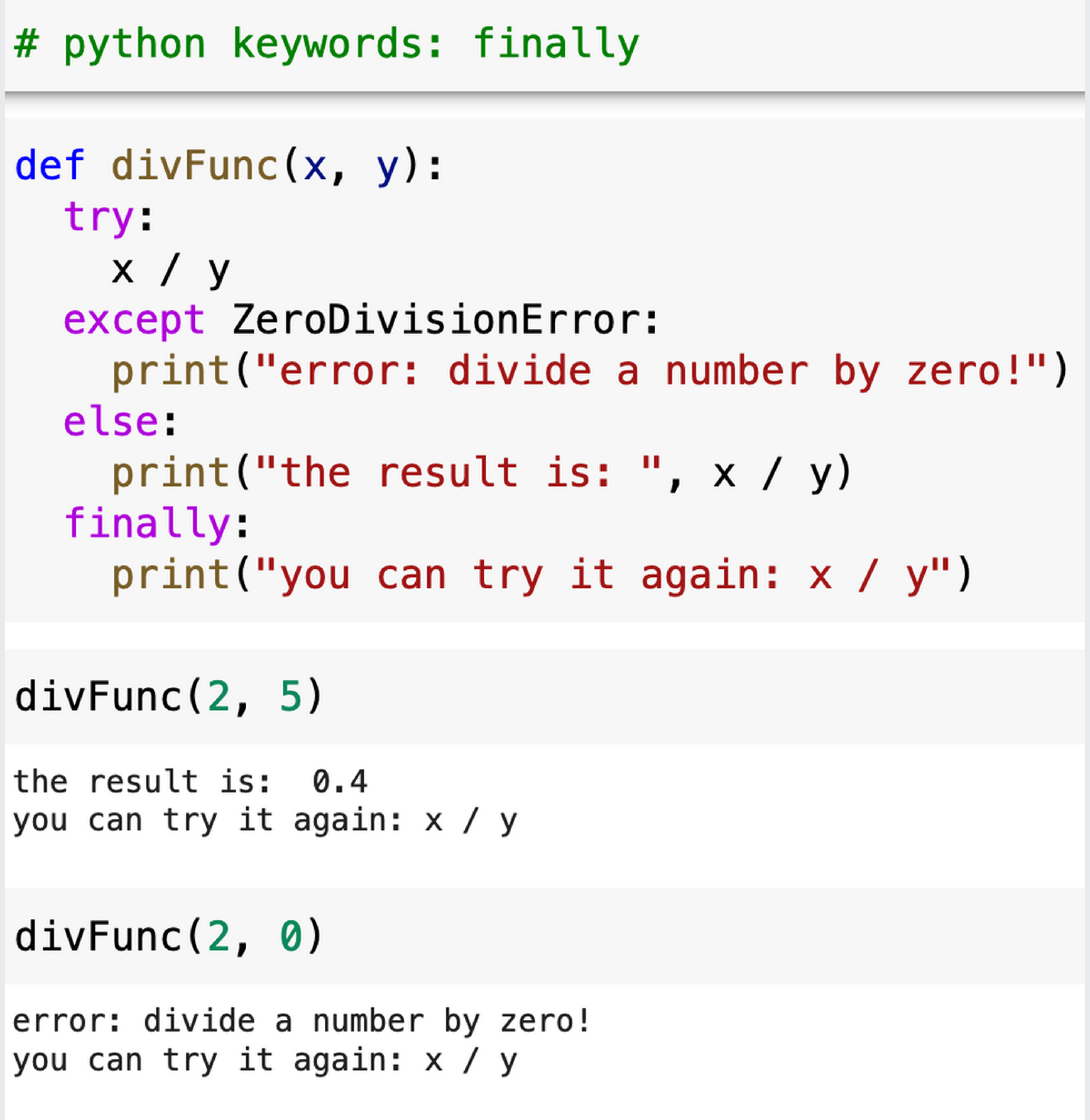
```
y = 100
```

```
try:
  print(y)
except:
  print("exception: undefined varaible?")
```

```
100
```

Python 3.x reserved words

finally



```
# python keywords: finally
```

```
def divFunc(x, y):
  try:
    x / y
  except ZeroDivisionError:
    print("error: divide a number by zero!")
  else:
    print("the result is: ", x / y)
  finally:
    print("you can try it again: x / y")
```

```
divFunc(2, 5)
```

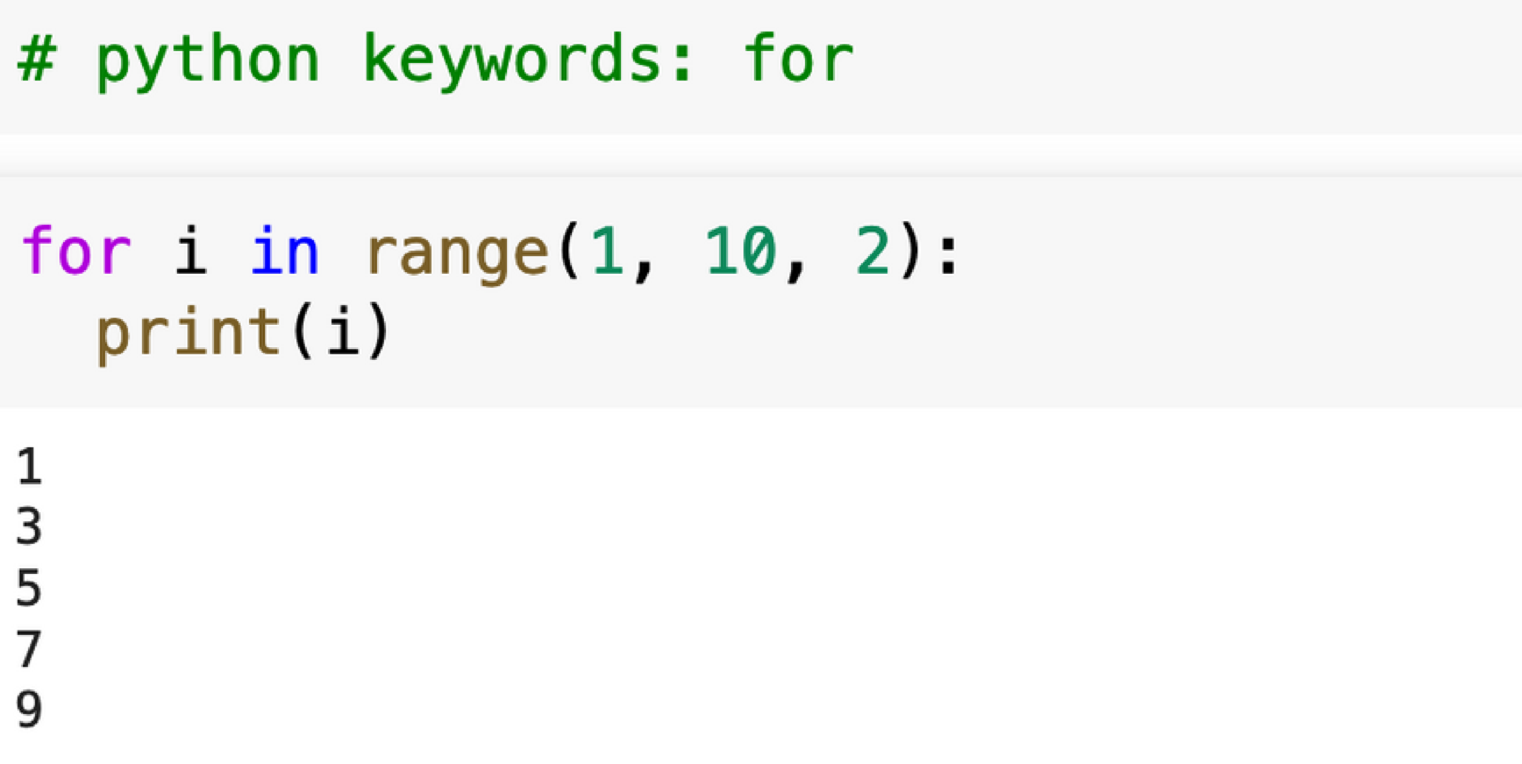
```
the result is:  0.4
you can try it again: x / y
```

```
divFunc(2, 0)
```

```
error: divide a number by zero!
you can try it again: x / y
```

Python 3.x reserved words

for



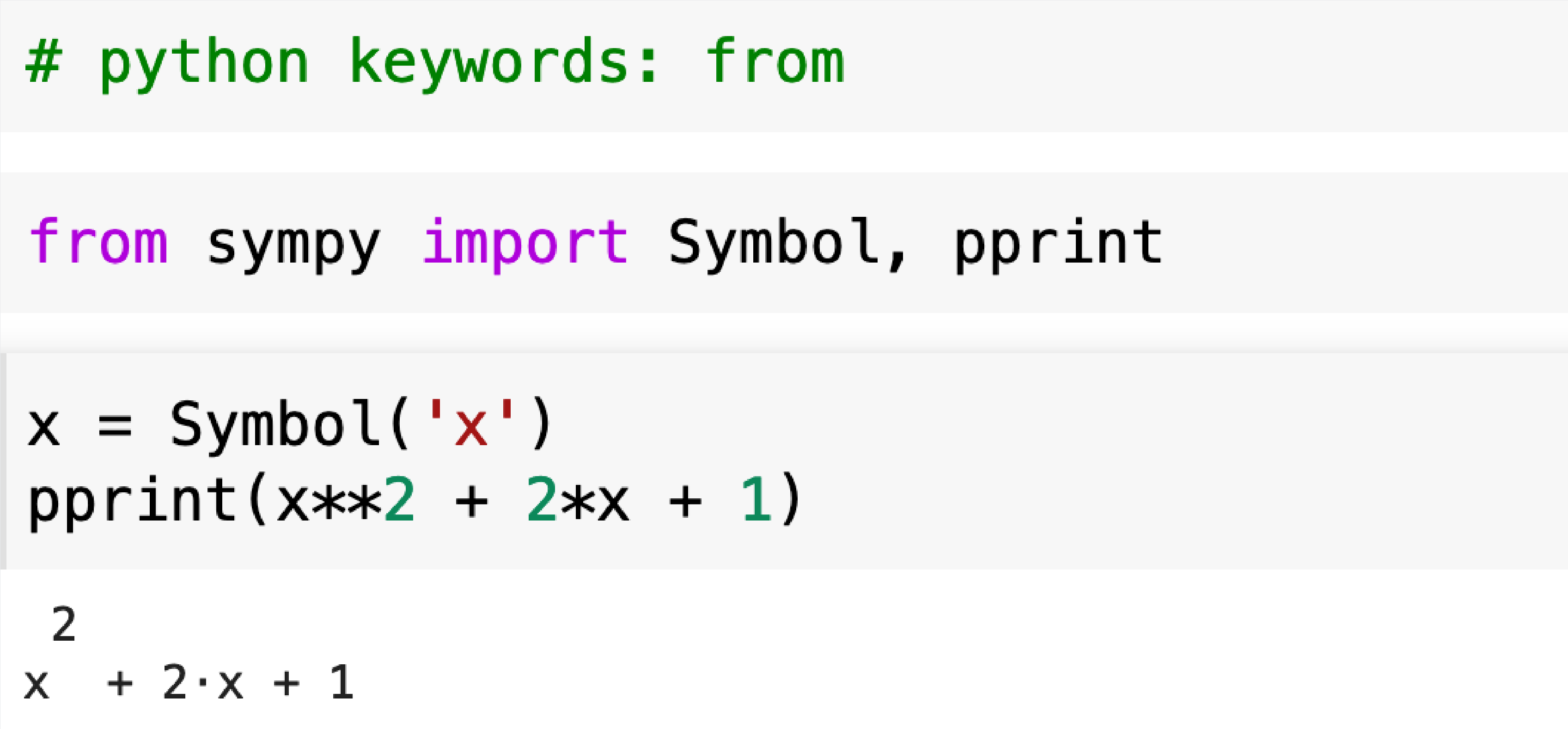
```
# python keywords: for
```

```
for i in range(1, 10, 2):
  print(i)
```

```
1
3
5
7
9
```

