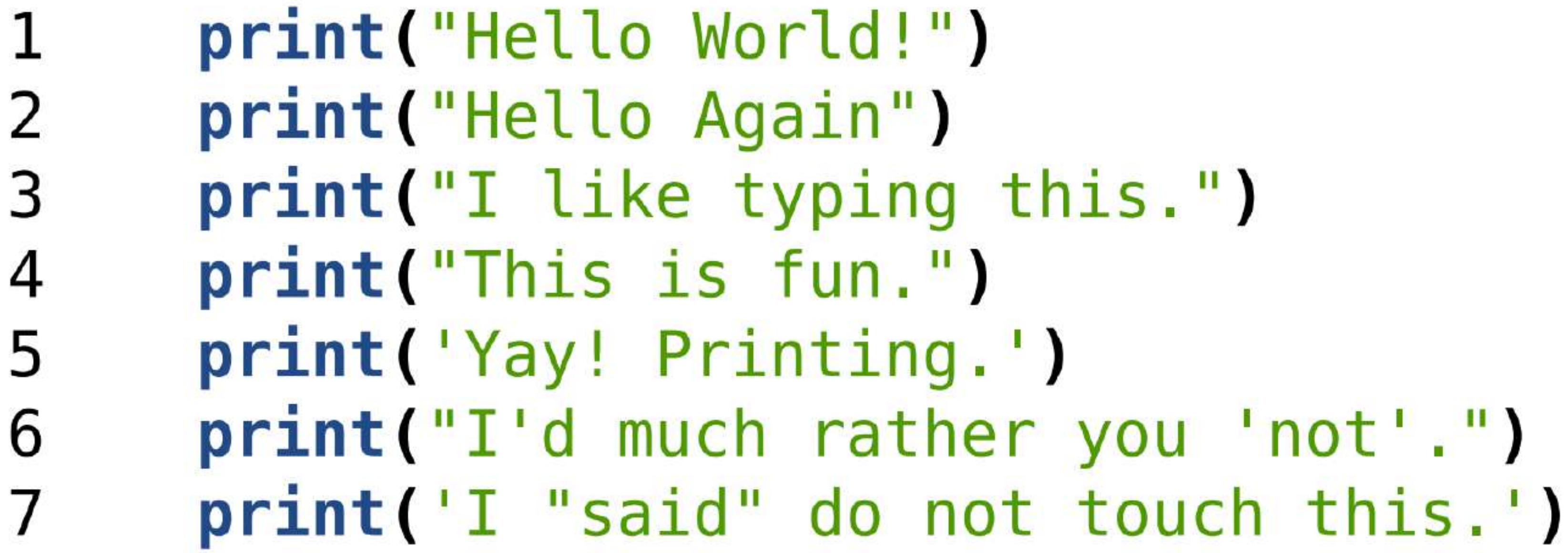
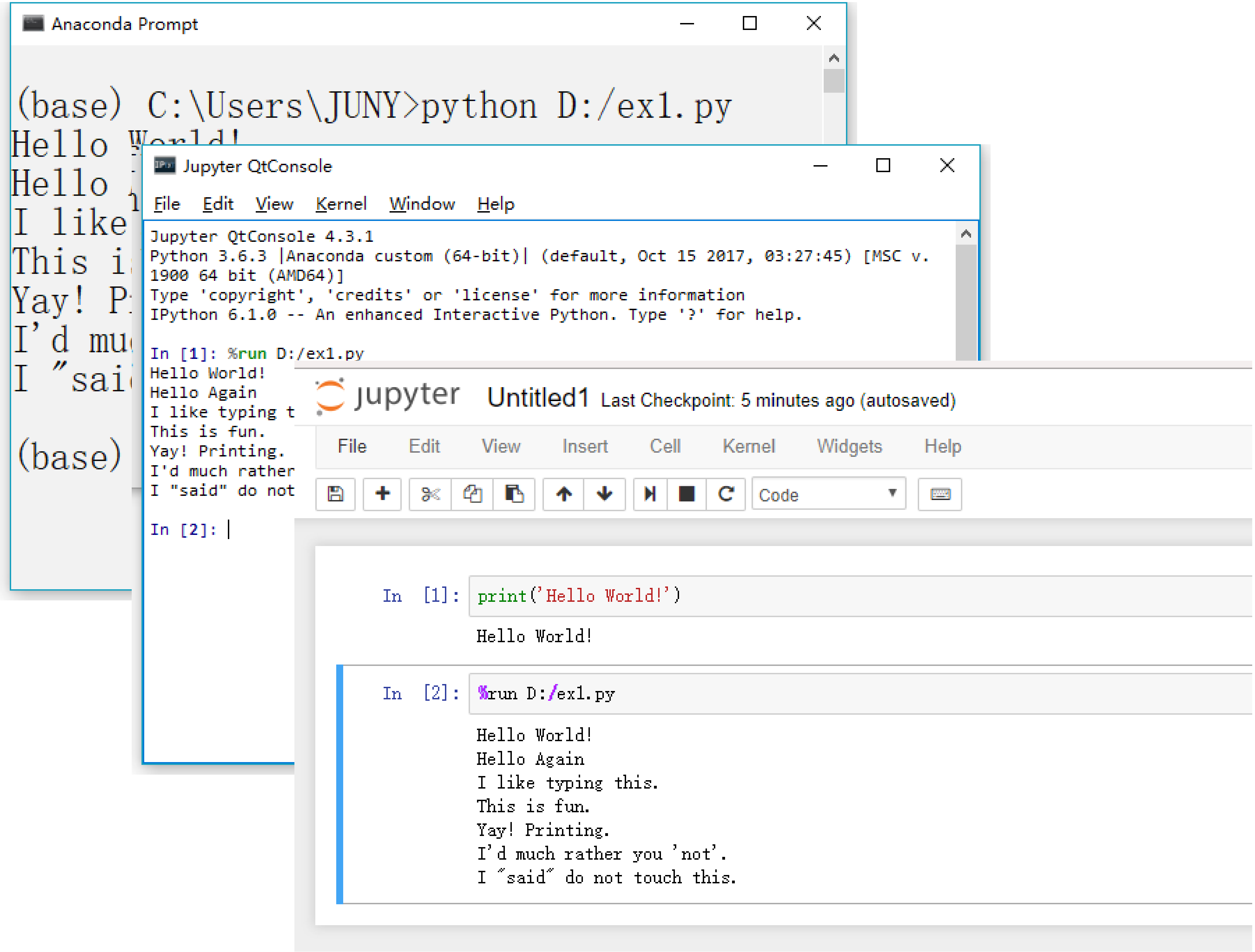
Exercises 上机练习

Exercise 1: 字符串



```python
print("Hello World!")
print("Hello Again")
print("I like typing this.")
print("This is fun.")
print('Yay! Printing.')
print("I'd much rather you 'not'.")
print('I "said" do not touch this.')
```

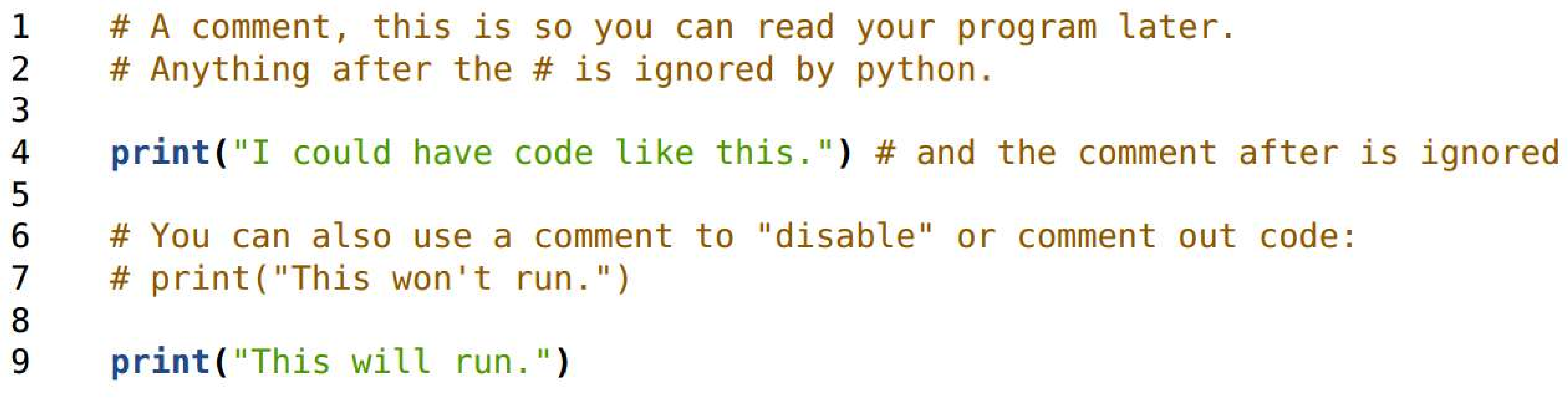


```
(base) C:\Users\JUNY>python D:/ex1.py
Hello World!
Hello
I like
This i
Yay! P
I'd mu
I "sai

(base)
```

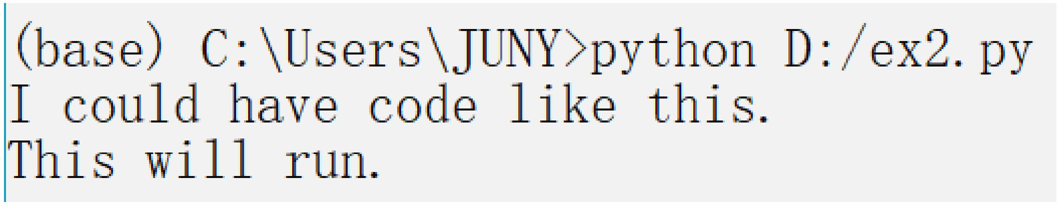
```
Jupyter QtConsole 4.3.1
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct 15 2017, 03:27:45) [MSC v.
1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %run D:/ex1.py
Hello World!
Hello Again
I like typing t
This is fun.
Yay! Printing.
I'd much rather
I "said" do not

In [2]:
```

```
In [1]: print('Hello World!')

Hello World!

In [2]: %run D:/ex1.py

Hello World!
Hello Again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you 'not'.
I "said" do not touch this.
```

Exercise 2: 注释

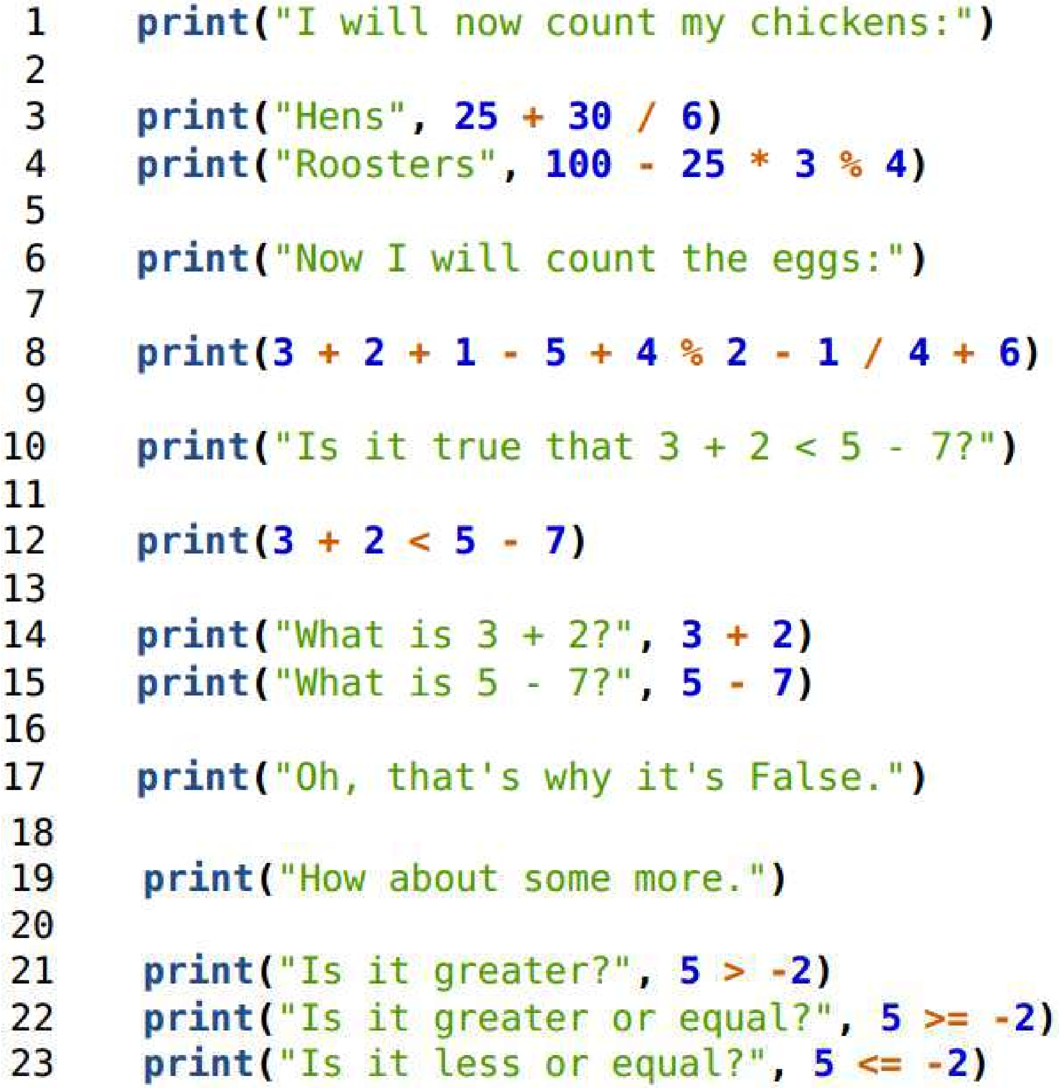


```python
# A comment, this is so you can read your program later.
# Anything after the # is ignored by python.

print("I could have code like this.") # and the comment after is ignored

# You can also use a comment to "disable" or comment out code:
# print("This won't run.")

print("This will run.")
```

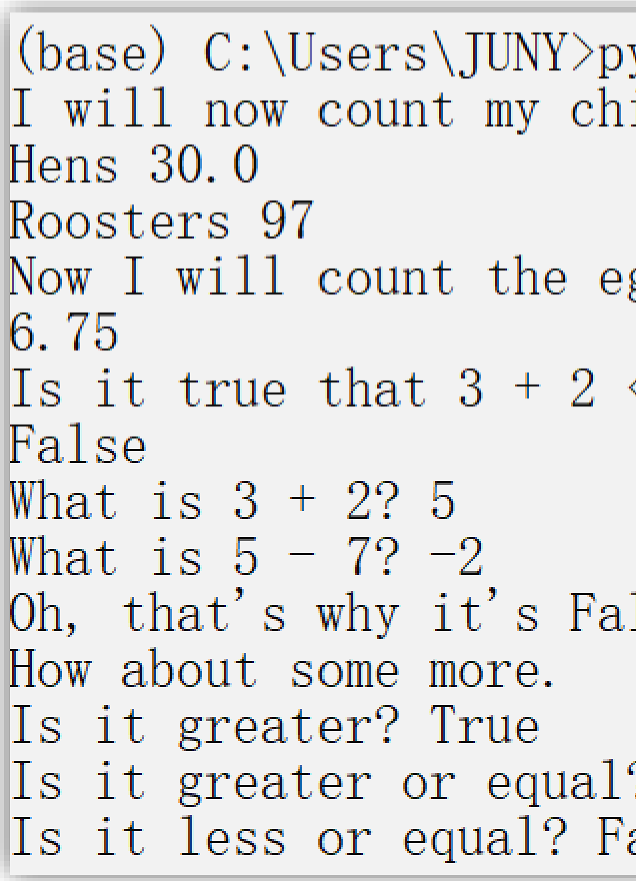


```
(base) C:\Users\JUNY>python D:/ex2.py
I could have code like this.
This will run.
```