Python 3.x reserved words

from



```
# python keywords: from
```

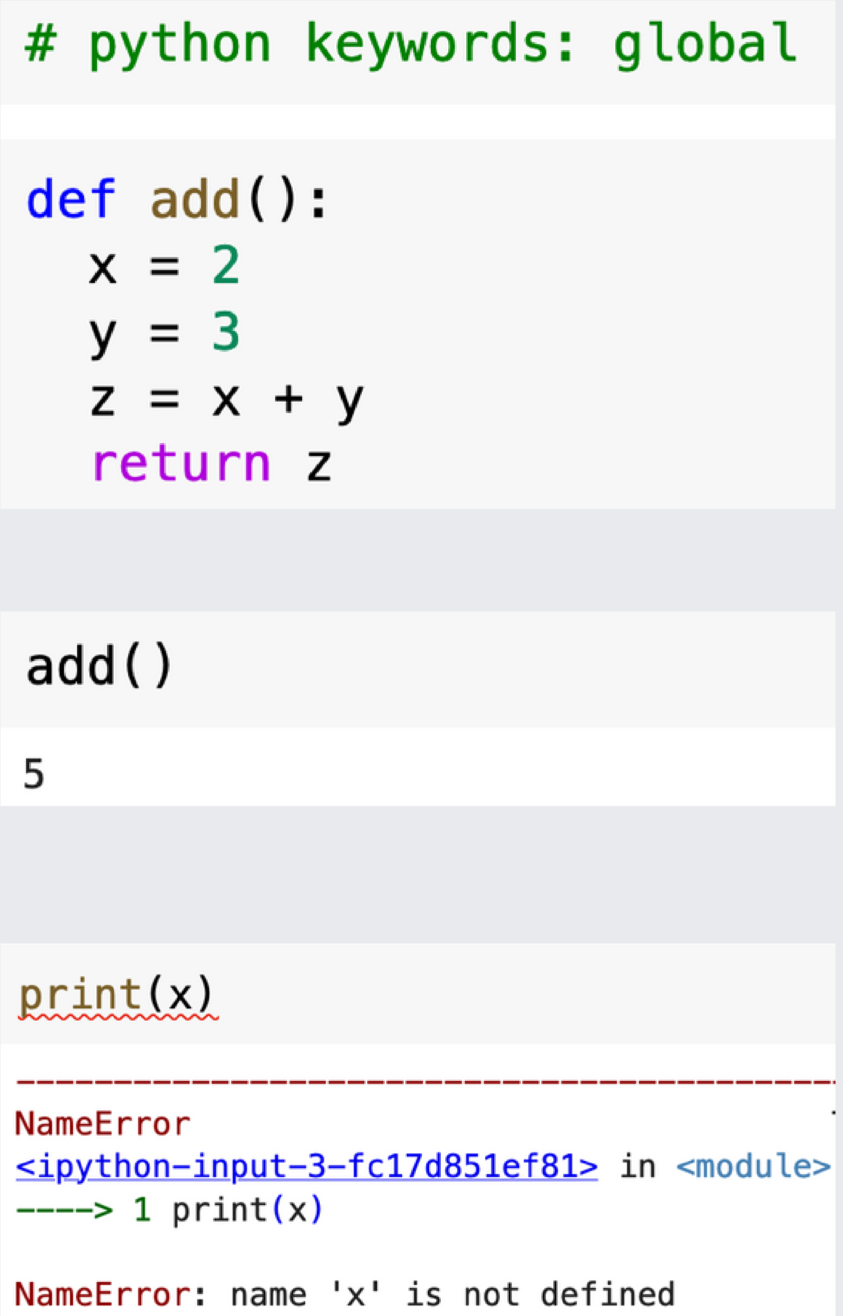
```
from sympy import Symbol, pprint
```

```
x = Symbol('x')
pprint(x**2 + 2*x + 1)
```

```
 2
x  + 2⋅x + 1
```

Python 3.x reserved words

global



```
# python keywords: global
```

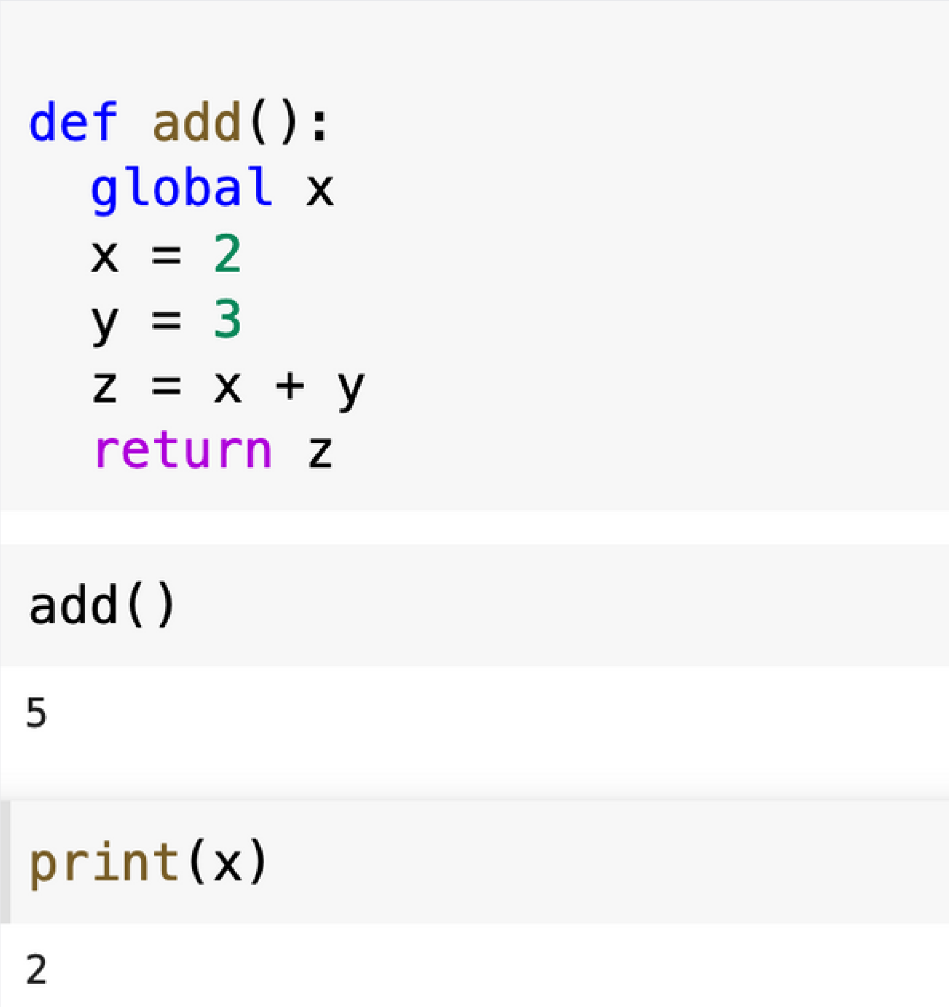
```
def add():
  x = 2
  y = 3
  z = x + y
  return z
```

```
add()
```

```
5
```

```
print(x)
```

```
-----------------------------------------
NameError
<ipython-input-3-fc17d851ef81> in <module>
----> 1 print(x)

NameError: name 'x' is not defined
```



```
def add():
  global x
  x = 2
  y = 3
  z = x + y
  return z
```

```
add()
```

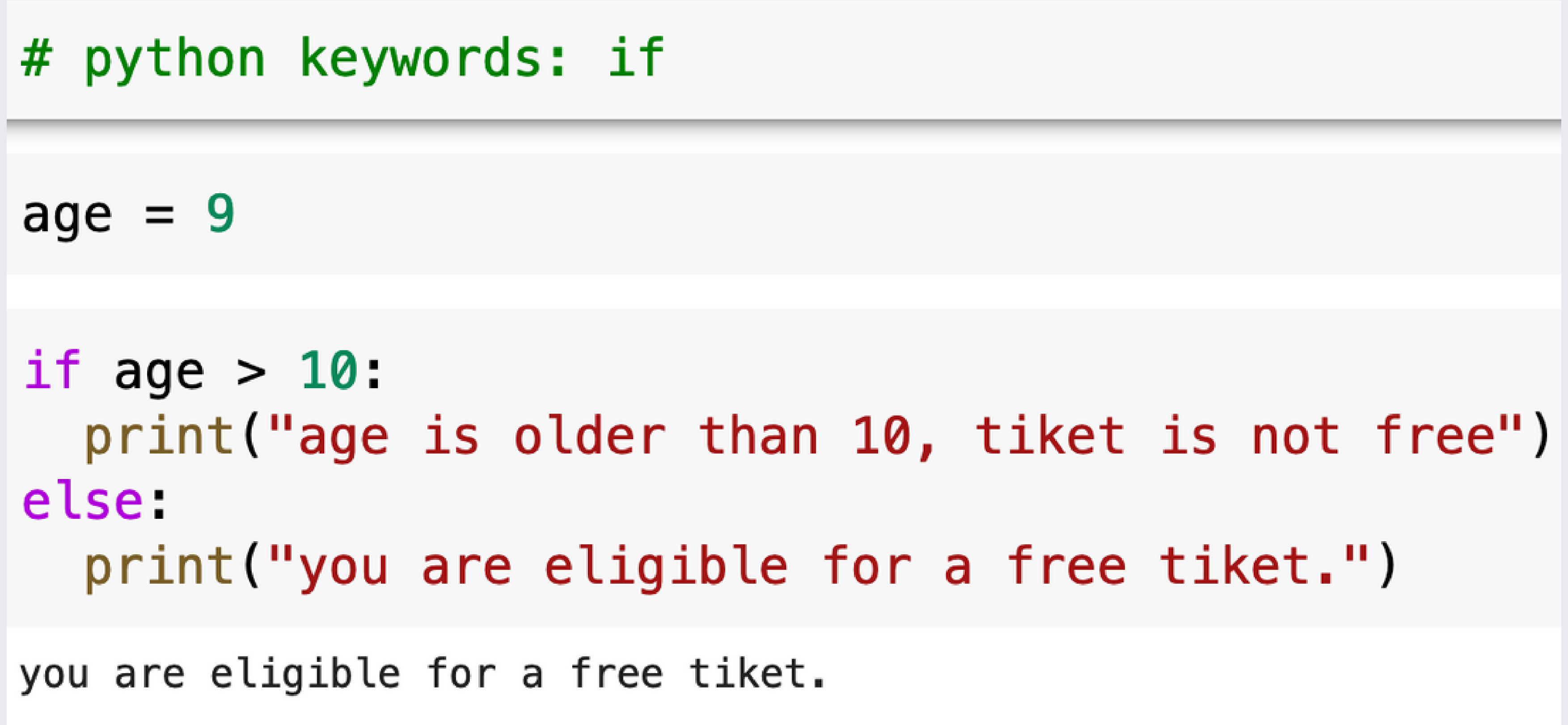
```
5
```

```
print(x)
```

```
2
```

Python 3.x reserved words

if



```python
# python keywords: if
```

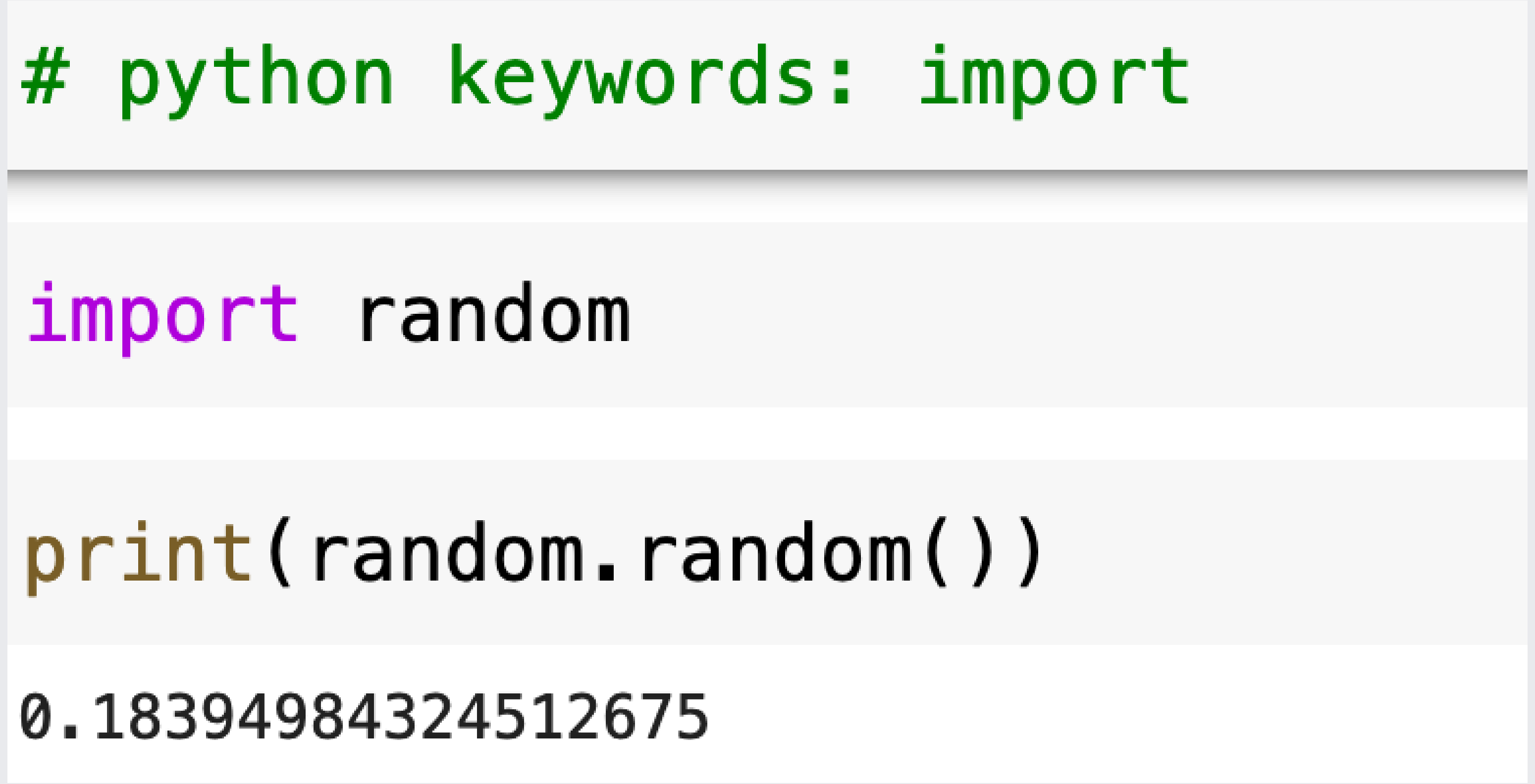
```python
age = 9
```

```python
if age > 10:
  print("age is older than 10, tiket is not free")
else:
  print("you are eligible for a free tiket.")
```

```
you are eligible for a free tiket.
```

Python 3.x reserved words

import

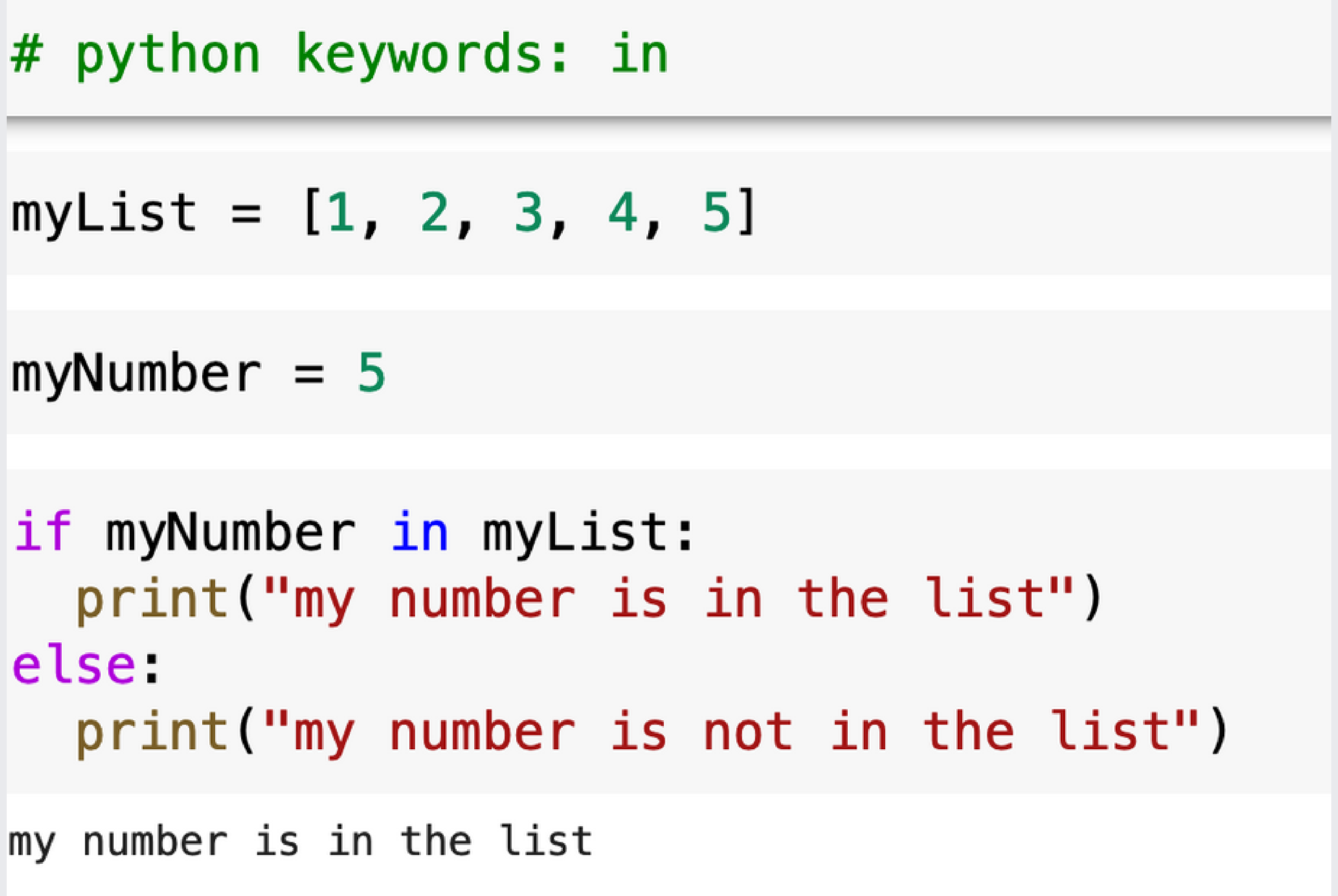


```
# python keywords: import

import random

print(random.random())
```

```
0.18394984324512675
```