Exercise 3: print 2



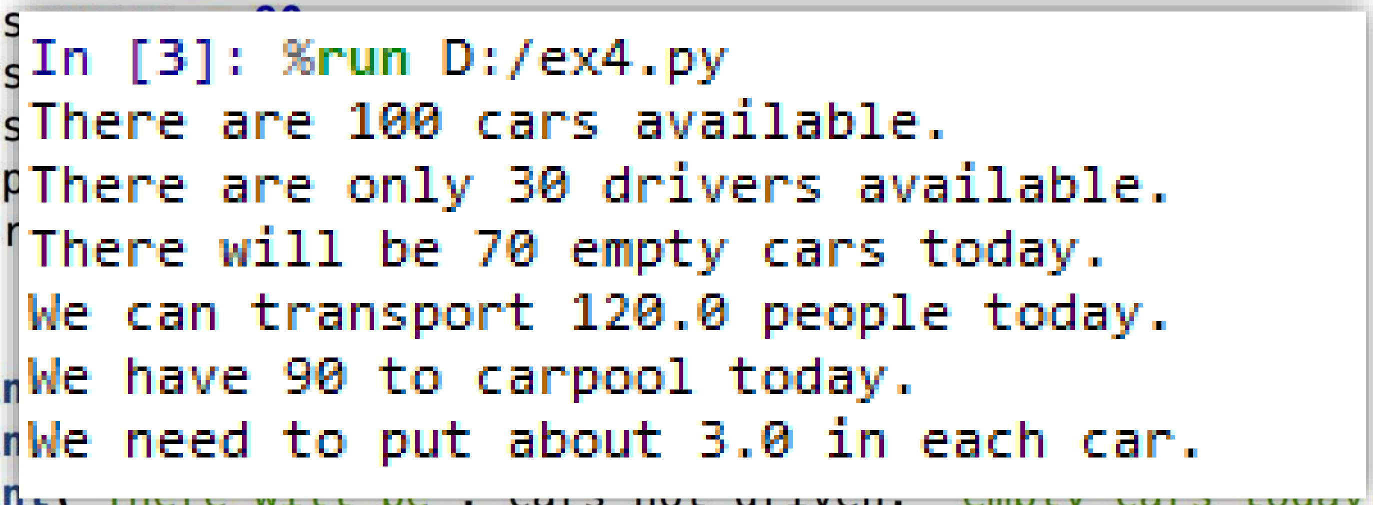
```python
print("I will now count my chickens:")

print("Hens", 25 + 30 / 6)
print("Roosters", 100 - 25 * 3 % 4)

print("Now I will count the eggs:")

print(3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6)

print("Is it true that 3 + 2 < 5 - 7?")

print(3 + 2 < 5 - 7)

print("What is 3 + 2?", 3 + 2)
print("What is 5 - 7?", 5 - 7)

print("Oh, that's why it's False.")

print("How about some more.")

print("Is it greater?", 5 > -2)
print("Is it greater or equal?", 5 >= -2)
print("Is it less or equal?", 5 <= -2)
```



```
(base) C:\Users\JUNY>py
I will now count my chi
Hens 30.0
Roosters 97
Now I will count the eg
6.75
Is it true that 3 + 2 <
False
What is 3 + 2? 5
What is 5 - 7? -2
Oh, that's why it's Fal
How about some more.
Is it greater? True
Is it greater or equal?
Is it less or equal? Fa
```

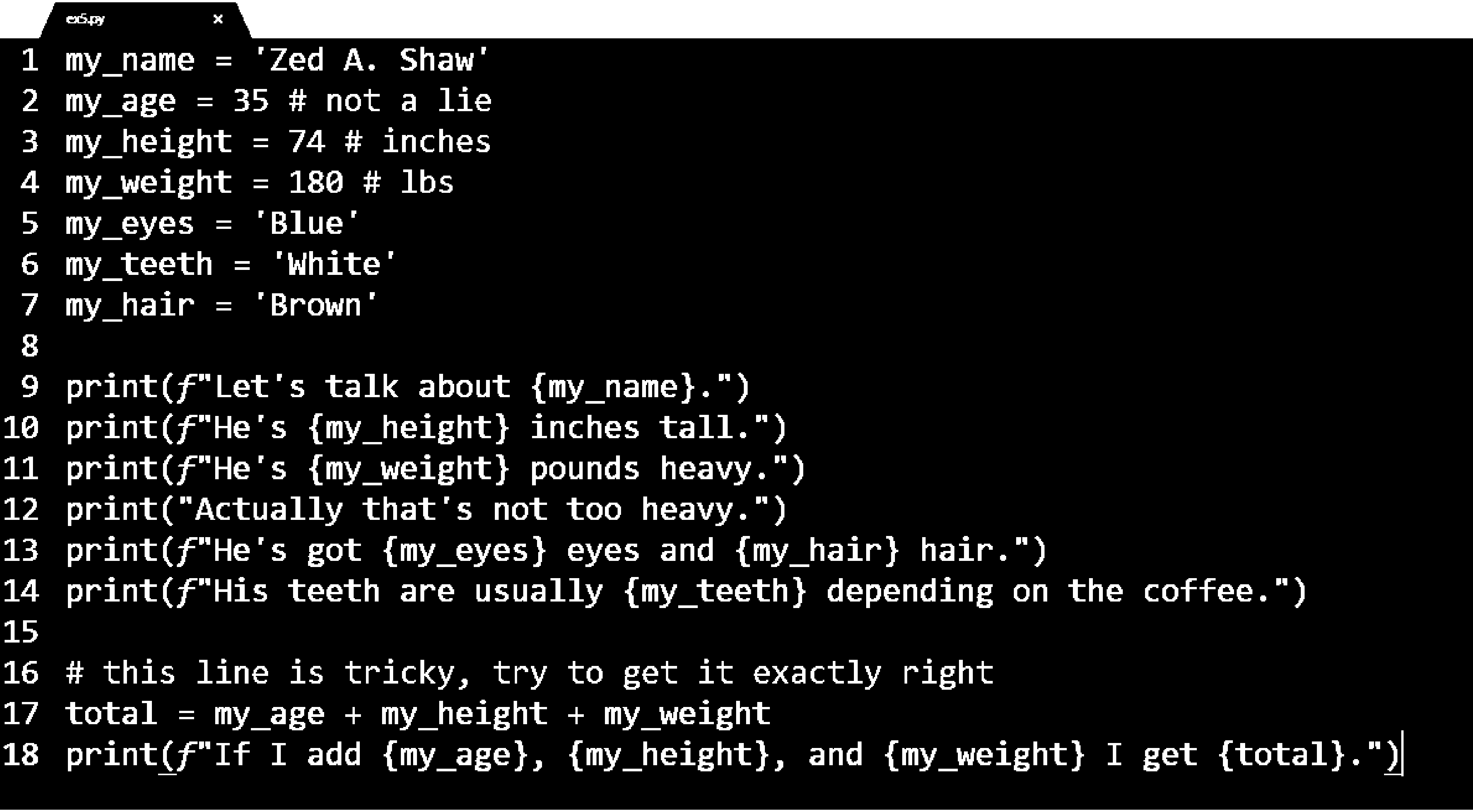
Exercise 4: 变量

```
cars = 100
space_in_a_car = 4.0
drivers = 30
pass
cars
cars
carp
aver


prin
prin
print("There will be", cars_not_driven, "empty cars today.")
print("We can transport", carpool_capacity, "people today.")
print("We have", passengers, "to carpool today.")
print("We need to put about", average_passengers_per_car,
      "in each car.")
```



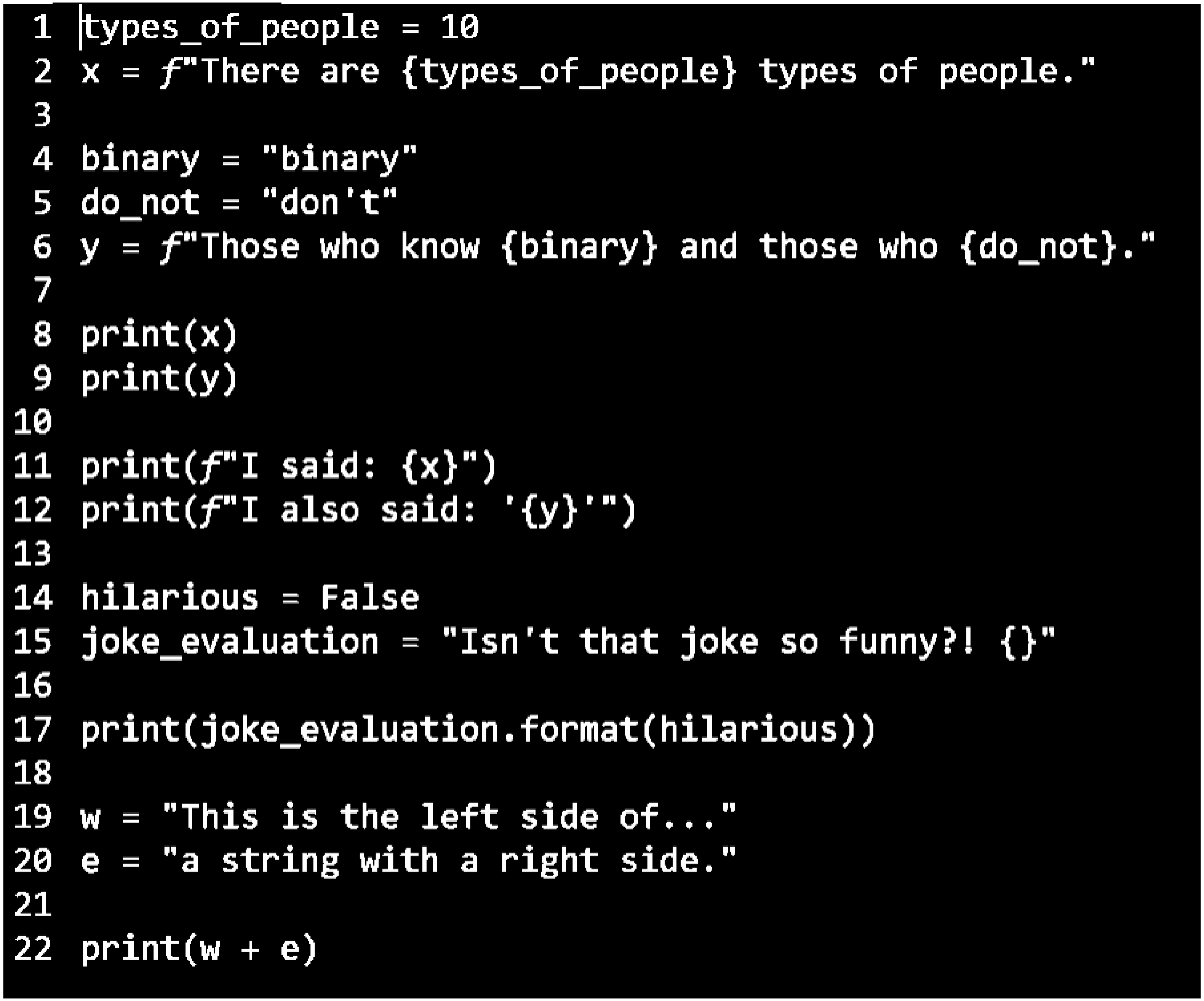
```
In [3]: %run D:/ex4.py
There are 100 cars available.
There are only 30 drivers available.
There will be 70 empty cars today.
We can transport 120.0 people today.
We have 90 to carpool today.
We need to put about 3.0 in each car.
```

Exercise 5: print与变量



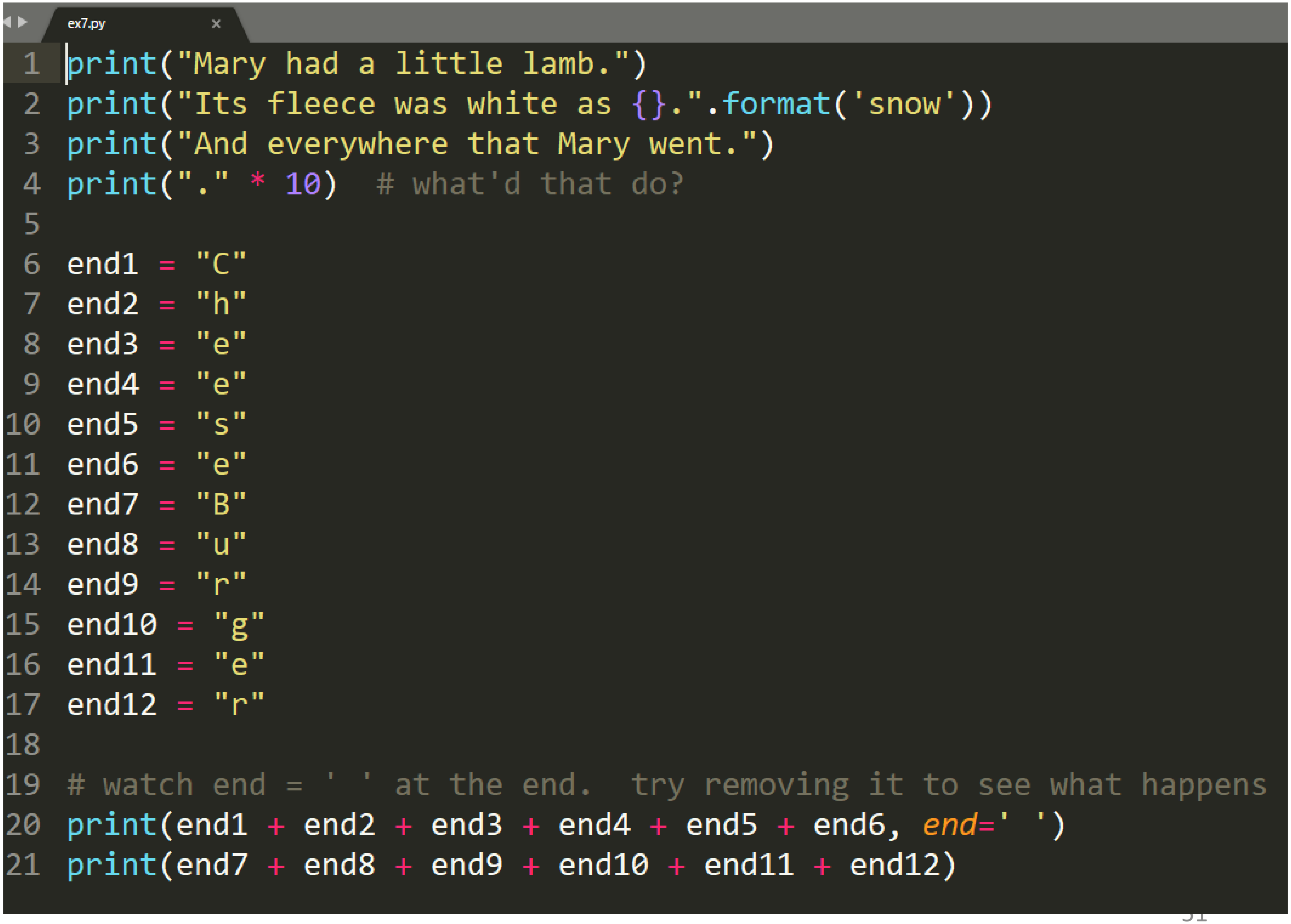
```
my_name = 'Zed A. Shaw'
my_age = 35 # not a lie
my_height = 74 # inches
my_weight = 180 # lbs
my_eyes = 'Blue'
my_teeth = 'White'
my_hair = 'Brown'

print(f"Let's talk about {my_name}.")
print(f"He's {my_height} inches tall.")
print(f"He's {my_weight} pounds heavy.")
print("Actually that's not too heavy.")
print(f"He's got {my_eyes} eyes and {my_hair} hair.")
print(f"His teeth are usually {my_teeth} depending on the coffee.")

# this line is tricky, try to get it exactly right
total = my_age + my_height + my_weight
print(f"If I add {my_age}, {my_height}, and {my_weight} I get {total}.")
```

Exercise 6: 字符串格式化2



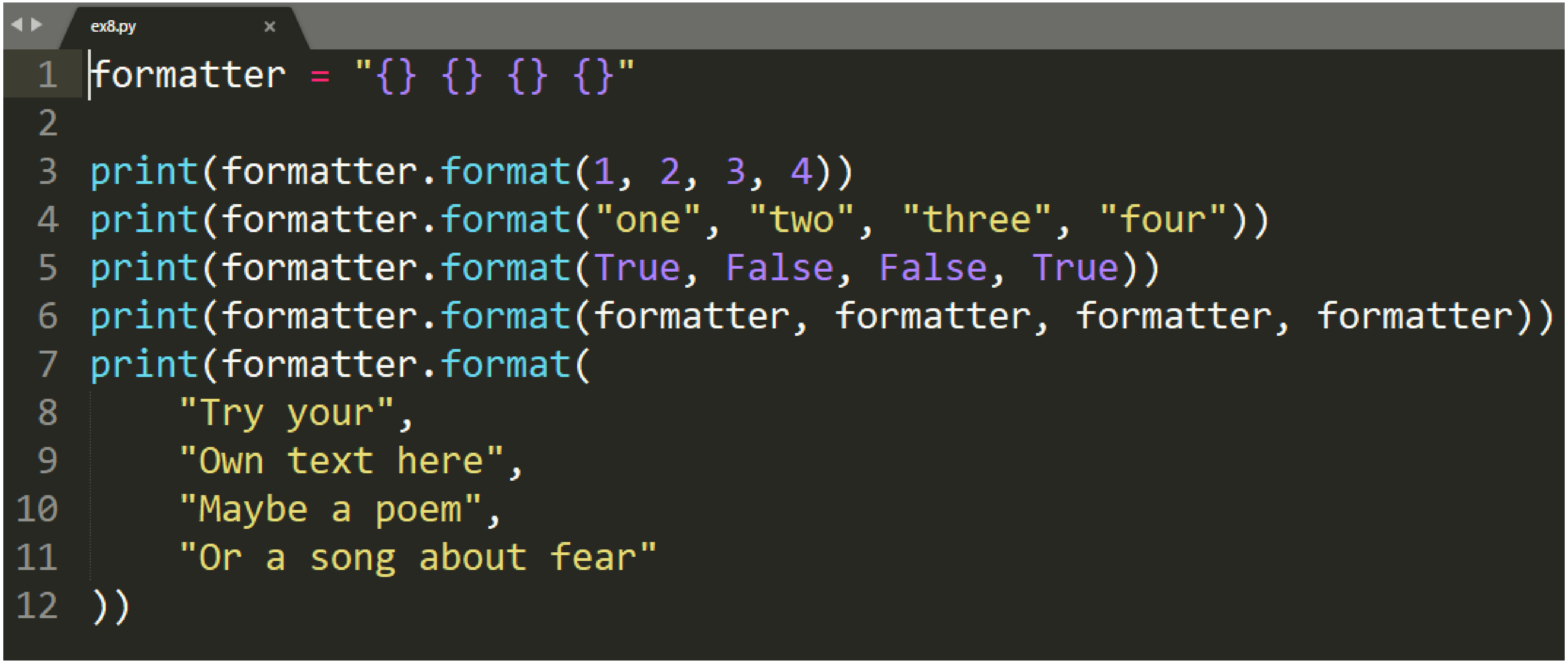
```python
types_of_people = 10
x = f"There are {types_of_people} types of people."

binary = "binary"
do_not = "don't"
y = f"Those who know {binary} and those who {do_not}."

print(x)
print(y)

print(f"I said: {x}")
print(f"I also said: '{y}'")

hilarious = False
joke_evaluation = "Isn't that joke so funny?! {}"

print(joke_evaluation.format(hilarious))

w = "This is the left side of..."
e = "a string with a right side."

print(w + e)
```

Exercise 7: ex7.py



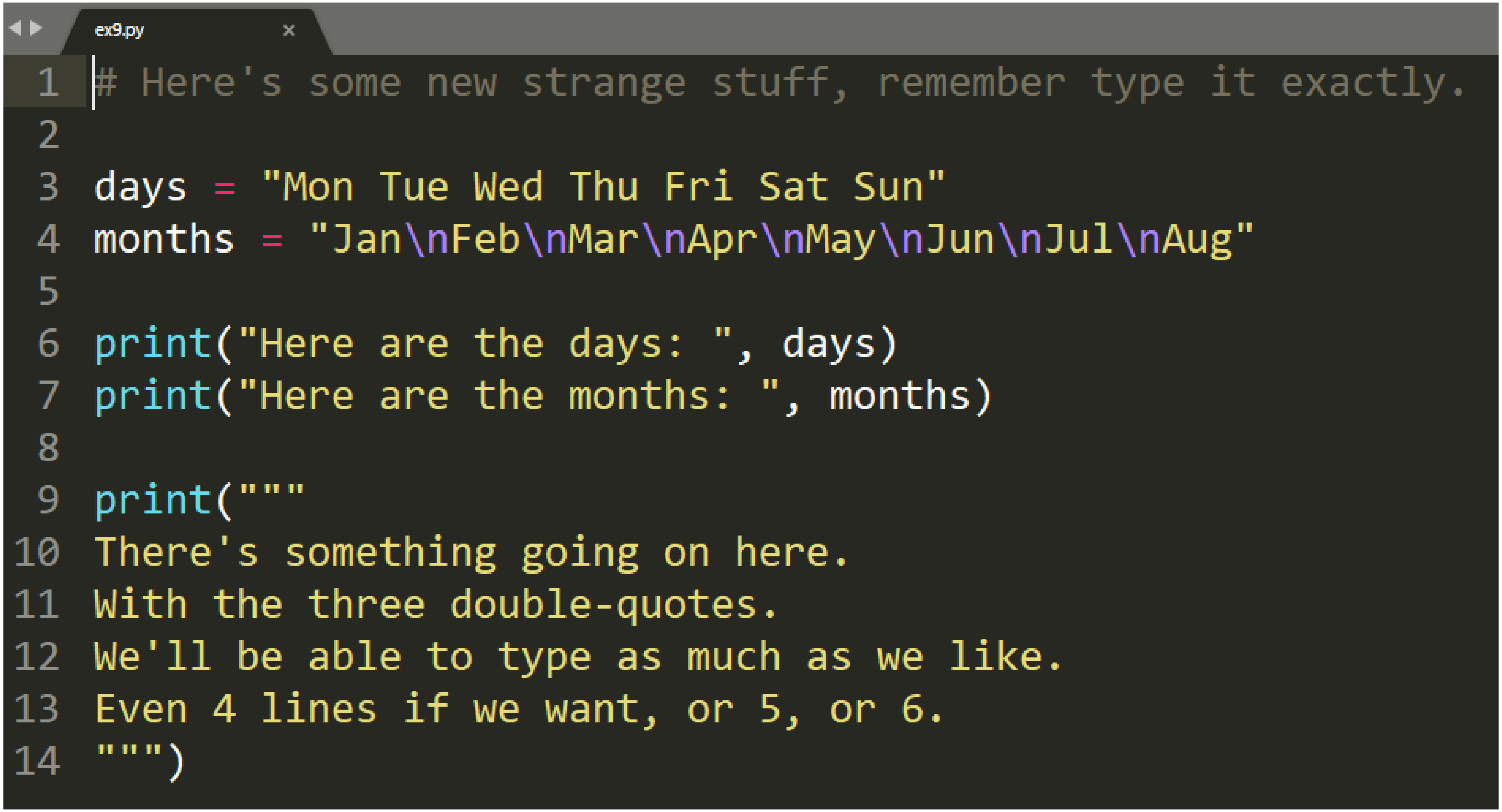
```python
print("Mary had a little lamb.")
print("Its fleece was white as {}.".format('snow'))
print("And everywhere that Mary went.")
print("." * 10)  # what'd that do?

end1 = "C"
end2 = "h"
end3 = "e"
end4 = "e"
end5 = "s"
end6 = "e"
end7 = "B"
end8 = "u"
end9 = "r"
end10 = "g"
end11 = "e"
end12 = "r"

# watch end = ' ' at the end.  try removing it to see what happens
print(end1 + end2 + end3 + end4 + end5 + end6, end=' ')
print(end7 + end8 + end9 + end10 + end11 + end12)
```

Exercise 8: ex8.py



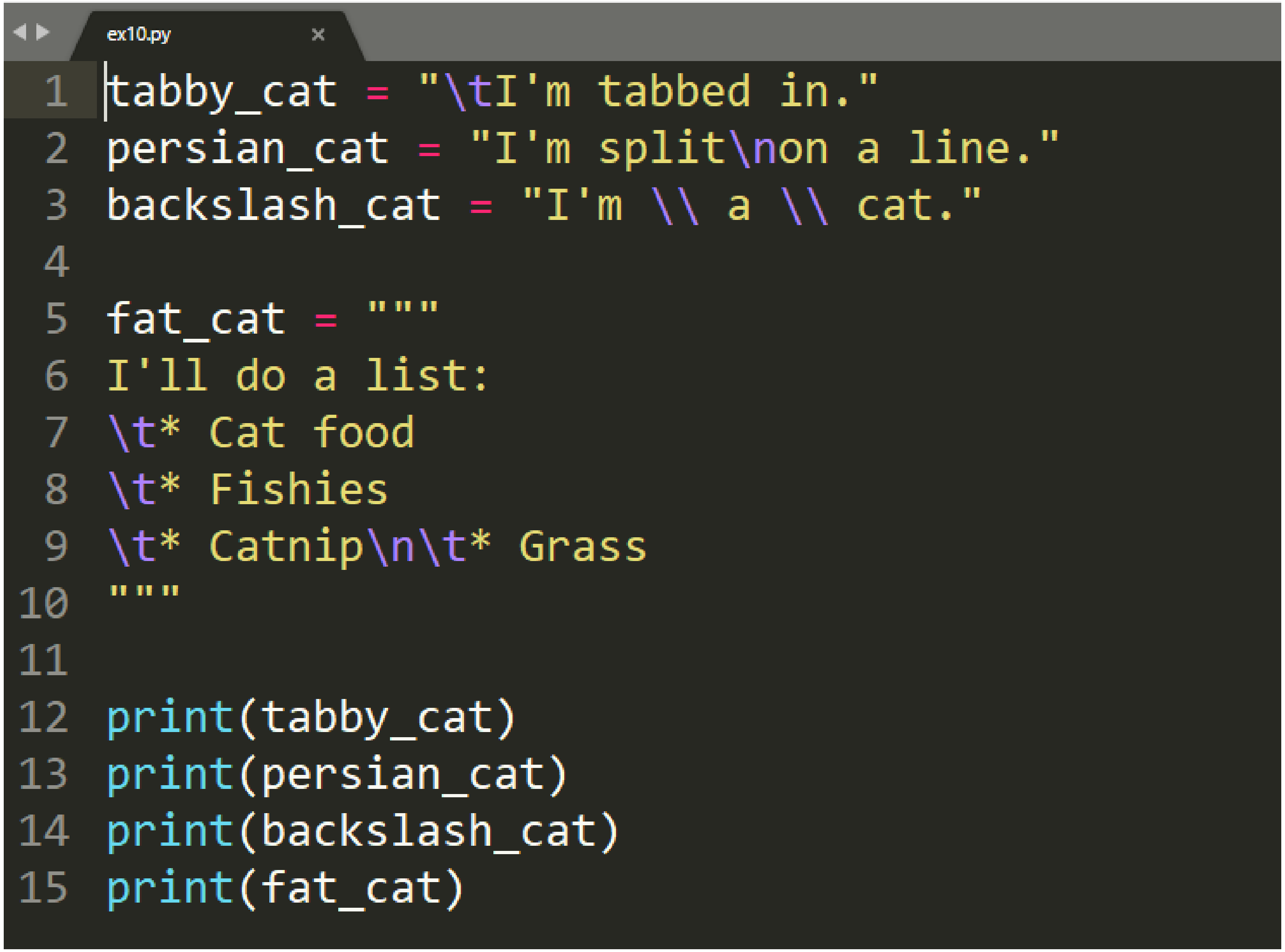
```python
formatter = "{} {} {} {}"

print(formatter.format(1, 2, 3, 4))
print(formatter.format("one", "two", "three", "four"))
print(formatter.format(True, False, False, True))
print(formatter.format(formatter, formatter, formatter, formatter))
print(formatter.format(
    "Try your",
    "Own text here",
    "Maybe a poem",
    "Or a song about fear"
))
```

Exercise 9: ex9.py

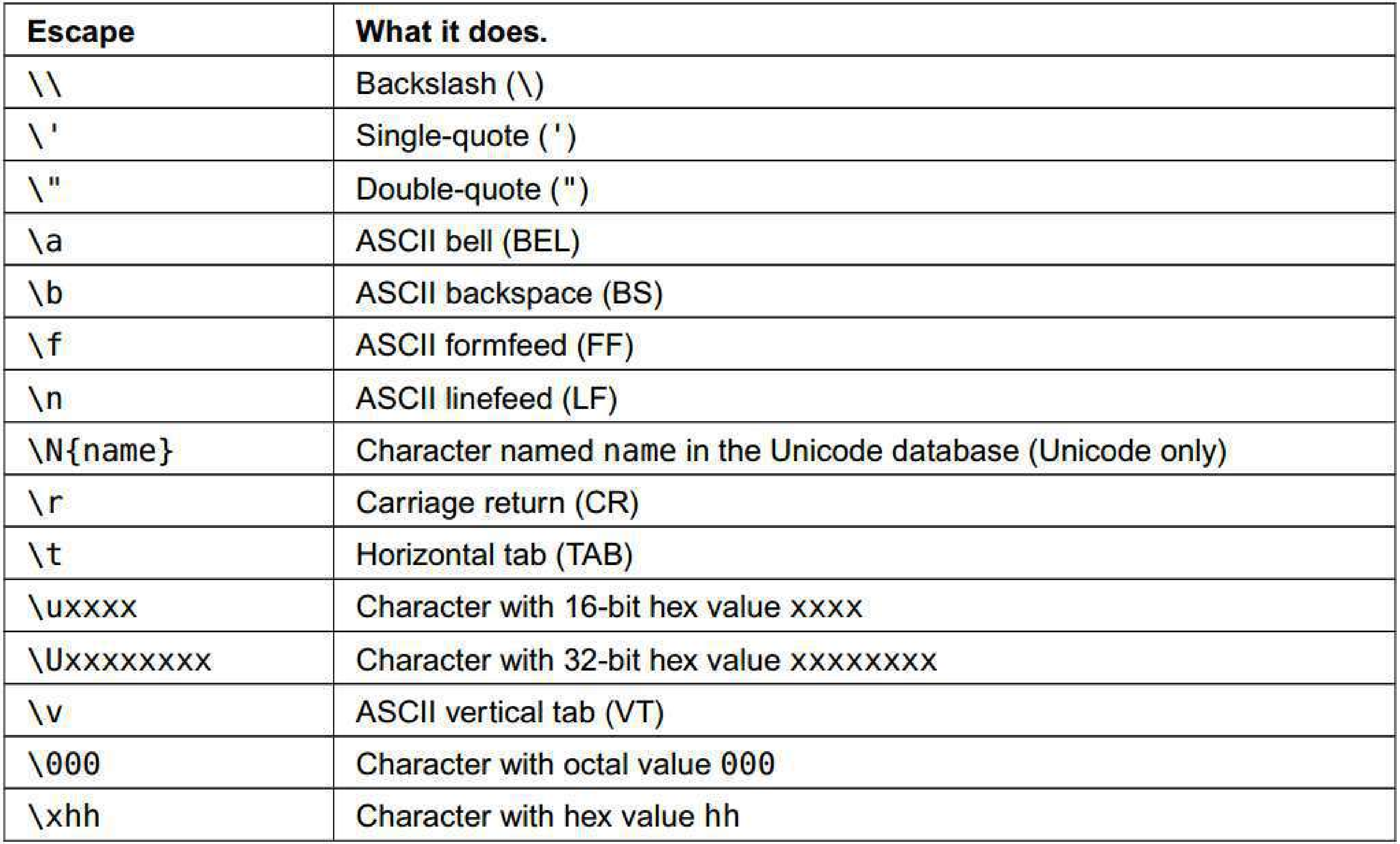


```python
# Here's some new strange stuff, remember type it exactly.

days = "Mon Tue Wed Thu Fri Sat Sun"
months = "Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"

print("Here are the days: ", days)
print("Here are the months: ", months)

print("""
There's something going on here.
With the three double-quotes.
We'll be able to type as much as we like.
Even 4 lines if we want, or 5, or 6.
""")
```