Python 3.x reserved words

in



```
# python keywords: in
```

```
myList = [1, 2, 3, 4, 5]
```

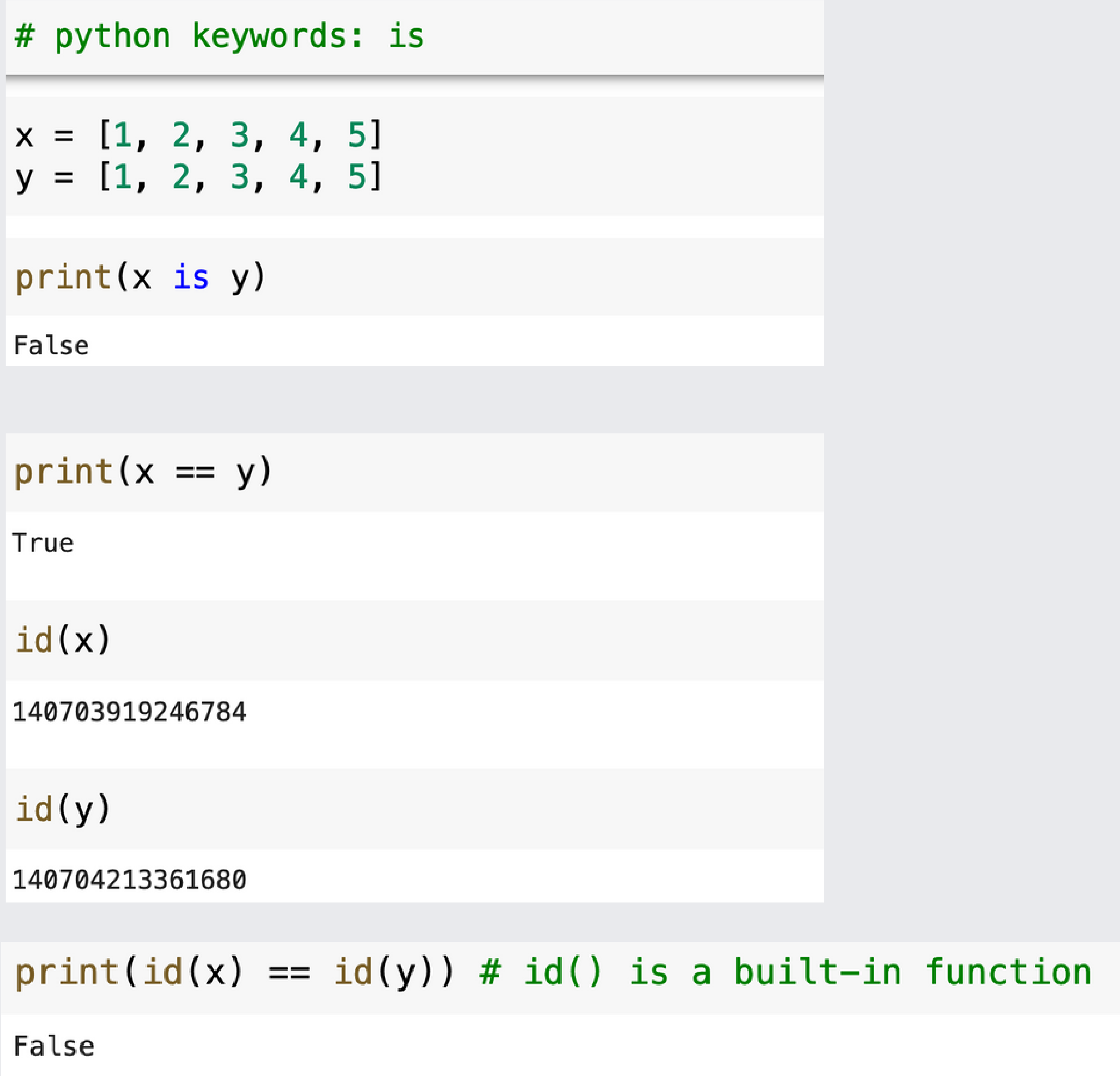
```
myNumber = 5
```

```
if myNumber in myList:
  print("my number is in the list")
else:
  print("my number is not in the list")
```

```
my number is in the list
```

Python 3.x reserved words

is



```
# python keywords: is
```

```
x = [1, 2, 3, 4, 5]
y = [1, 2, 3, 4, 5]
```

```
print(x is y)
```

```
False
```

```
print(x == y)
```

```
True
```

```
id(x)
```

```
140703919246784
```

```
id(y)
```

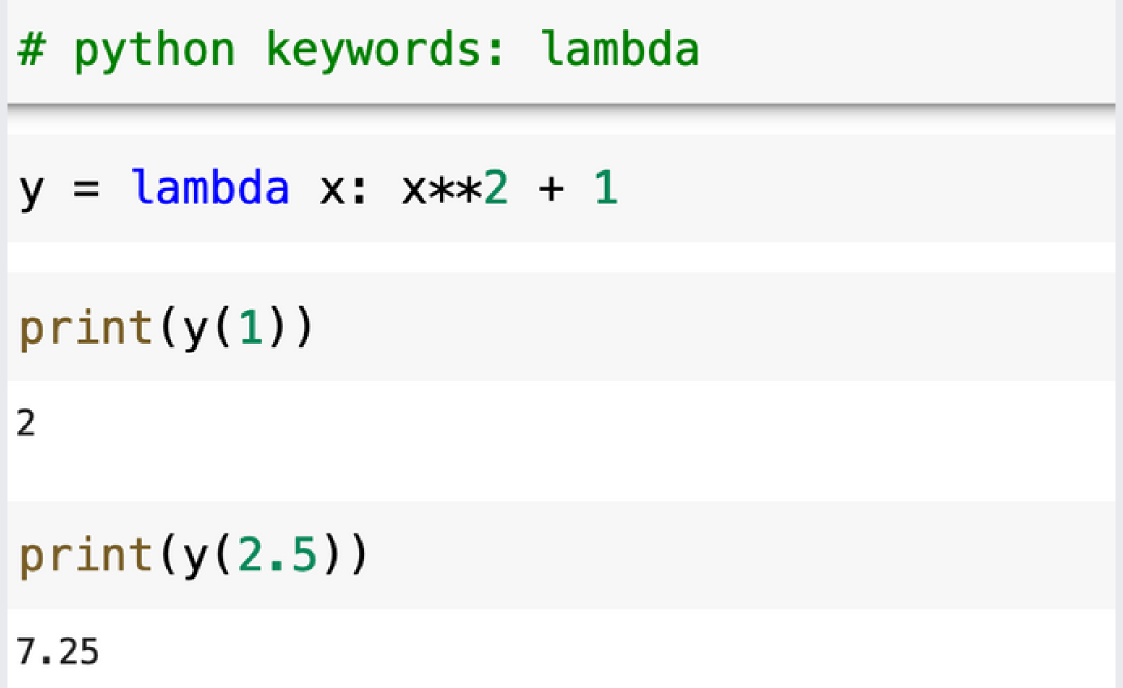
```
140704213361680
```

```
print(id(x) == id(y)) # id() is a built-in function
```

```
False
```

Python 3.x reserved words

lambda

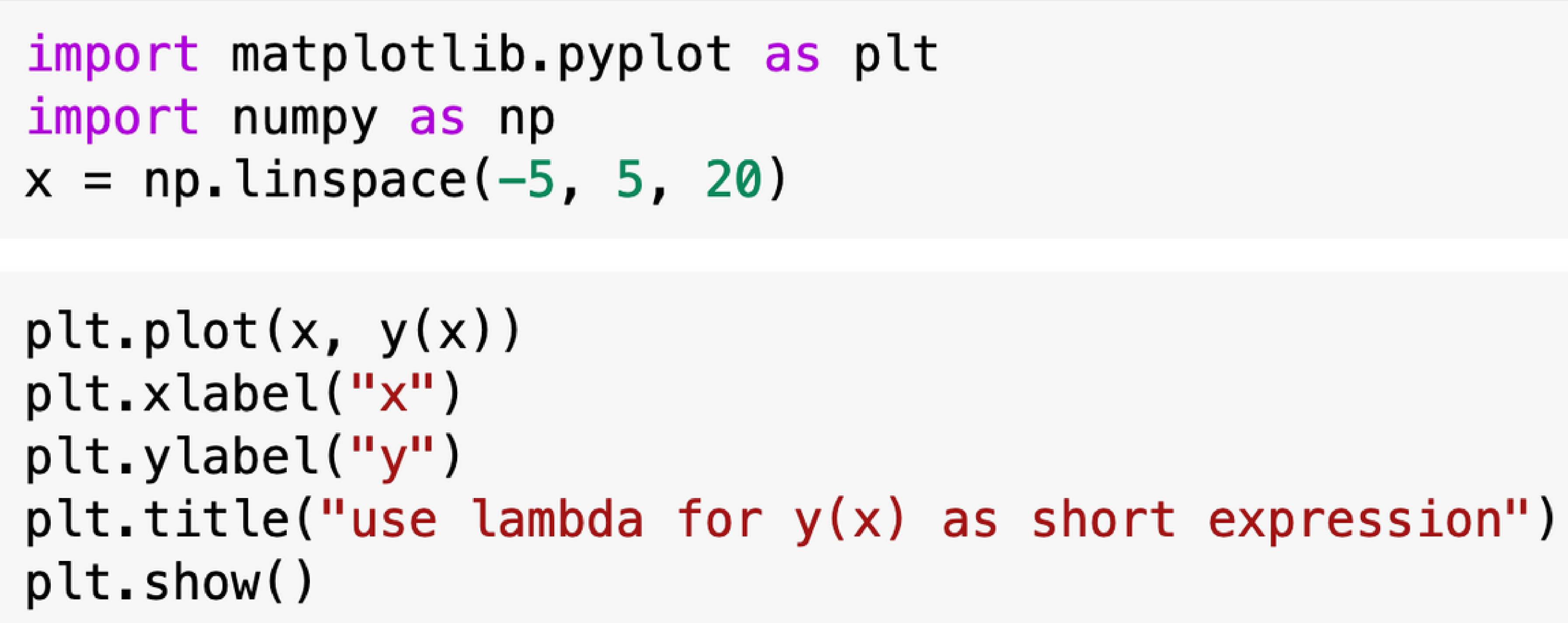


```
# python keywords: lambda

y = lambda x: x**2 + 1

print(y(1))
```