Exercise 10: ex10.py

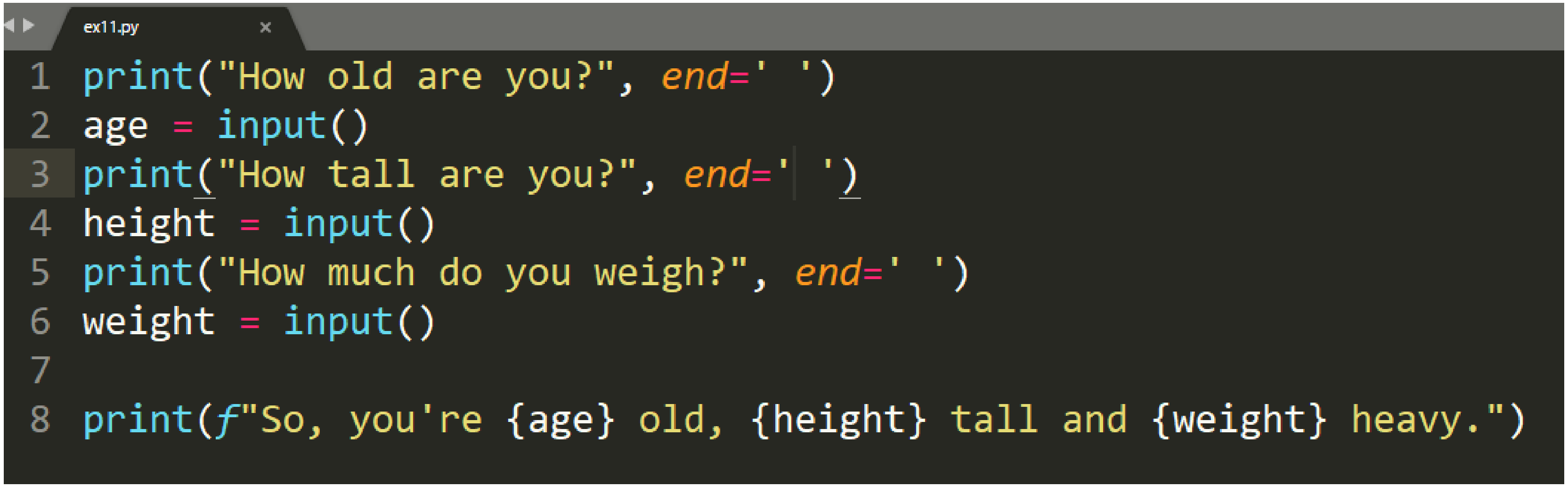


```python
tabby_cat = "\tI'm tabbed in."
persian_cat = "I'm split\non a line."
backslash_cat = "I'm \\ a \\ cat."

fat_cat = """
I'll do a list:
\t* Cat food
\t* Fishies
\t* Catnip\n\t* Grass
"""

print(tabby_cat)
print(persian_cat)
print(backslash_cat)
print(fat_cat)
```

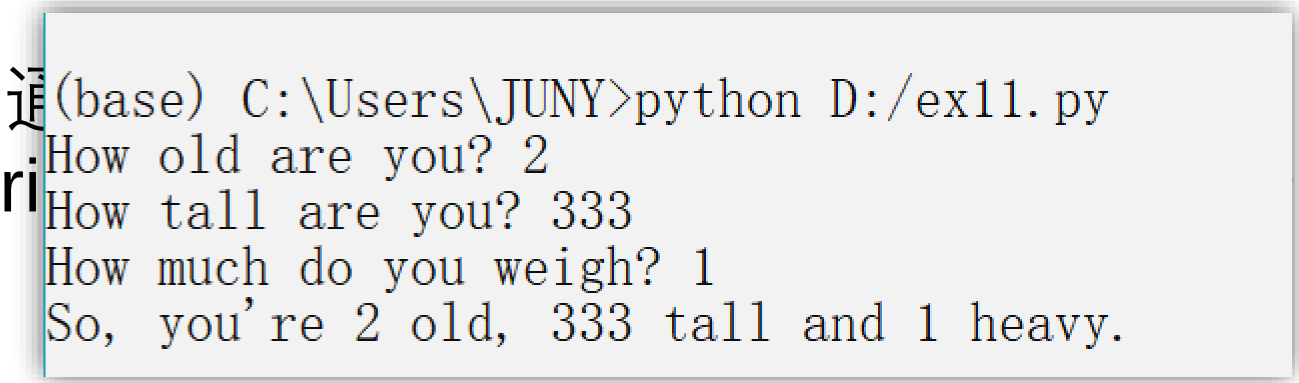


Escape	What it does.
`\\`	Backslash (\)
`\'`	Single-quote (')
`\"`	Double-quote (")
`\a`	ASCII bell (BEL)
`\b`	ASCII backspace (BS)
`\f`	ASCII formfeed (FF)
`\n`	ASCII linefeed (LF)
`\N{name}`	Character named `name` in the Unicode database (Unicode only)
`\r`	Carriage return (CR)
`\t`	Horizontal tab (TAB)
`\uxxxx`	Character with 16-bit hex value `xxxx`
`\Uxxxxxxxx`	Character with 32-bit hex value `xxxxxxxx`
`\v`	ASCII vertical tab (VT)
`\000`	Character with octal value `000`
`\xhh`	Character with hex value `hh`

EXERCISE 11: Asking Questions



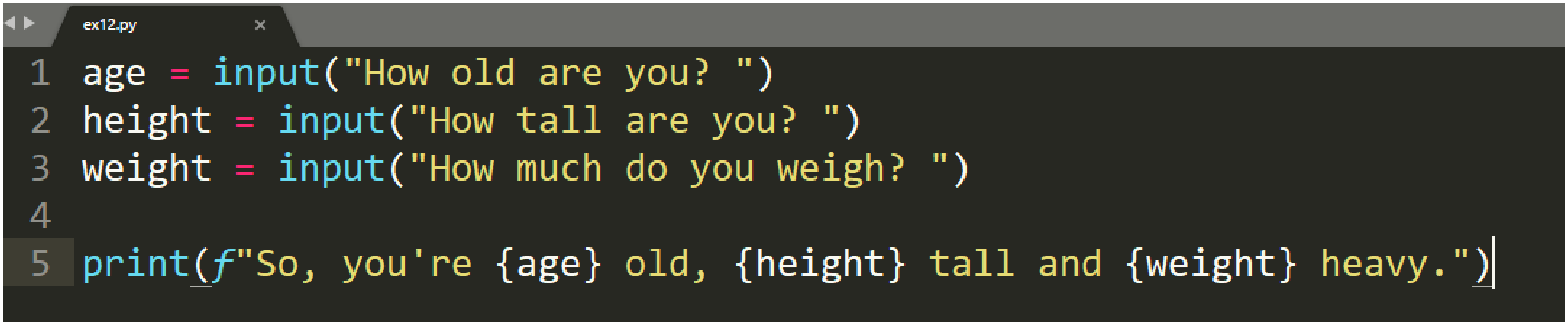
```python
print("How old are you?", end=' ')
age = input()
print("How tall are you?", end=' ')
height = input()
print("How much do you weigh?", end=' ')
weight = input()

print(f"So, you're {age} old, {height} tall and {weight} heavy.")
```



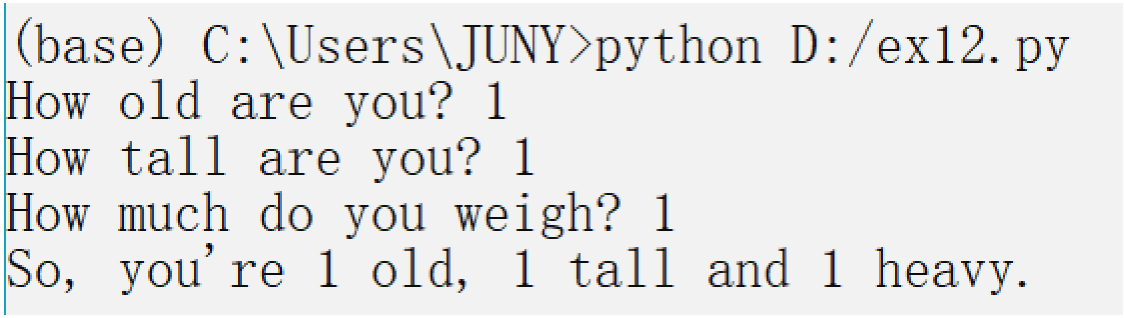
```
(base) C:\Users\JUNY>python D:/ex11.py
How old are you? 2
How tall are you? 333
How much do you weigh? 1
So, you’re 2 old, 333 tall and 1 heavy.
```

注意! 通 ’可以
告诉pri 向下一
行。

EXERCISE 12:

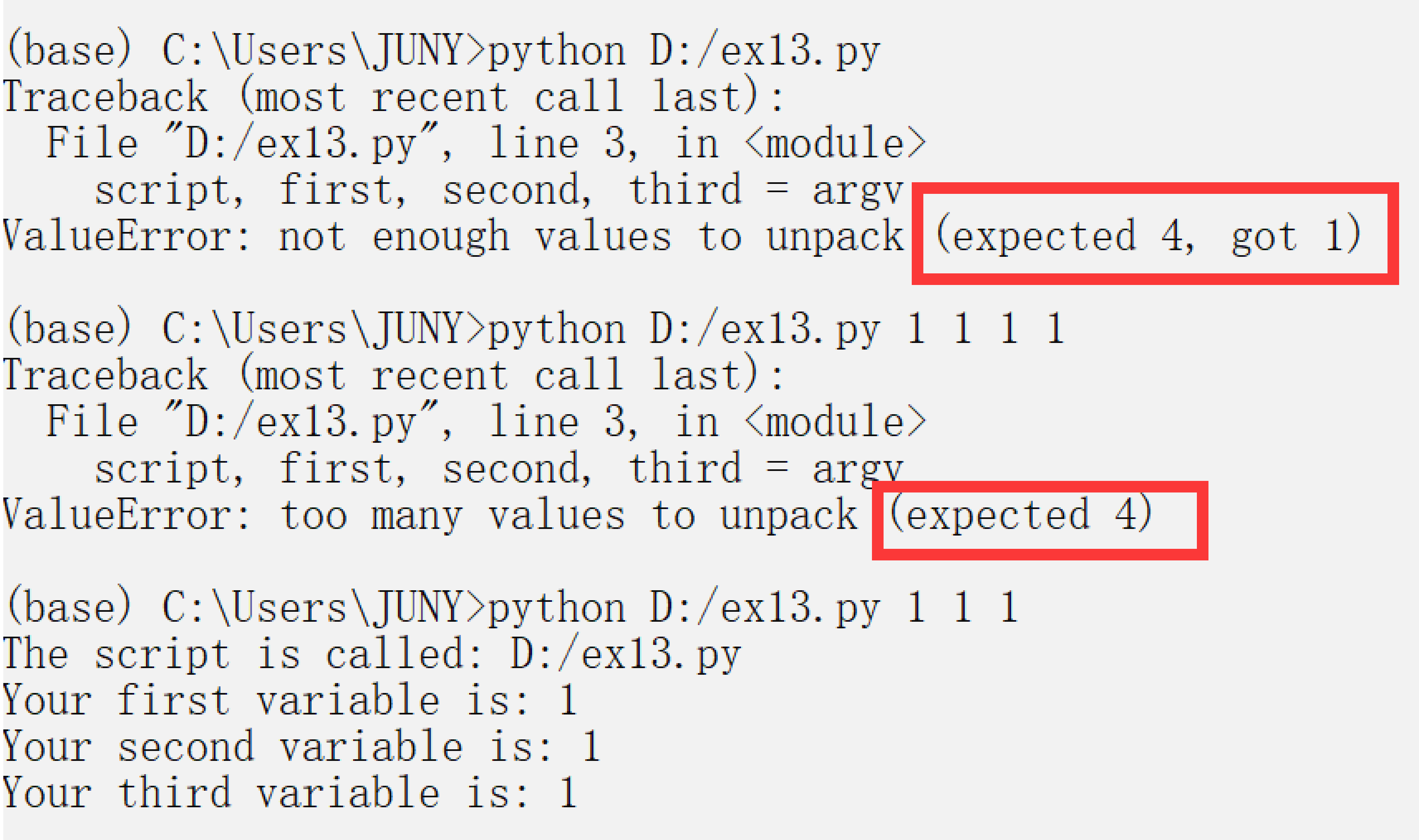


```python
age = input("How old are you? ")
height = input("How tall are you? ")
weight = input("How much do you weigh? ")

print(f"So, you're {age} old, {height} tall and {weight} heavy.")
```



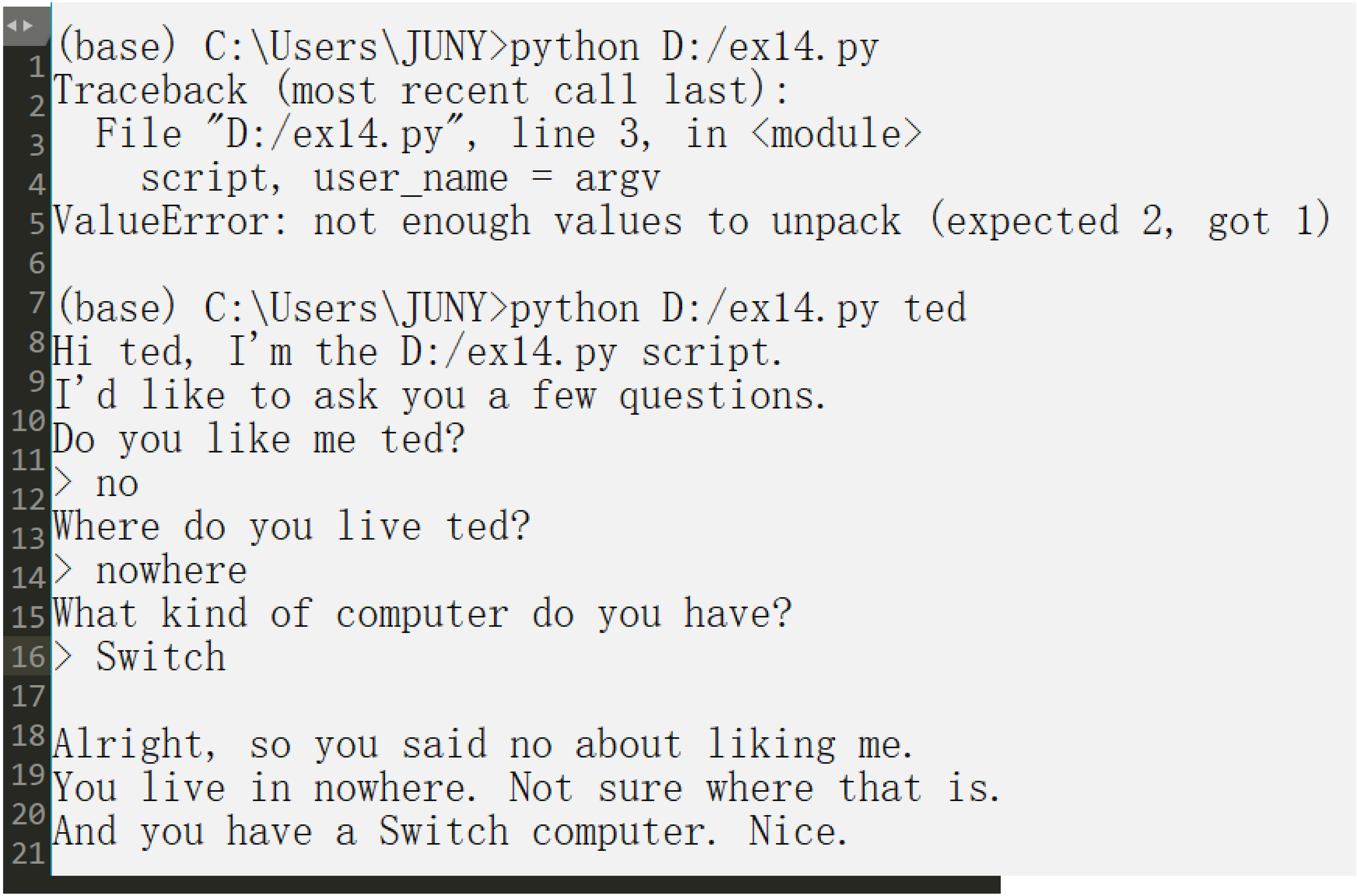
```
(base) C:\Users\JUNY>python D:/ex12.py
How old are you? 1
How tall are you? 1
How much do you weigh? 1
So, you’re 1 old, 1 tall and 1 heavy.
```

EXERCISE 13: Parameters, Unpacking, Variables



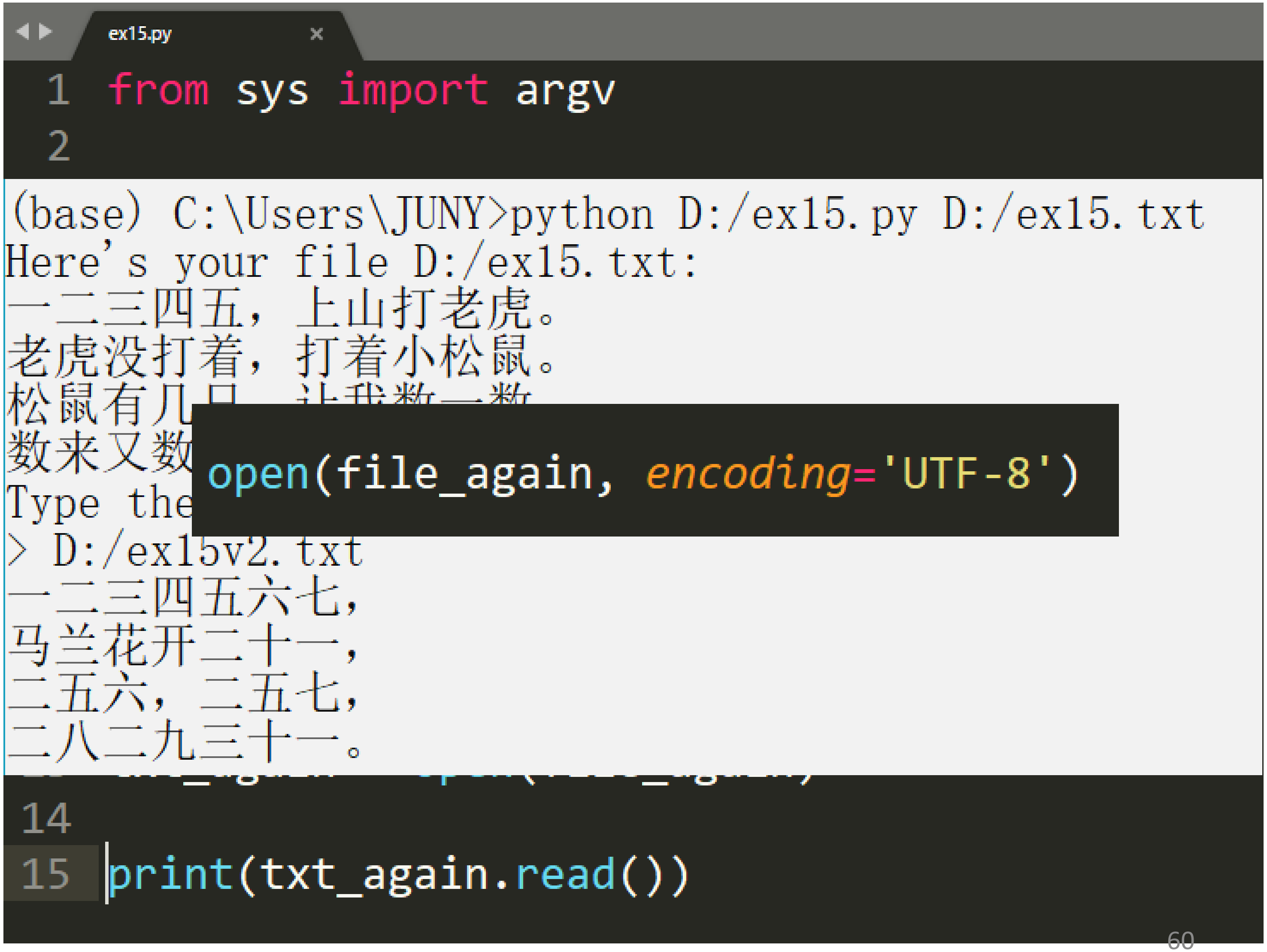
```
(base) C:\Users\JUNY>python D:/ex13.py
Traceback (most recent call last):
  File "D:/ex13.py", line 3, in <module>
    script, first, second, third = argv
ValueError: not enough values to unpack (expected 4, got 1)

(base) C:\Users\JUNY>python D:/ex13.py 1 1 1 1
Traceback (most recent call last):
  File "D:/ex13.py", line 3, in <module>
    script, first, second, third = argv
ValueError: too many values to unpack (expected 4)

(base) C:\Users\JUNY>python D:/ex13.py 1 1 1
The script is called: D:/ex13.py
Your first variable is: 1
Your second variable is: 1
Your third variable is: 1
```

EXERCISE 14: 提示与解析



```
(base) C:\Users\JUNY>python D:/ex14.py
Traceback (most recent call last):
  File "D:/ex14.py", line 3, in <module>
    script, user_name = argv
ValueError: not enough values to unpack (expected 2, got 1)

(base) C:\Users\JUNY>python D:/ex14.py ted
Hi ted, I’m the D:/ex14.py script.
I’d like to ask you a few questions.
Do you like me ted?
> no
Where do you live ted?
> nowhere
What kind of computer do you have?
> Switch

Alright, so you said no about liking me.
You live in nowhere. Not sure where that is.
And you have a Switch computer. Nice.
```

EX15：读取文件



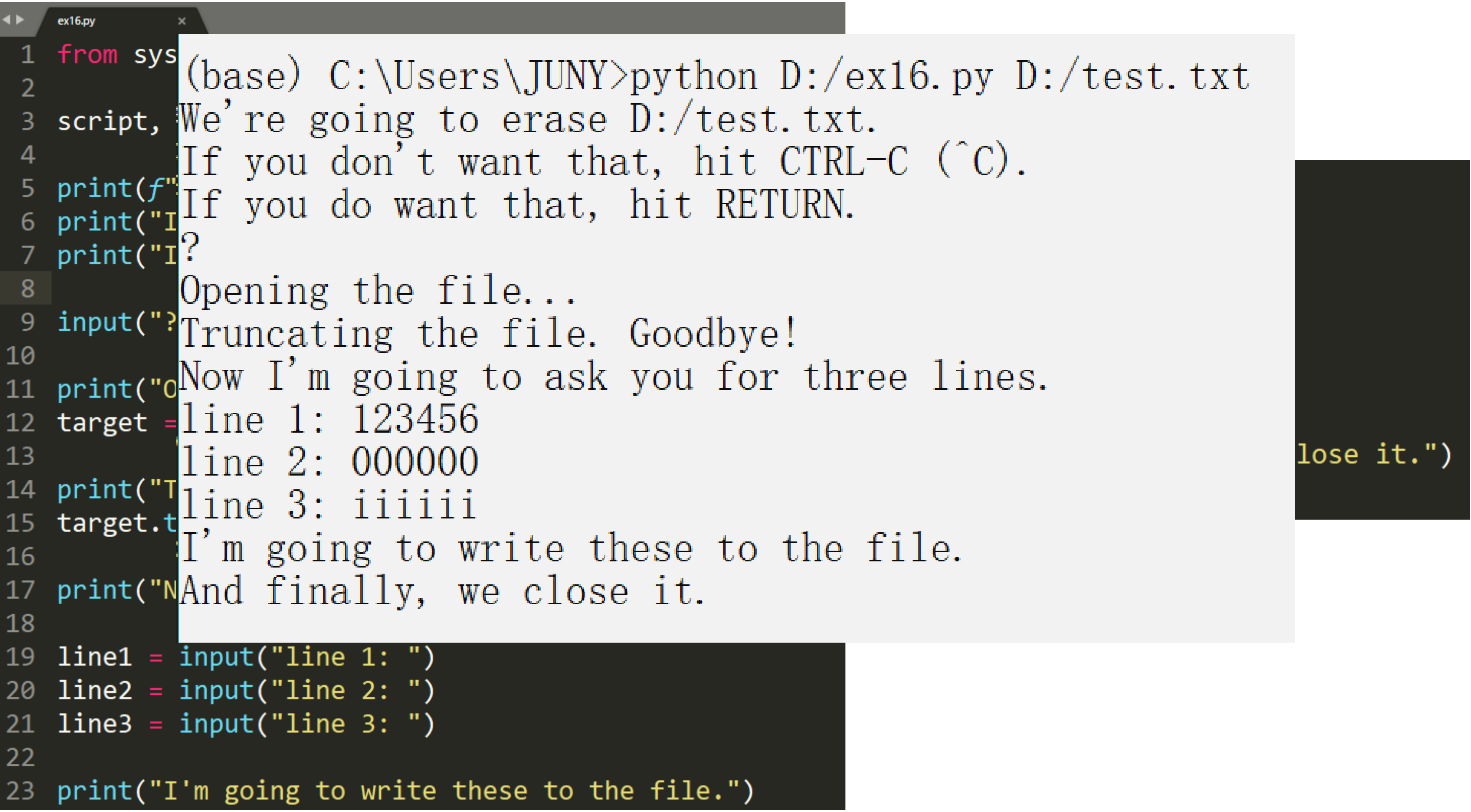
```
from sys import argv

```

```
(base) C:\Users\JUNY>python D:/ex15.py D:/ex15.txt
Here's your file D:/ex15.txt:
一二三四五，上山打老虎。
老虎没打着，打着小松鼠。
松鼠有几只，让我数一数
数来又数
Type the
> D:/ex15v2.txt
一二三四五六七，
马兰花开二十一，
二五六，二五七，
二八二九三十一。
```

```
open(file_again, encoding='UTF-8')
```

```

print(txt_again.read())
```

EX16：读写文件

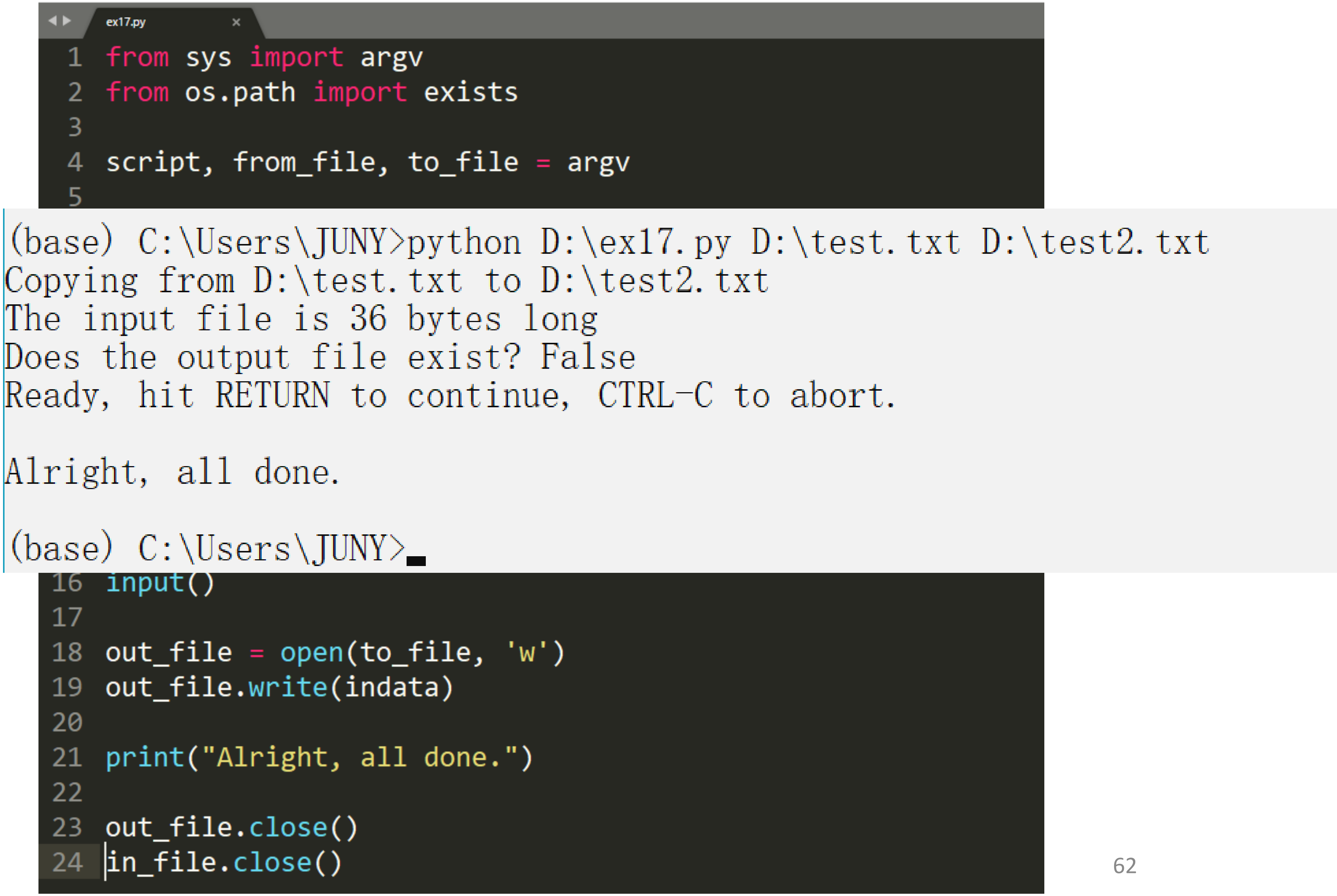


```
from sys

script,

print(f"
print("I
print("I

input("?

print("O
target =

print("T
target.t

print("N

line1 = input("line 1: ")
line2 = input("line 2: ")
line3 = input("line 3: ")

print("I'm going to write these to the file.")
```

```
lose it.")
```

```
(base) C:\Users\JUNY>python D:/ex16.py D:/test.txt
We're going to erase D:/test.txt.
If you don't want that, hit CTRL-C (^C).
If you do want that, hit RETURN.
?
Opening the file...
Truncating the file.  Goodbye!
Now I'm going to ask you for three lines.
line 1: 123456
line 2: 000000
line 3: iiiiii
I'm going to write these to the file.
And finally, we close it.
```

EX17: 更多文件操作

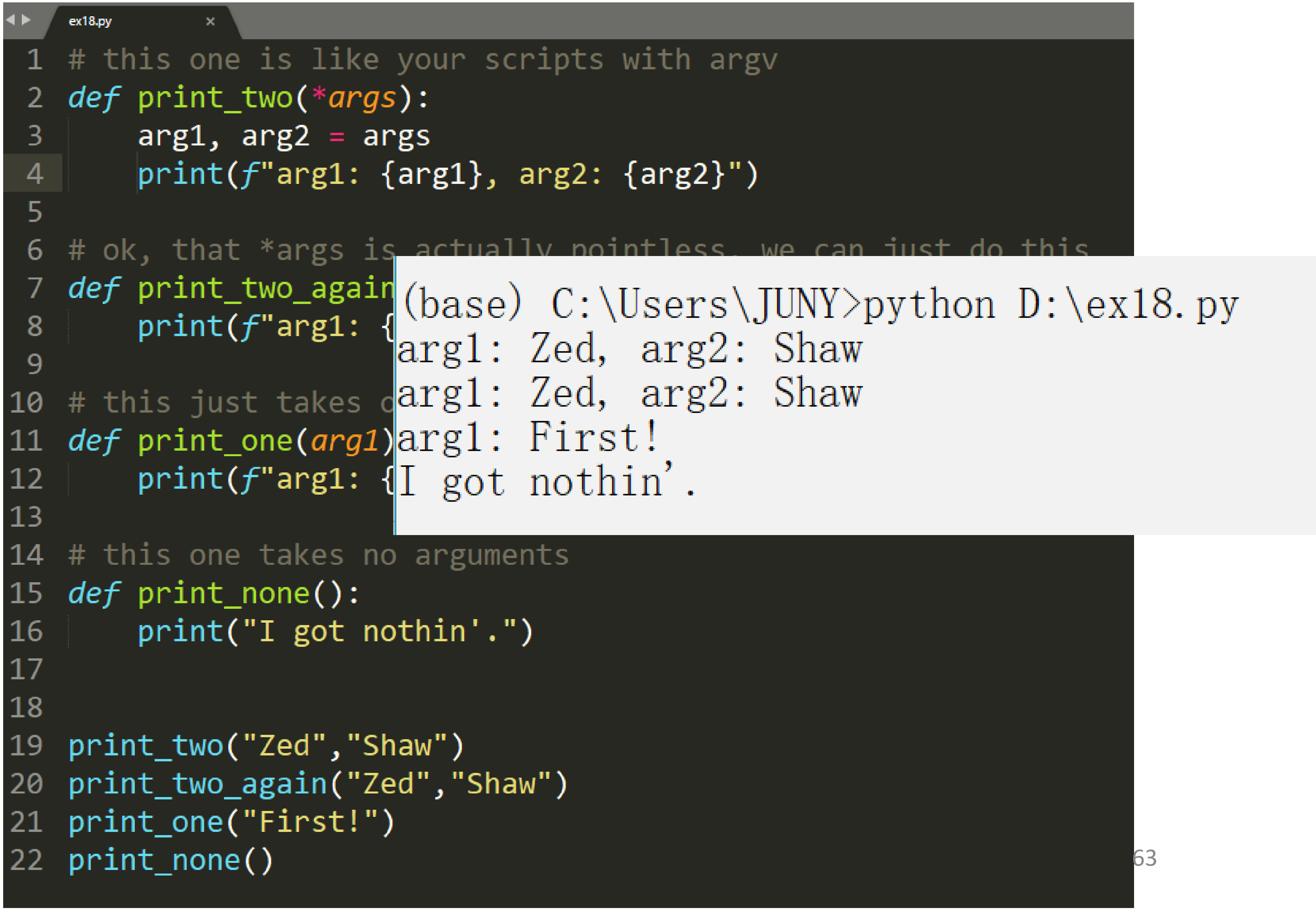


```python
from sys import argv
from os.path import exists

script, from_file, to_file = argv