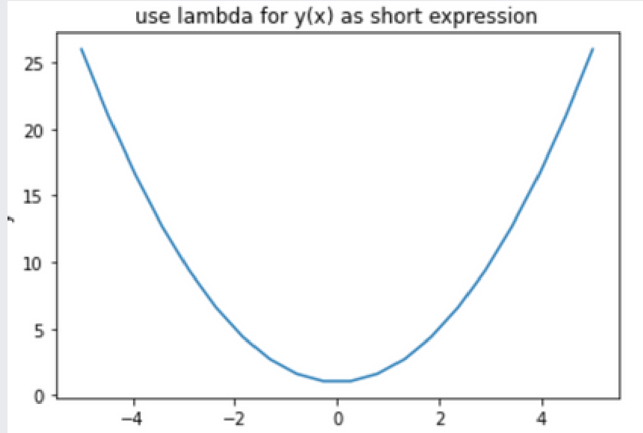
```
2
```

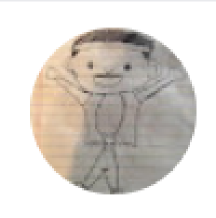
```
print(y(2.5))
```

```
7.25
```



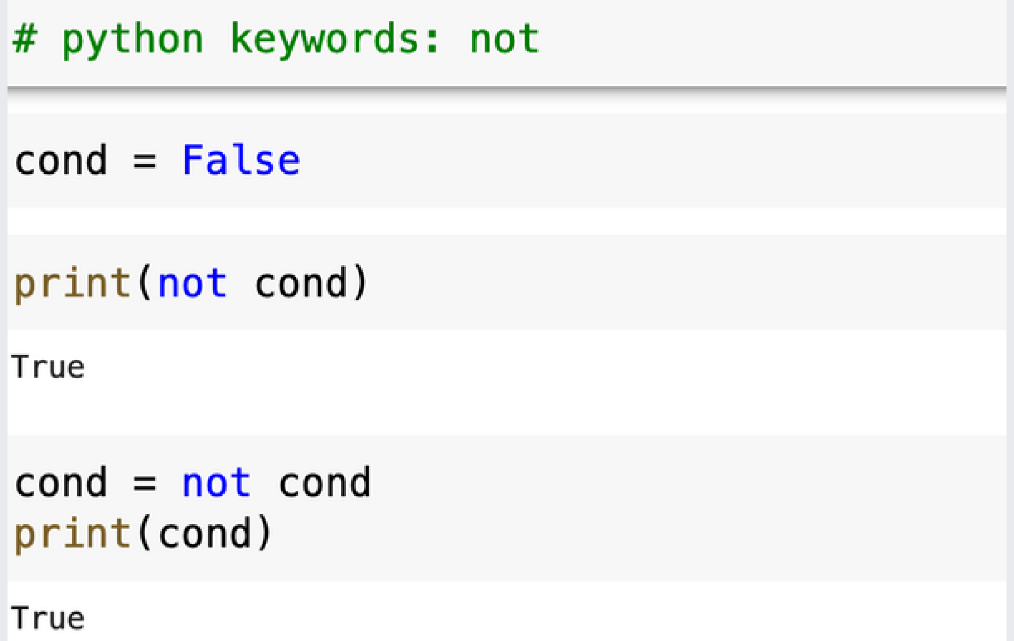
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-5, 5, 20)

plt.plot(x, y(x))
plt.xlabel("x")
plt.ylabel("y")
plt.title("use lambda for y(x) as short expression")
plt.show()
```





Python 3.x reserved words

not



```
# python keywords: not
```

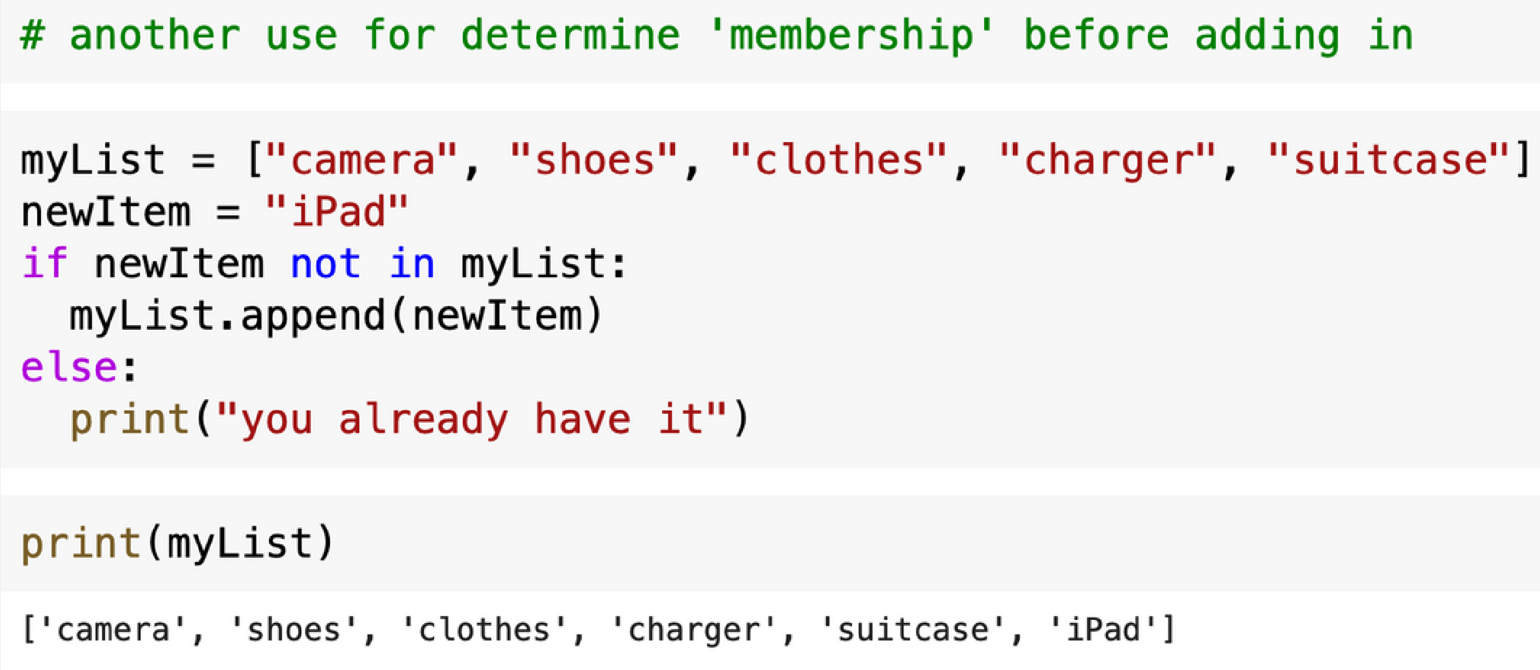
```
cond = False
```

```
print(not cond)
```

```
True
```

```
cond = not cond
print(cond)
```

```
True
```

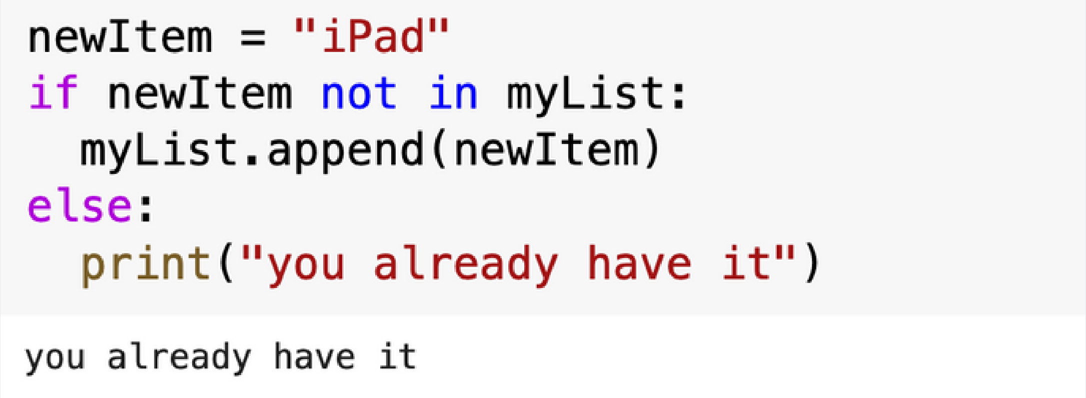


```
# another use for determine 'membership' before adding in
```

```
myList = ["camera", "shoes", "clothes", "charger", "suitcase"]
newItem = "iPad"
if newItem not in myList:
  myList.append(newItem)
else:
  print("you already have it")
```

```
print(myList)
```

```
['camera', 'shoes', 'clothes', 'charger', 'suitcase', 'iPad']
```

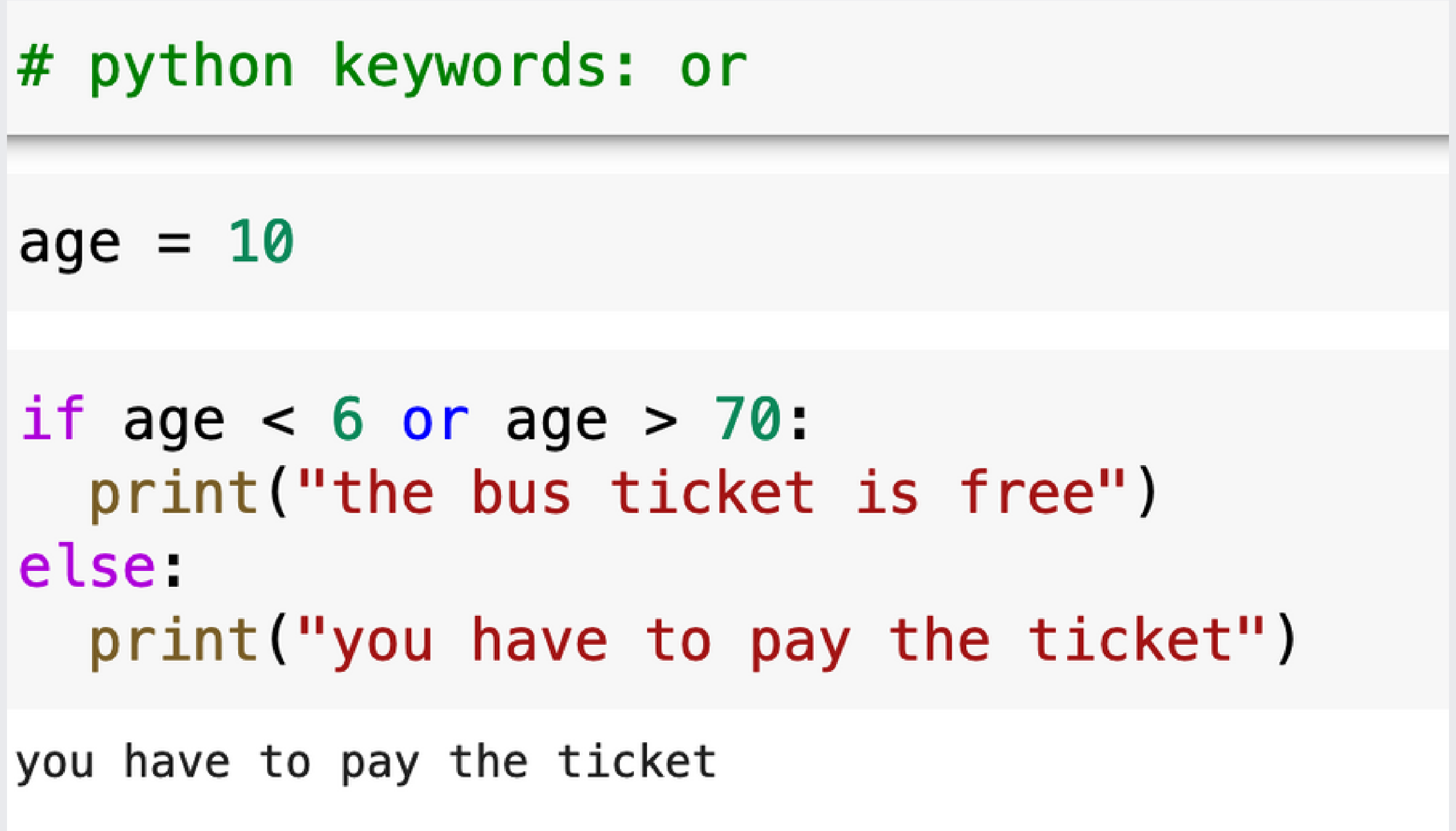