```

```
(base) C:\Users\JUNY>python D:\ex17.py D:\test.txt D:\test2.txt
Copying from D:\test.txt to D:\test2.txt
The input file is 36 bytes long
Does the output file exist? False
Ready, hit RETURN to continue, CTRL-C to abort.

Alright, all done.

(base) C:\Users\JUNY>
```

```python
input()

out_file = open(to_file, 'w')
out_file.write(indata)

print("Alright, all done.")

out_file.close()
in_file.close()
```

EX18: 函数

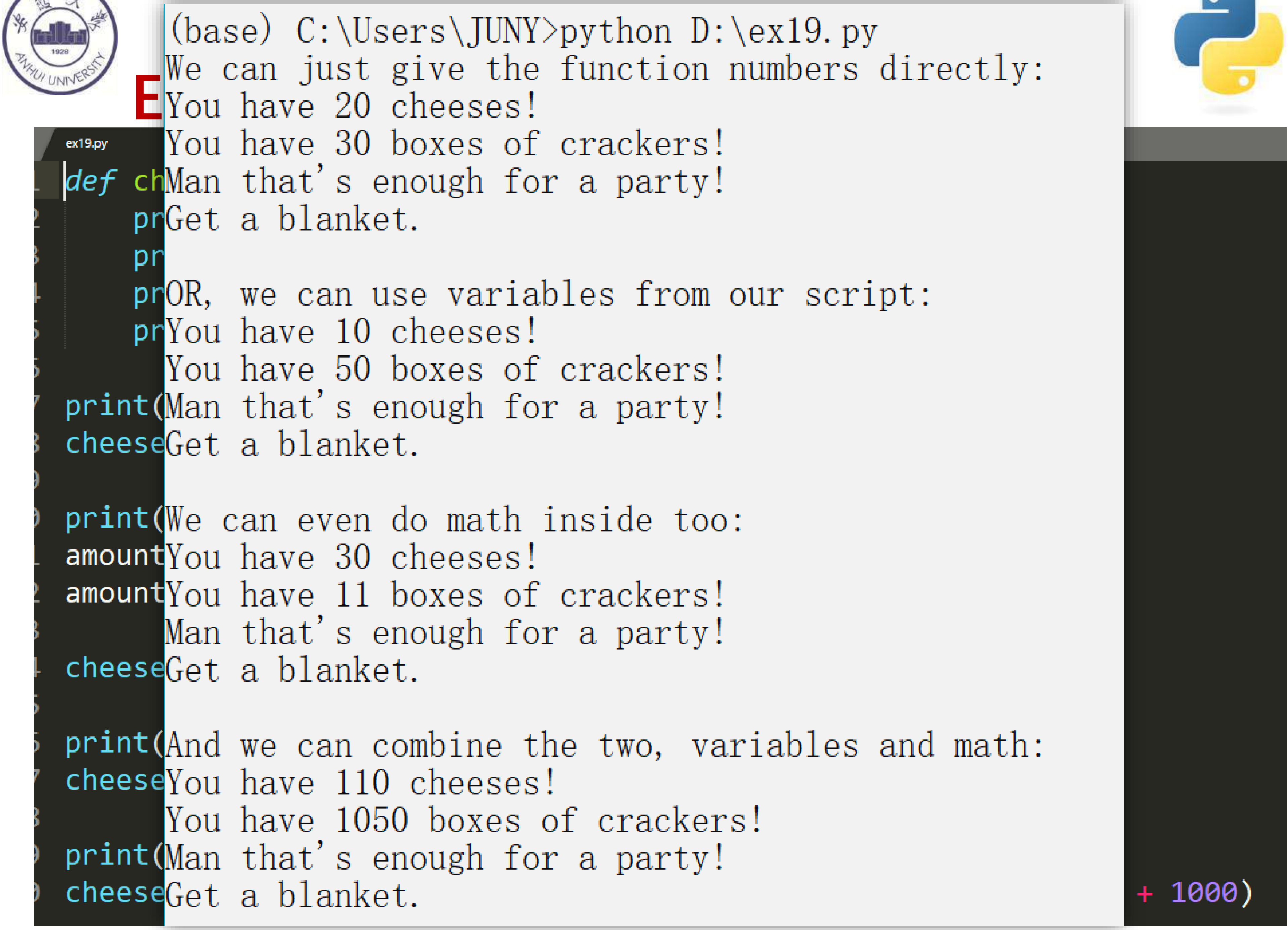


ex18.py

```python
# this one is like your scripts with argv
def print_two(*args):
    arg1, arg2 = args
    print(f"arg1: {arg1}, arg2: {arg2}")

# ok, that *args is actually pointless, we can just do this
def print_two_again
    print(f"arg1: {

# this just takes o
def print_one(arg1)
    print(f"arg1: {

# this one takes no arguments
def print_none():
    print("I got nothin'.")


print_two("Zed","Shaw")
print_two_again("Zed","Shaw")
print_one("First!")
print_none()
```

```
(base) C:\Users\JUNY>python D:\ex18.py
arg1: Zed, arg2: Shaw
arg1: Zed, arg2: Shaw
arg1: First!
I got nothin’.
```

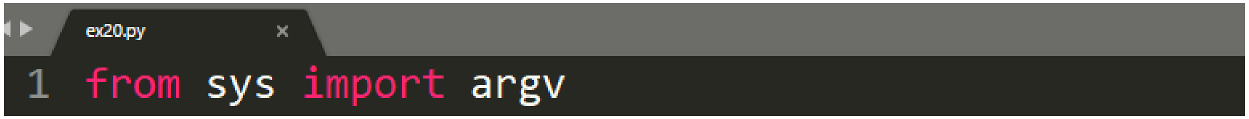
E



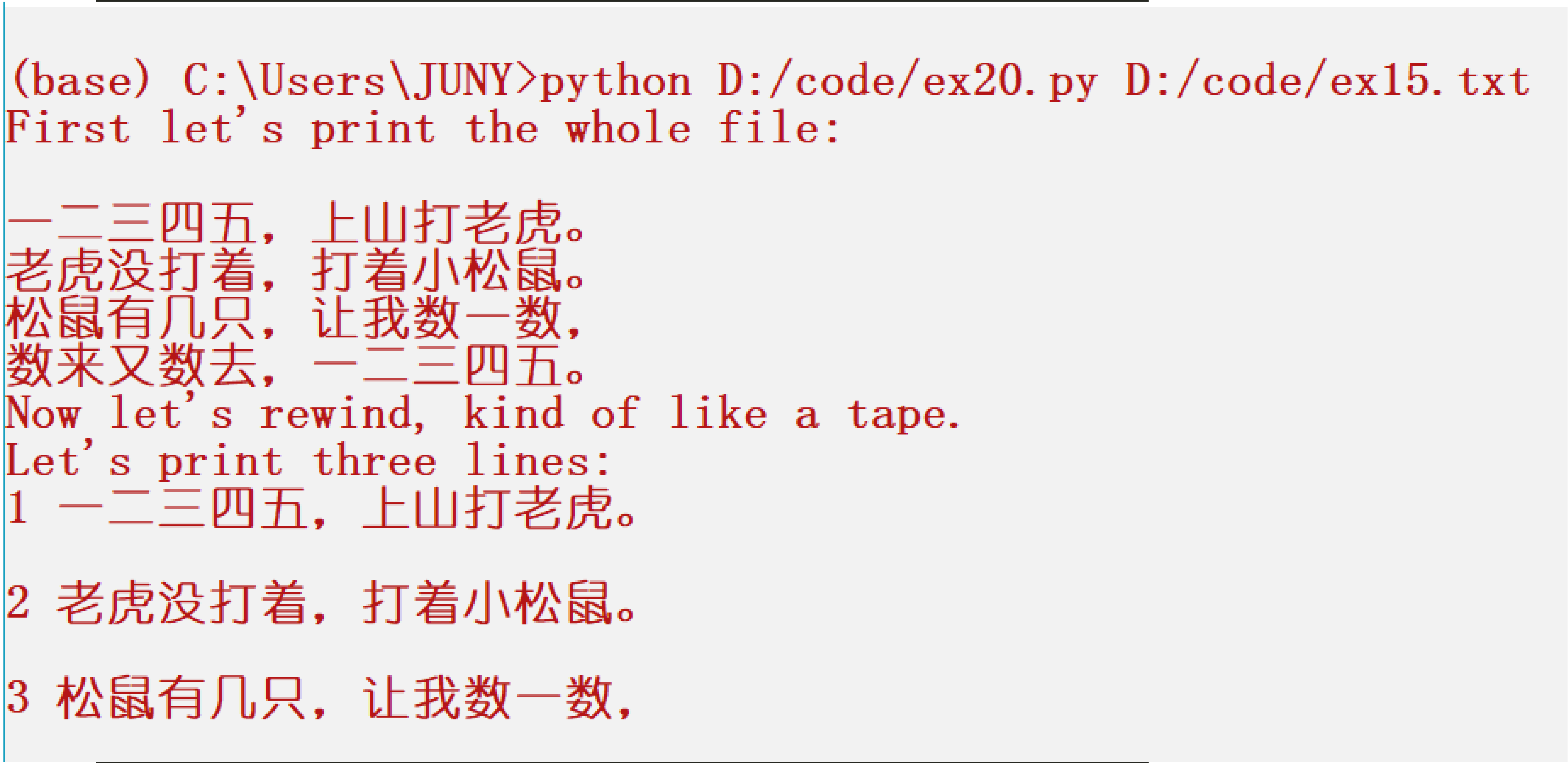
```
ex19.py
def ch
    pr
    pr
    pr
    pr

print(
cheese

print(
amount
amount

cheese

print(
cheese

print(
cheese                                                  + 1000)
```

```
(base) C:\Users\JUNY>python D:\ex19.py
We can just give the function numbers directly:
You have 20 cheeses!
You have 30 boxes of crackers!
Man that's enough for a party!
Get a blanket.

OR, we can use variables from our script:
You have 10 cheeses!
You have 50 boxes of crackers!
Man that's enough for a party!
Get a blanket.

We can even do math inside too:
You have 30 cheeses!
You have 11 boxes of crackers!
Man that's enough for a party!
Get a blanket.

And we can combine the two, variables and math:
You have 110 cheeses!
You have 1050 boxes of crackers!
Man that's enough for a party!
Get a blanket.
```

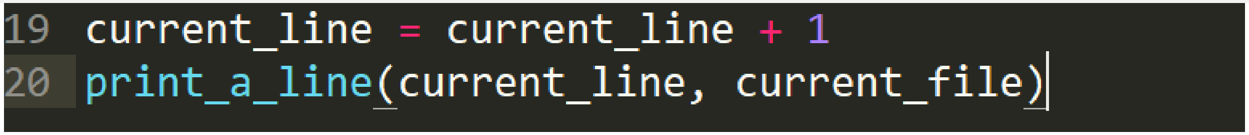
EX20：函数和文件



```python
from sys import argv
```

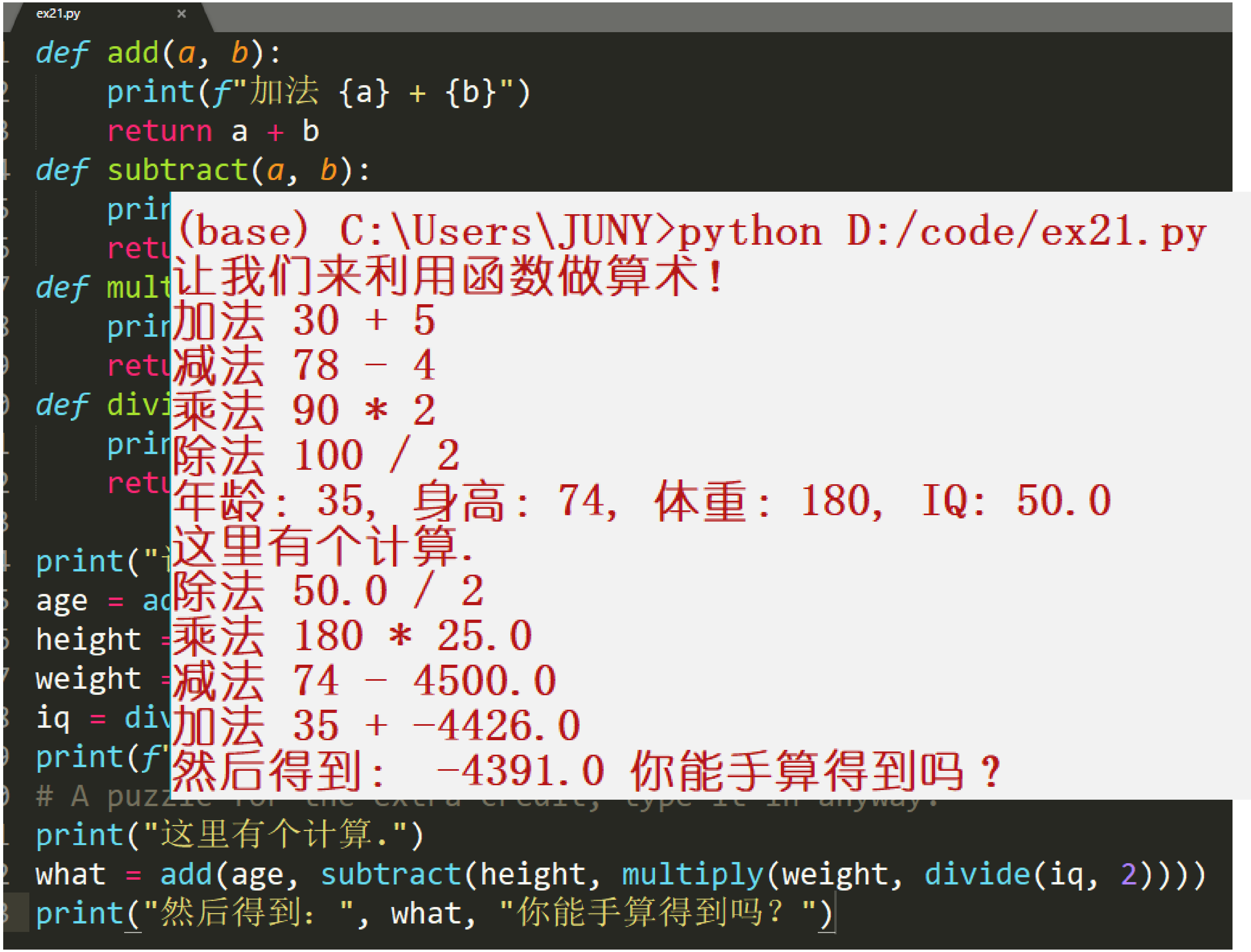


```
(base) C:\Users\JUNY>python D:/code/ex20.py D:/code/ex15.txt
First let’s print the whole file:

一二三四五，上山打老虎。
老虎没打着，打着小松鼠。
松鼠有几只，让我数一数，
数来又数去，一二三四五。
Now let’s rewind, kind of like a tape.
Let’s print three lines:
1 一二三四五，上山打老虎。

2 老虎没打着，打着小松鼠。

3 松鼠有几只，让我数一数，
```



```python
current_line = current_line + 1
print_a_line(current_line, current_file)
```

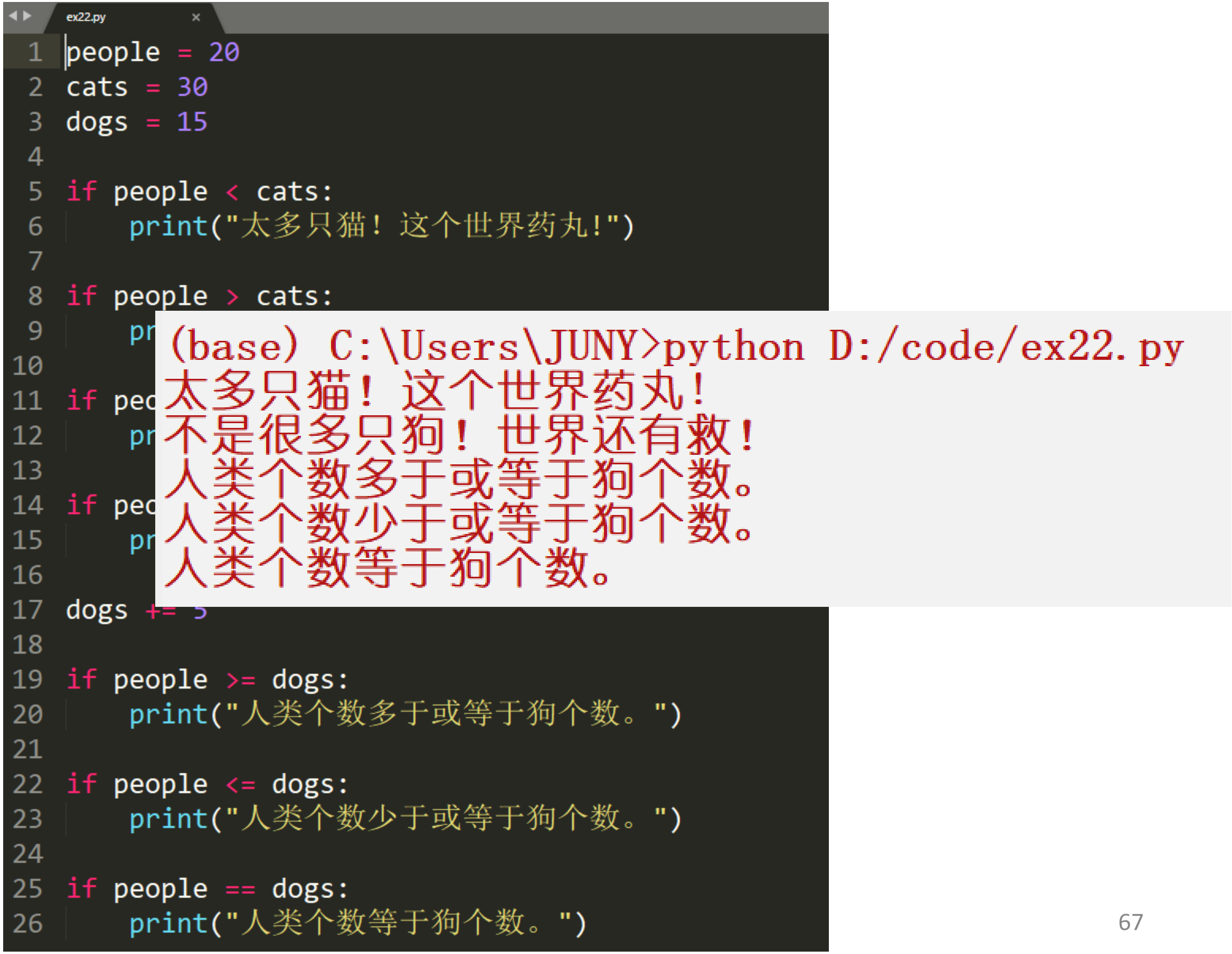
EX21: 函数返回值



```
def add(a, b):
    print(f"加法 {a} + {b}")
    return a + b
def subtract(a, b):
    prin
    retu
def mult
    prin
    retu
def divi
    prin
    retu

print("i
age = ad
height =
weight =
iq = div
print(f'
# A puzzle for the extra credit, type it in anyway.
print("这里有个计算.")
what = add(age, subtract(height, multiply(weight, divide(iq, 2))))
print("然后得到：", what, "你能手算得到吗？")
```

```
(base) C:\Users\JUNY>python D:/code/ex21.py
让我们来利用函数做算术!
加法 30 + 5
减法 78 - 4
乘法 90 * 2
除法 100 / 2
年龄： 35， 身高： 74， 体重： 180， IQ： 50.0
这里有个计算.
除法 50.0 / 2
乘法 180 * 25.0
减法 74 - 4500.0
加法 35 + -4426.0
然后得到：  -4391.0 你能手算得到吗？
```

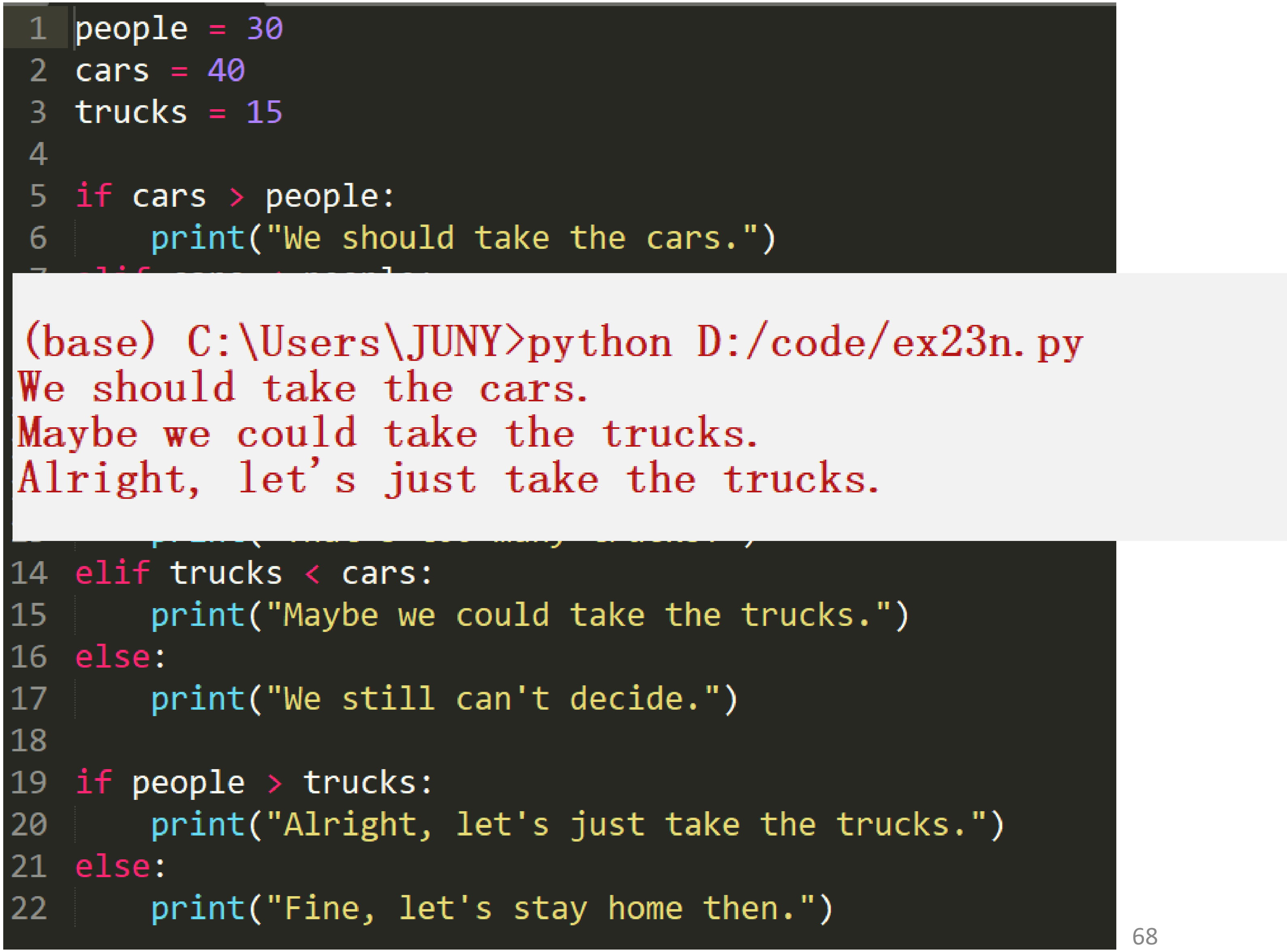
EX22: IF



```python
people = 20
cats = 30
dogs = 15

if people < cats:
    print("太多只猫！ 这个世界药丸！")

if people > cats:
    pr

if peo
    pr

if peo
    pr

dogs += 5

if people >= dogs:
    print("人类个数多于或等于狗个数。")

if people <= dogs:
    print("人类个数少于或等于狗个数。")

if people == dogs:
    print("人类个数等于狗个数。")
```

```
(base) C:\Users\JUNY>python D:/code/ex22.py
太多只猫！ 这个世界药丸！
不是很多只狗！ 世界还有救！
人类个数多于或等于狗个数。
人类个数少于或等于狗个数。
人类个数等于狗个数。
```

EX23: ELSE IF

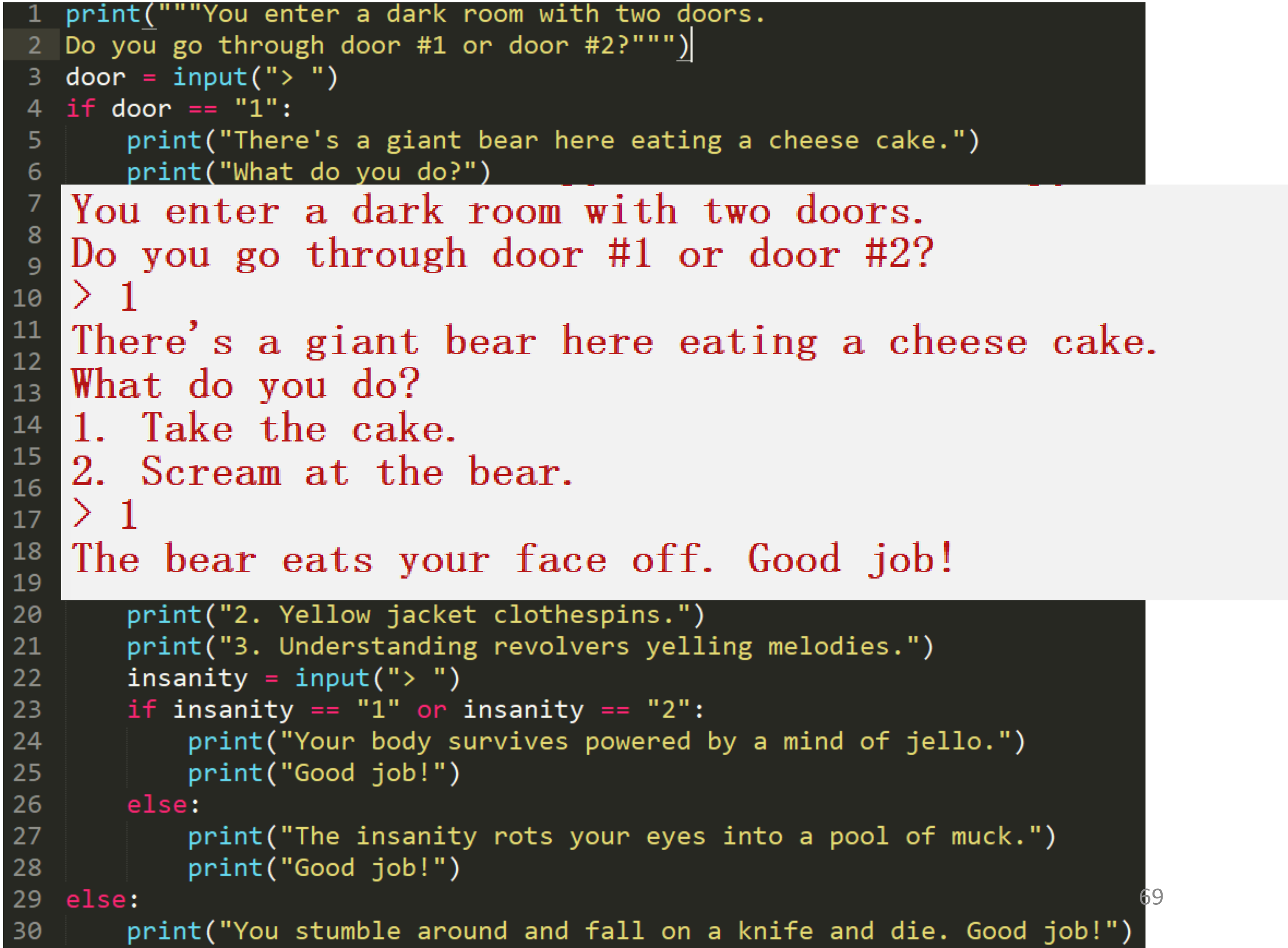


```python
people = 30
cars = 40
trucks = 15

if cars > people:
    print("We should take the cars.")
elif cars < people:
```

```
(base) C:\Users\JUNY>python D:/code/ex23n.py
We should take the cars.
Maybe we could take the trucks.
Alright, let’s just take the trucks.
```

```python
    print("That's too many trucks.")
elif trucks < cars:
    print("Maybe we could take the trucks.")
else:
    print("We still can't decide.")

if people > trucks:
    print("Alright, let's just take the trucks.")
else:
    print("Fine, let's stay home then.")
```

EX24: 做决定

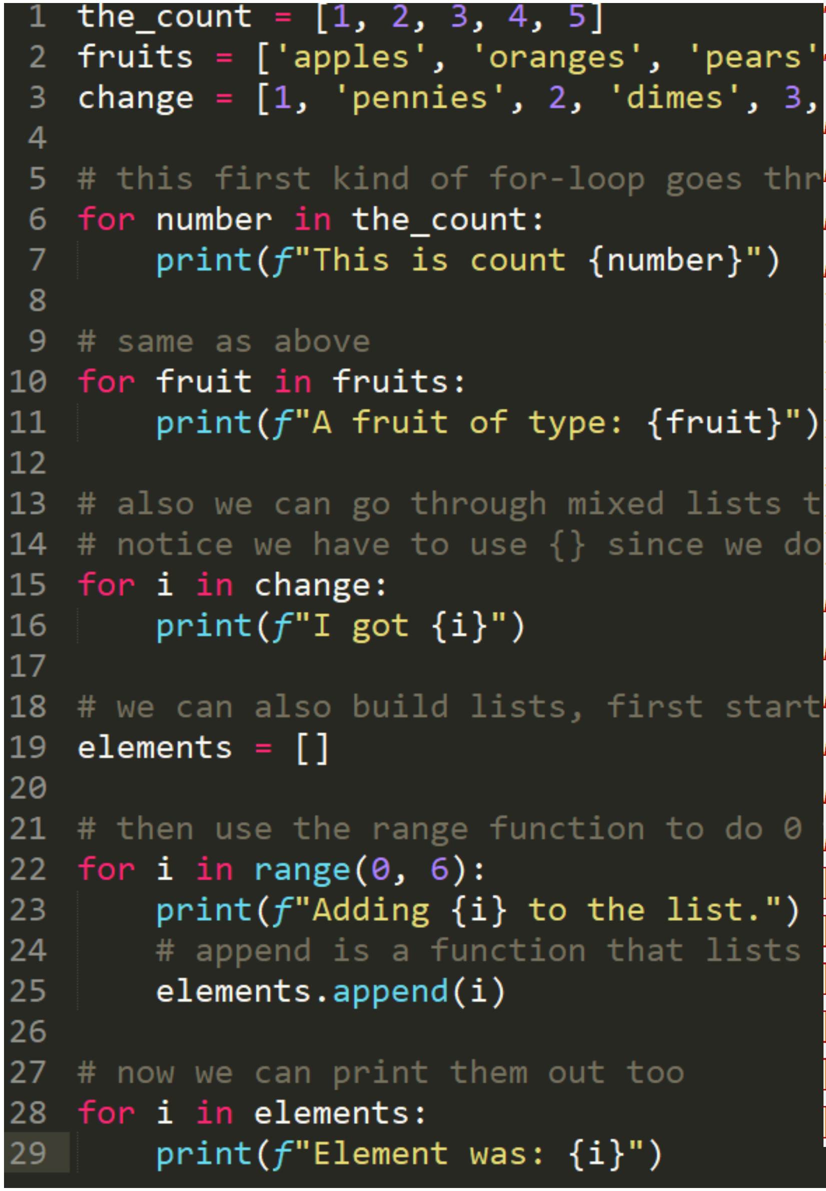


```python
print("""You enter a dark room with two doors.
Do you go through door #1 or door #2?""")
door = input("> ")
if door == "1":
    print("There's a giant bear here eating a cheese cake.")
    print("What do you do?")
```