```
newItem = "iPad"
if newItem not in myList:
  myList.append(newItem)
else:
  print("you already have it")
```

```
you already have it
```

Python 3.x reserved words

or



```
# python keywords: or
```

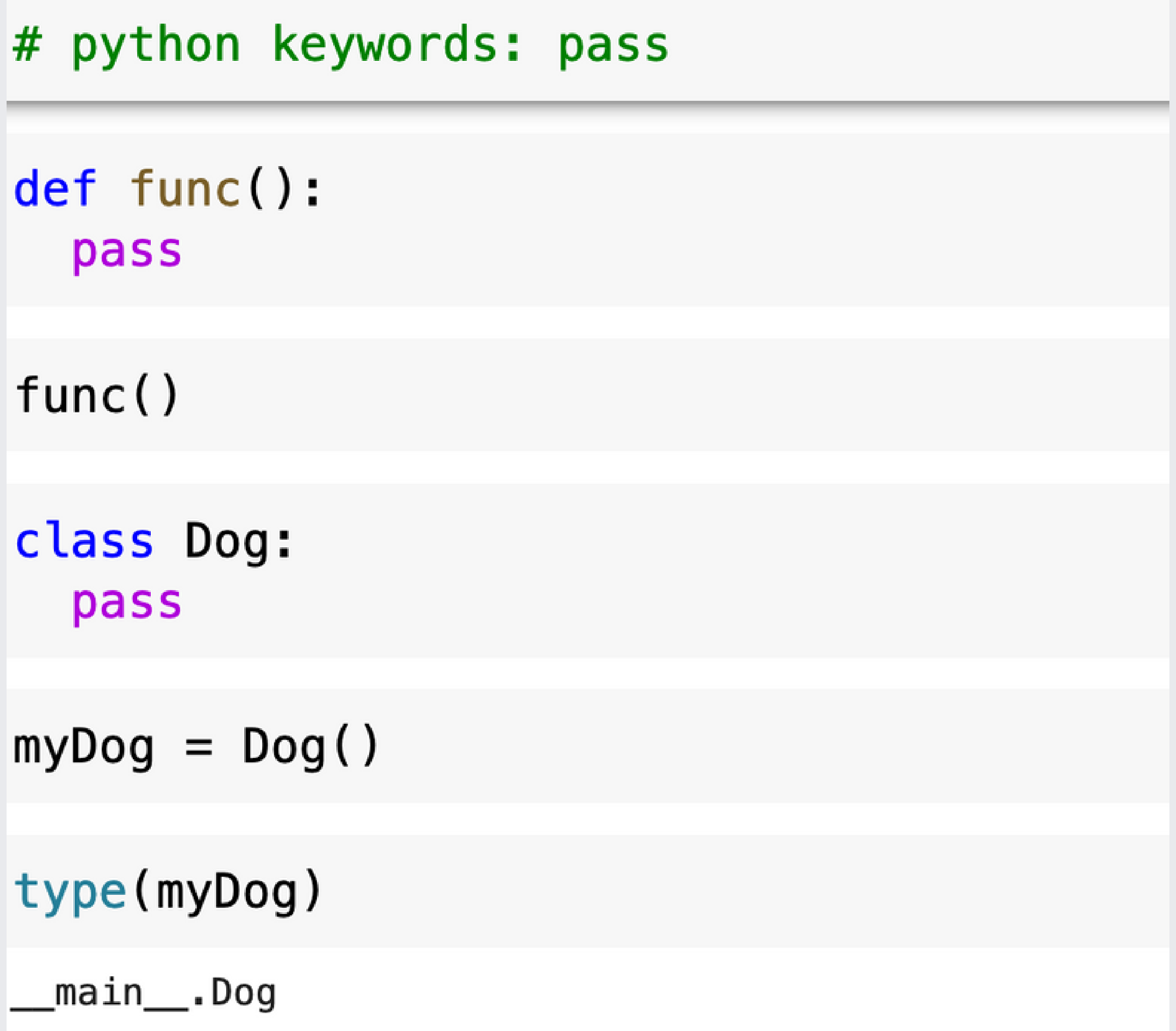
```
age = 10
```

```
if age < 6 or age > 70:
  print("the bus ticket is free")
else:
  print("you have to pay the ticket")
```

```
you have to pay the ticket
```

Python 3.x reserved words

pass



```
# python keywords: pass
```

```
def func():
  pass
```

```
func()
```

```
class Dog:
  pass
```

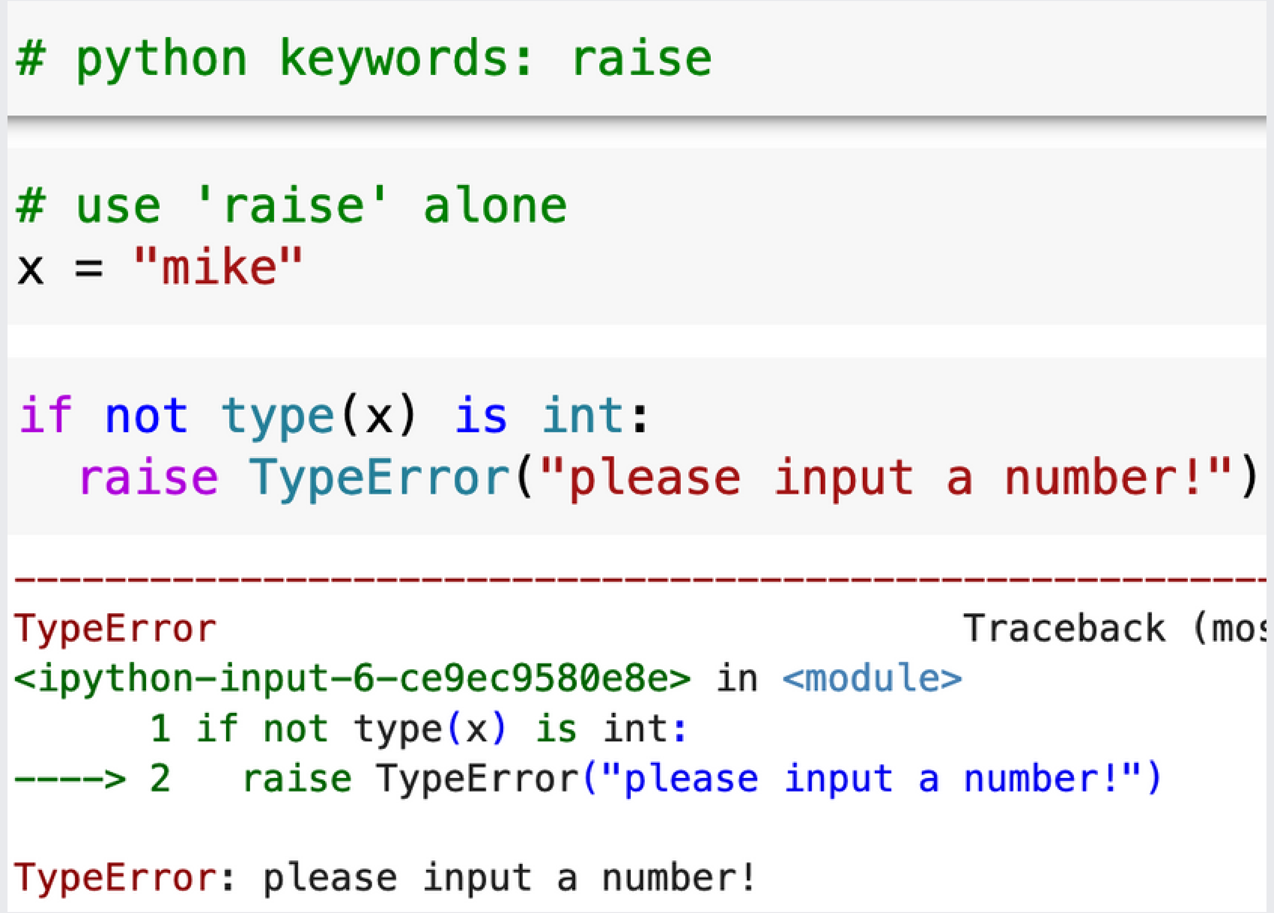
```
myDog = Dog()
```

```
type(myDog)
```

```
__main__.Dog
```

Python 3.x reserved words

raise

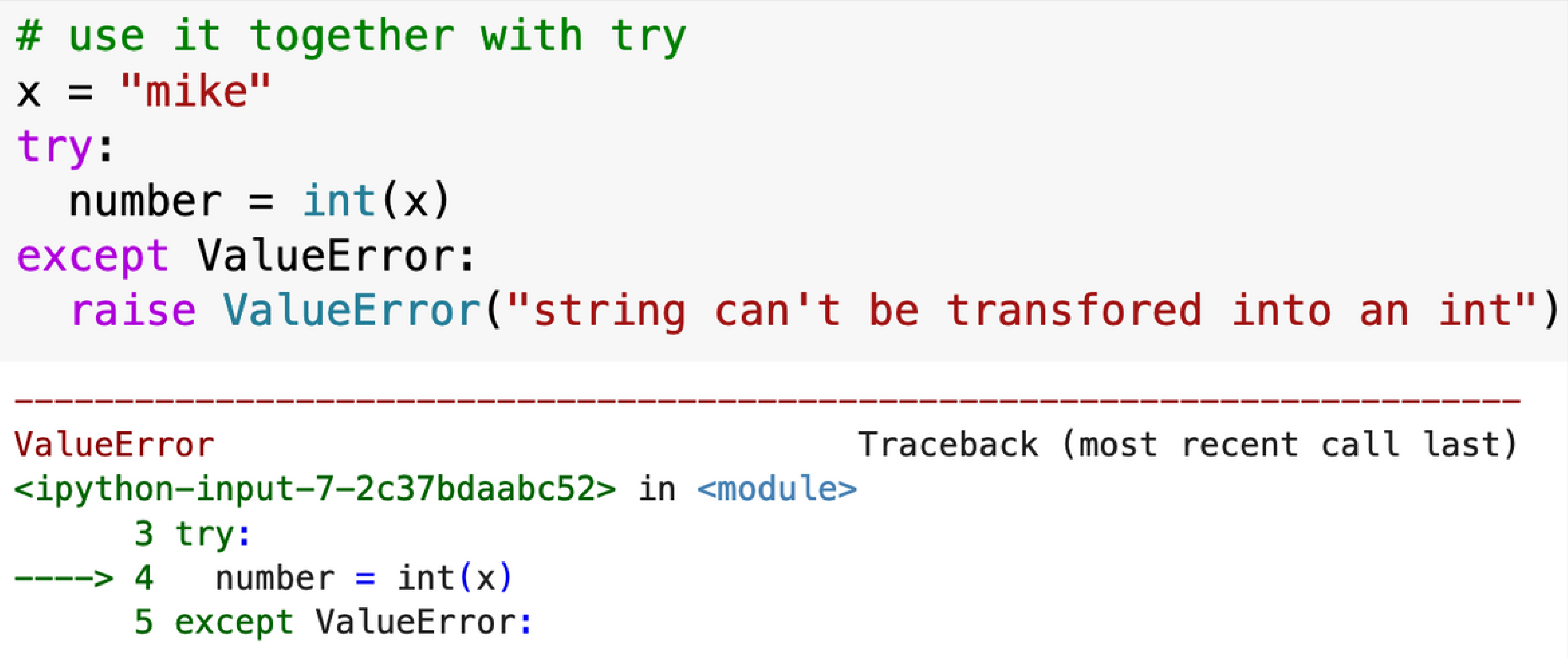


```
# python keywords: raise
```

```
# use 'raise' alone
x = "mike"
```

```
if not type(x) is int:
  raise TypeError("please input a number!")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (mos
<ipython-input-6-ce9ec9580e8e> in <module>
      1 if not type(x) is int:
----> 2   raise TypeError("please input a number!")

TypeError: please input a number!
```

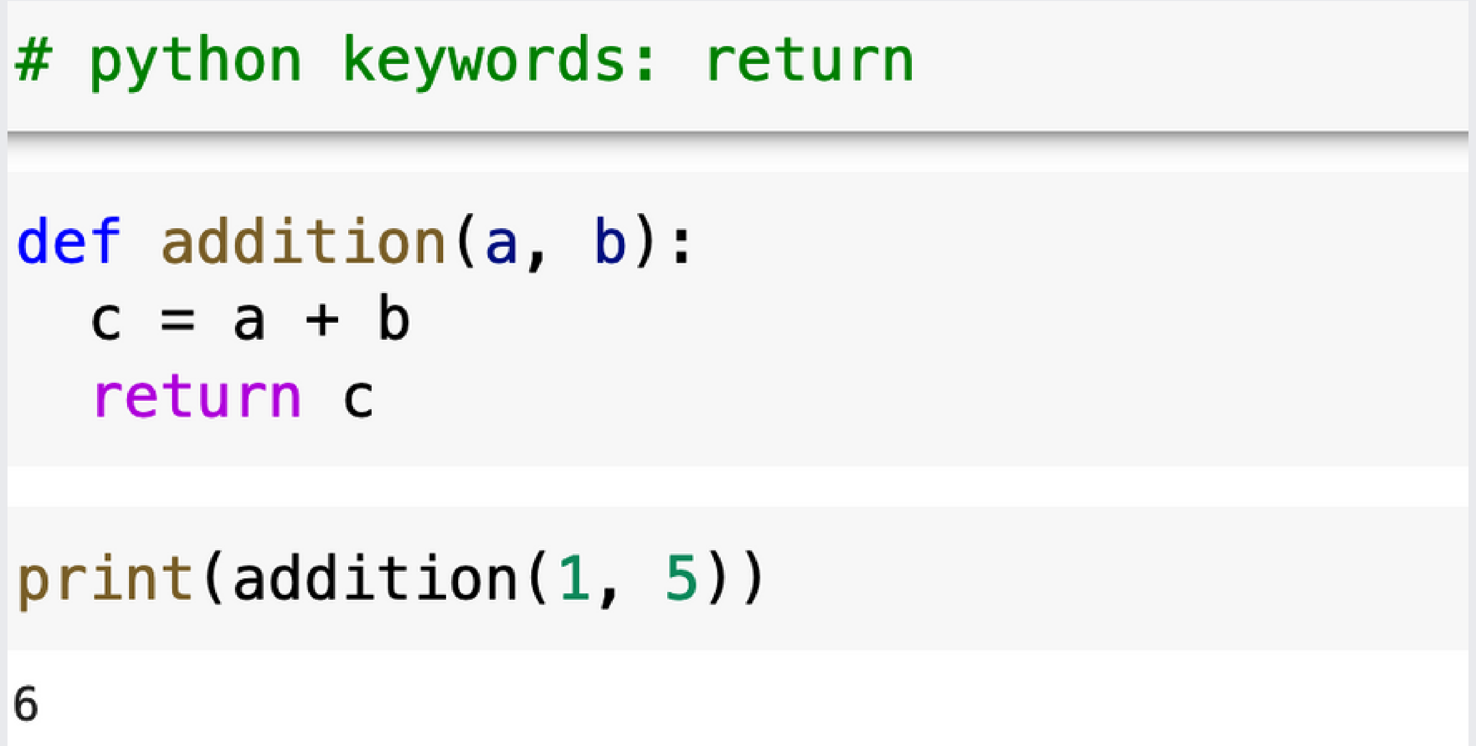


```
# use it together with try
x = "mike"
try:
  number = int(x)
except ValueError:
  raise ValueError("string can't be transfored into an int")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-7-2c37bdaabc52> in <module>
      3 try:
----> 4   number = int(x)
      5 except ValueError:
```

Python 3.x reserved words

return

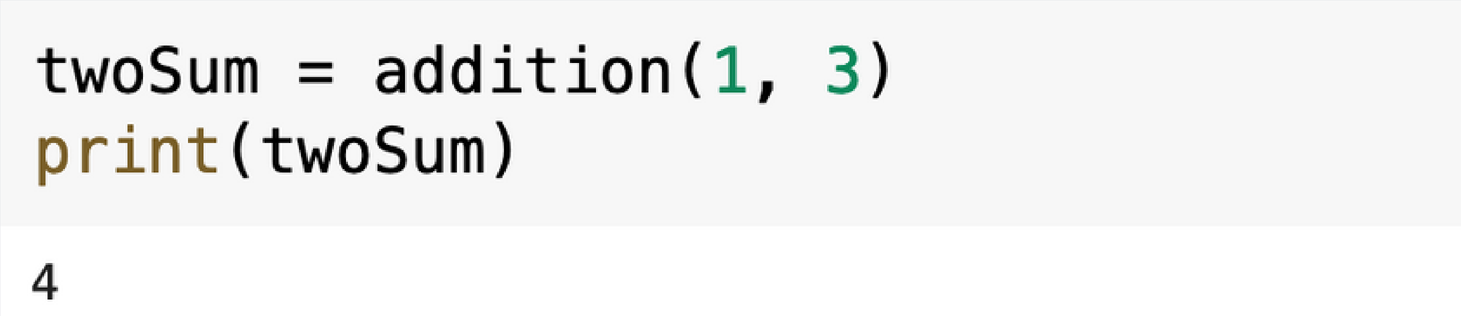


```
# python keywords: return
```

```
def addition(a, b):
  c = a + b
  return c
```

```
print(addition(1, 5))
```

```
6
```

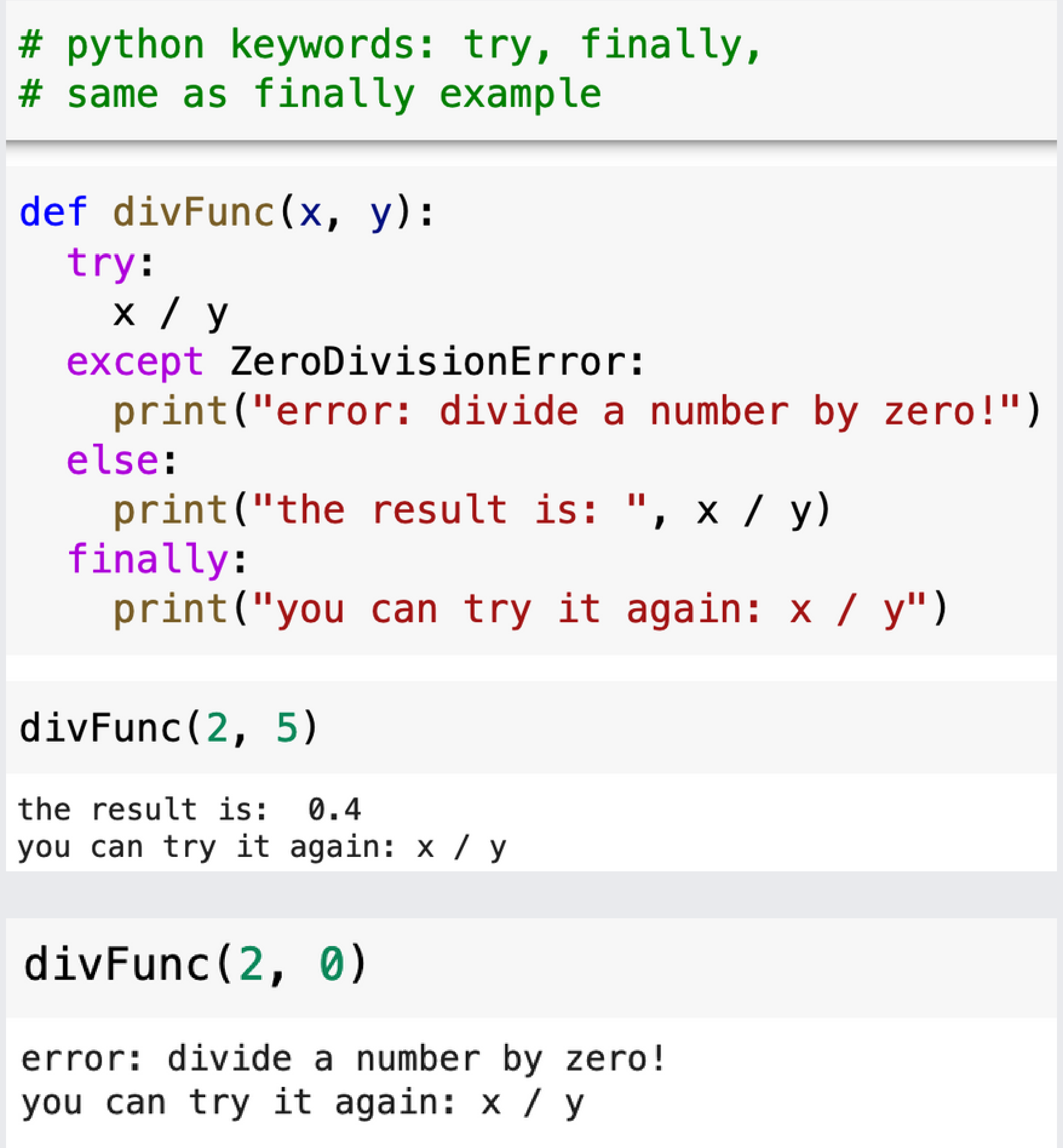


```
twoSum = addition(1, 3)
print(twoSum)
```

```
4
```

Python 3.x reserved words

try



```python
# python keywords: try, finally,
# same as finally example
```

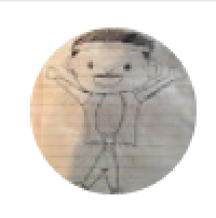
```python
def divFunc(x, y):
  try:
    x / y
  except ZeroDivisionError:
    print("error: divide a number by zero!")
  else:
    print("the result is: ", x / y)
  finally:
    print("you can try it again: x / y")
```

```python
divFunc(2, 5)
```

```
the result is:  0.4
you can try it again: x / y
```

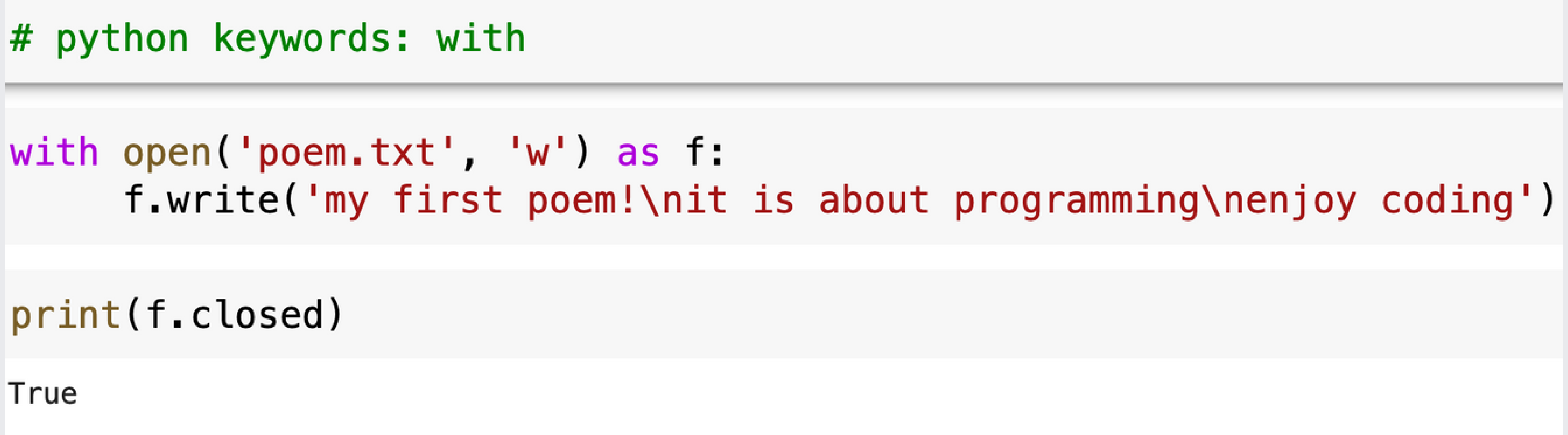
```python
divFunc(2, 0)
```

```
error: divide a number by zero!
you can try it again: x / y
```



Python 3.x reserved words

with

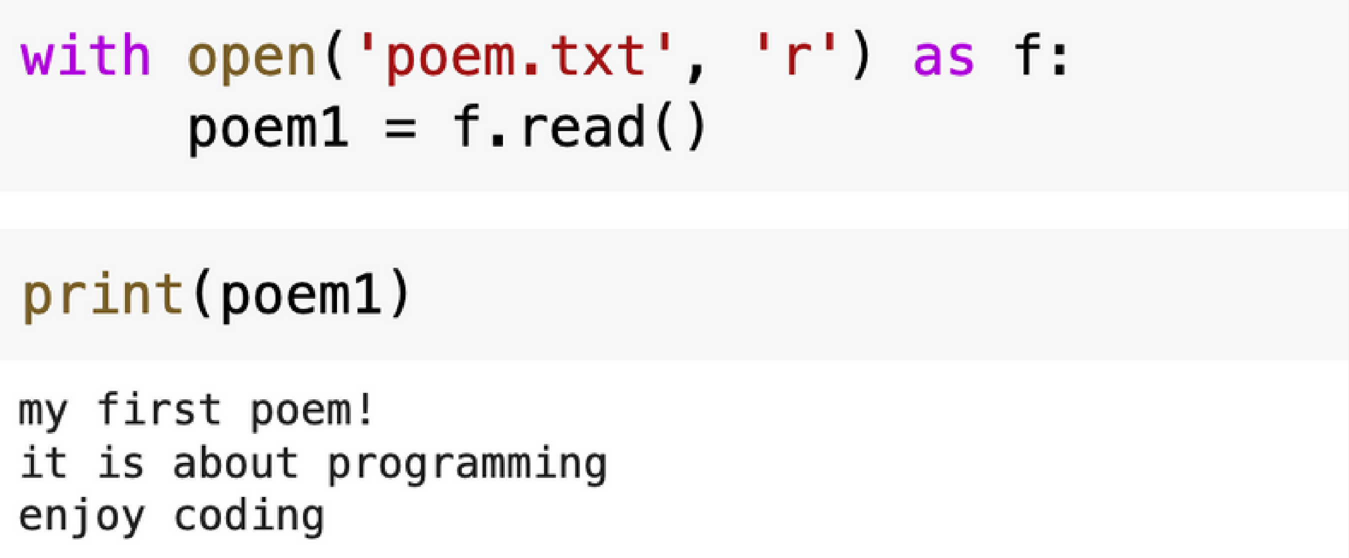


```
# python keywords: with
```

```
with open('poem.txt', 'w') as f:
     f.write('my first poem!\nit is about programming\nenjoy coding')
```

```
print(f.closed)
```

```
True
```



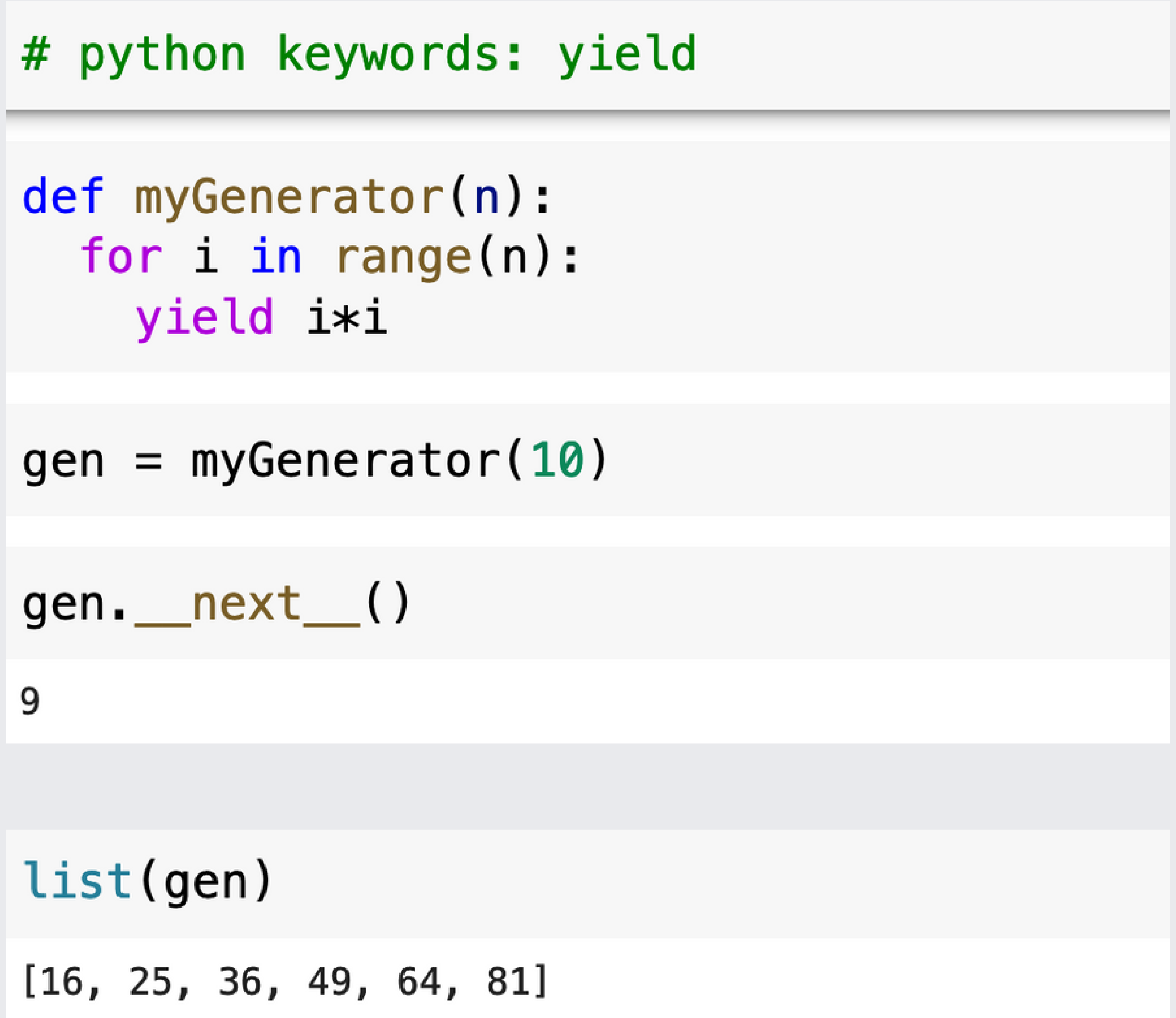
```
with open('poem.txt', 'r') as f:
     poem1 = f.read()
```

```
print(poem1)
```

```
my first poem!
it is about programming
enjoy coding
```

Python 3.x reserved words

yield



```
# python keywords: yield
```

```
def myGenerator(n):
  for i in range(n):
    yield i*i
```

```
gen = myGenerator(10)
```

```
gen.__next__()
```

```
9
```

```
list(gen)
```

```
[16, 25, 36, 49, 64, 81]
```

Python 3.x reserved words

THANK YOU!