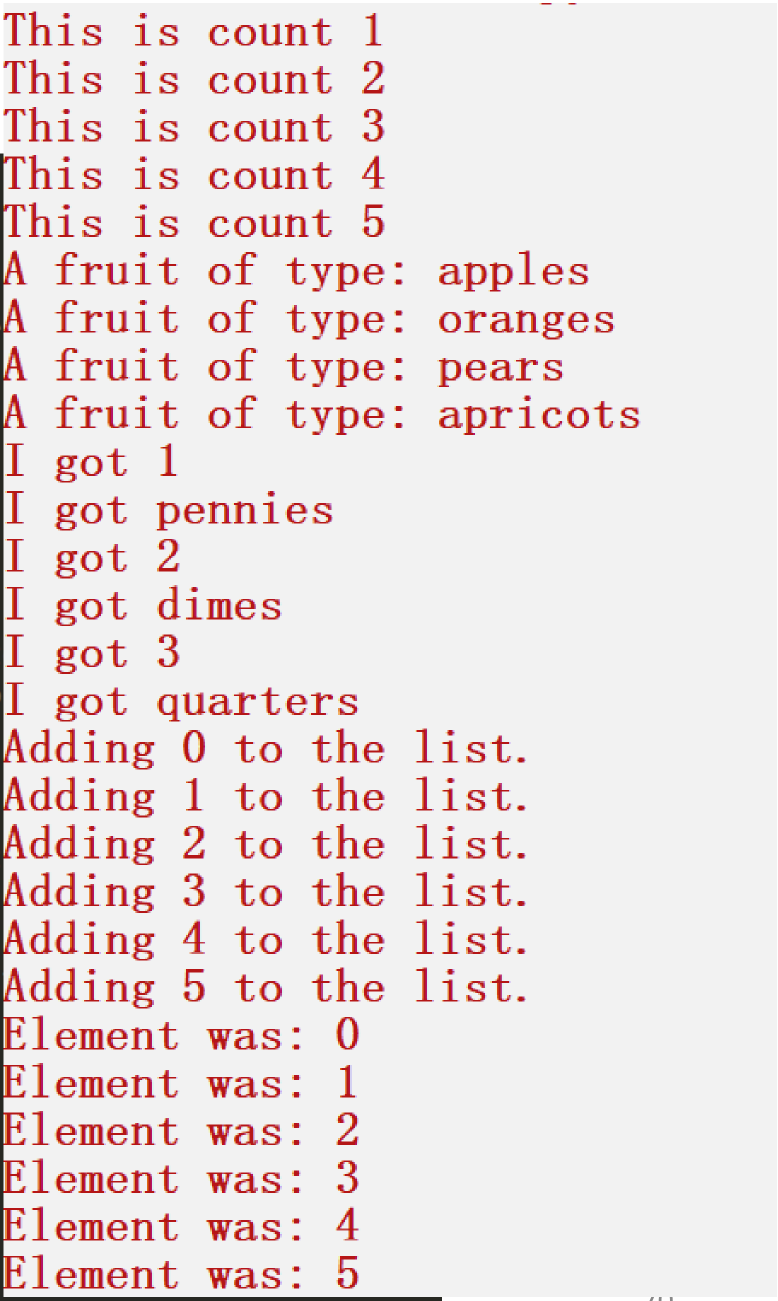
```
You enter a dark room with two doors.
Do you go through door #1 or door #2?
> 1
There’s a giant bear here eating a cheese cake.
What do you do?
1. Take the cake.
2. Scream at the bear.
> 1
The bear eats your face off. Good job!
```

```python
    print("2. Yellow jacket clothespins.")
    print("3. Understanding revolvers yelling melodies.")
    insanity = input("> ")
    if insanity == "1" or insanity == "2":
        print("Your body survives powered by a mind of jello.")
        print("Good job!")
    else:
        print("The insanity rots your eyes into a pool of muck.")
        print("Good job!")
else:
    print("You stumble around and fall on a knife and die. Good job!")
```

EX25： 循环和列

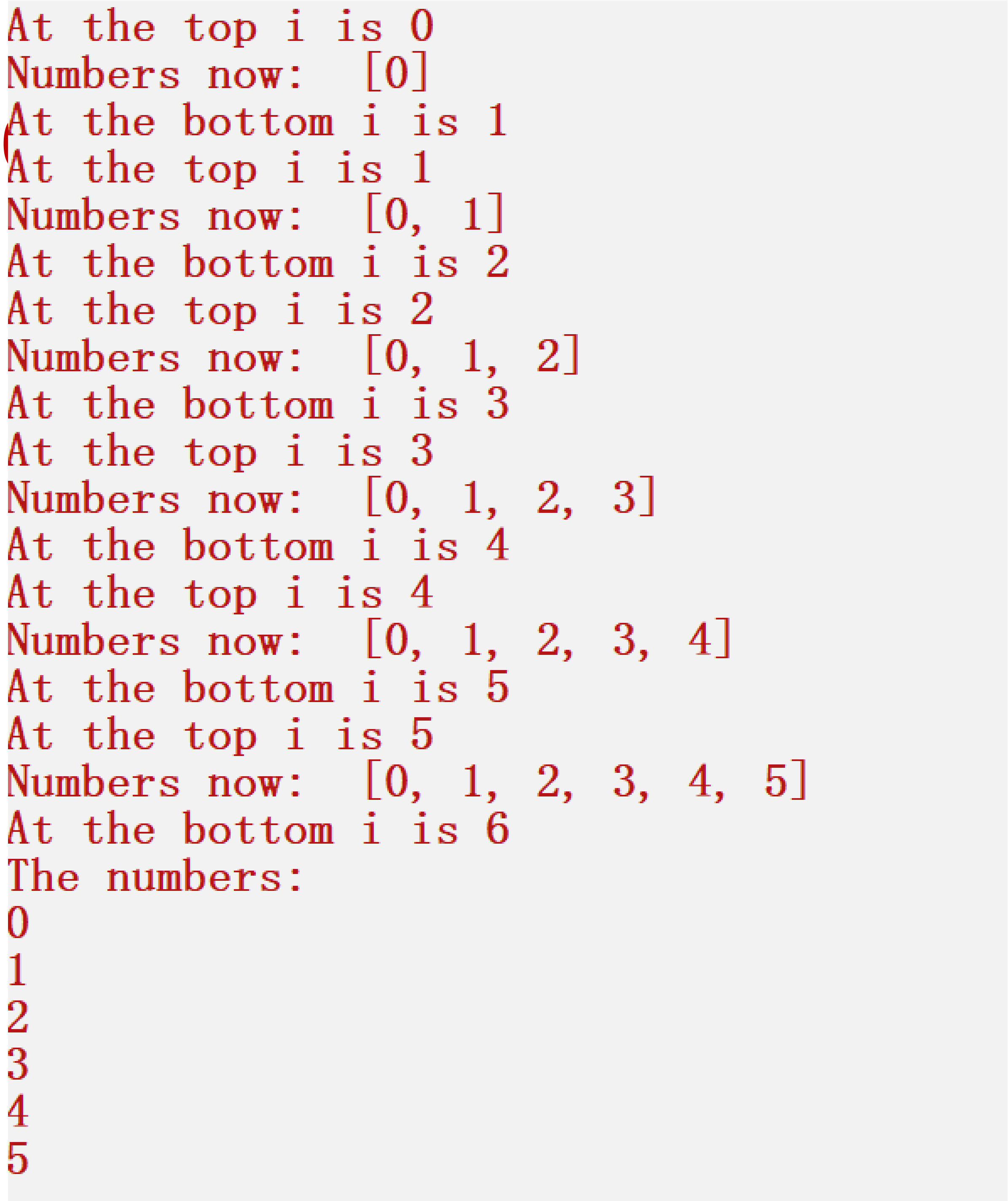


```python
the_count = [1, 2, 3, 4, 5]
fruits = ['apples', 'oranges', 'pears'
change = [1, 'pennies', 2, 'dimes', 3,

# this first kind of for-loop goes thr
for number in the_count:
    print(f"This is count {number}")

# same as above
for fruit in fruits:
    print(f"A fruit of type: {fruit}")

# also we can go through mixed lists t
# notice we have to use {} since we do
for i in change:
    print(f"I got {i}")

# we can also build lists, first start
elements = []

# then use the range function to do 0
for i in range(0, 6):
    print(f"Adding {i} to the list.")
    # append is a function that lists
    elements.append(i)

# now we can print them out too
for i in elements:
    print(f"Element was: {i}")
```



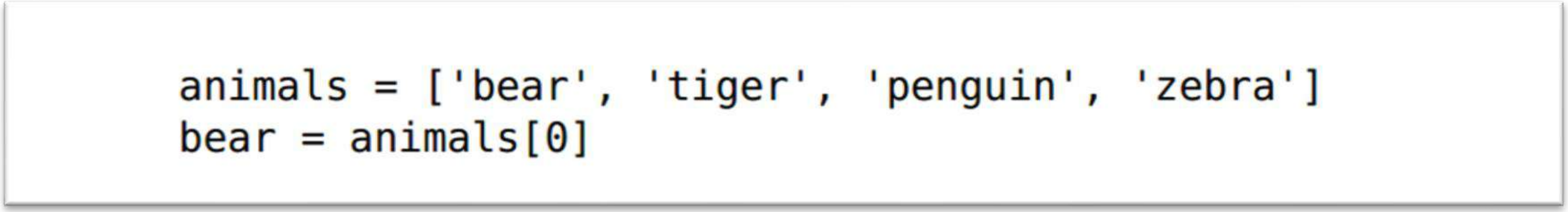
```
This is count 1
This is count 2
This is count 3
This is count 4
This is count 5
A fruit of type: apples
A fruit of type: oranges
A fruit of type: pears
A fruit of type: apricots
I got 1
I got pennies
I got 2
I got dimes
I got 3
I got quarters
Adding 0 to the list.
Adding 1 to the list.
Adding 2 to the list.
Adding 3 to the list.
Adding 4 to the list.
Adding 5 to the list.
Element was: 0
Element was: 1
Element was: 2
Element was: 3
Element was: 4
Element was: 5
```

EX2



```
At the top i is 0
Numbers now:  [0]
At the bottom i is 1
At the top i is 1
Numbers now:  [0, 1]
At the bottom i is 2
At the top i is 2
Numbers now:  [0, 1, 2]
At the bottom i is 3
At the top i is 3
Numbers now:  [0, 1, 2, 3]
At the bottom i is 4
At the top i is 4
Numbers now:  [0, 1, 2, 3, 4]
At the bottom i is 5
At the top i is 5
Numbers now:  [0, 1, 2, 3, 4, 5]
At the bottom i is 6
The numbers:
0
1
2
3
4
5
```

EX27：访问列表元素

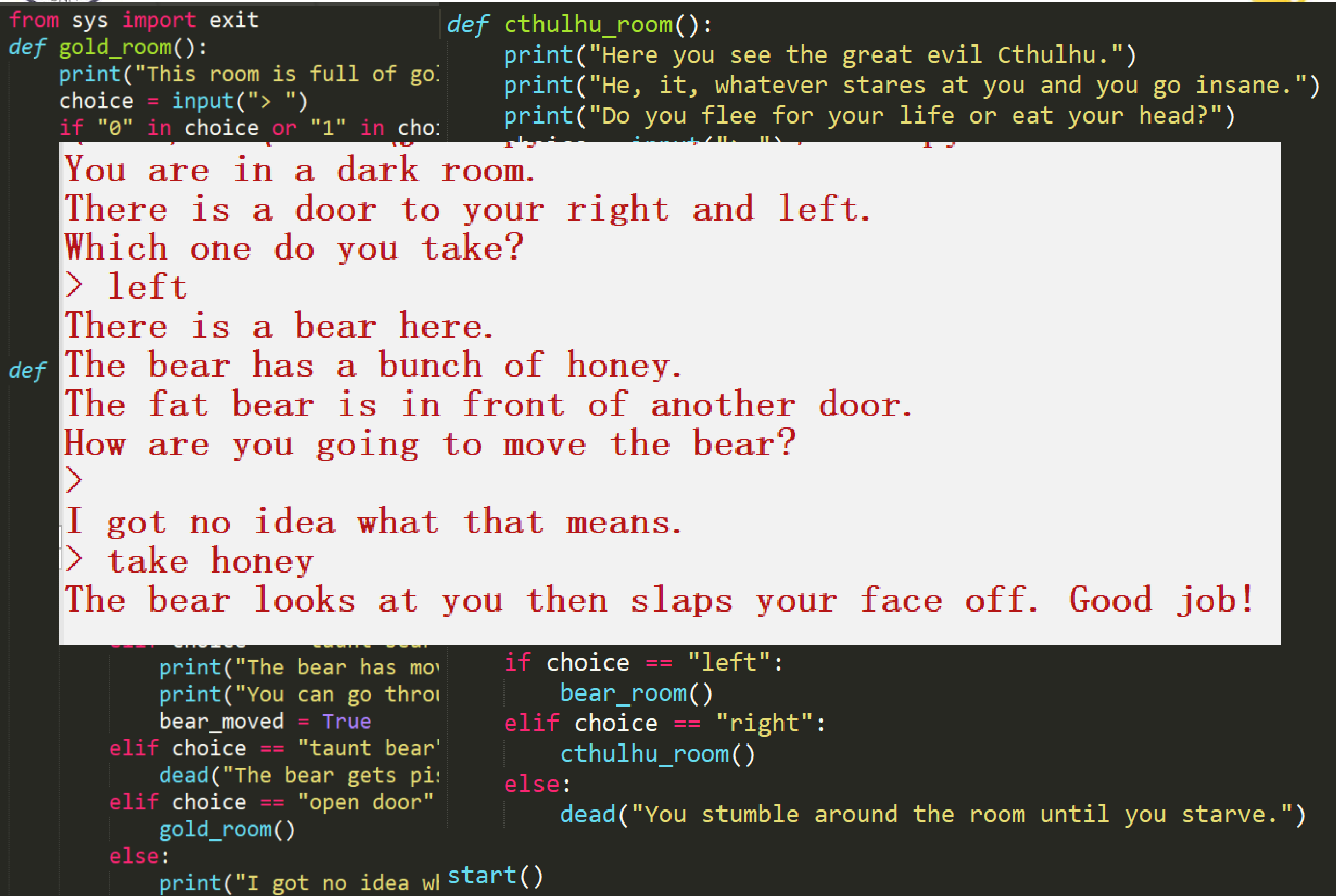


```
animals = ['bear', 'tiger', 'penguin', 'zebra']
bear = animals[0]
```

```
animals = ['bear', 'python3.6', 'peacock', 'kangaroo', 'whale', 'platypus']
```

1. The animal at 1.
2. The third (3rd) animal.
3. The first (1st) animal.
4. The animal at 3.
5. The fifth (5th) animal.
6. The animal at 2.
7. The sixth (6th) animal.
8. The animal at 4.

EX28: 分支与函数

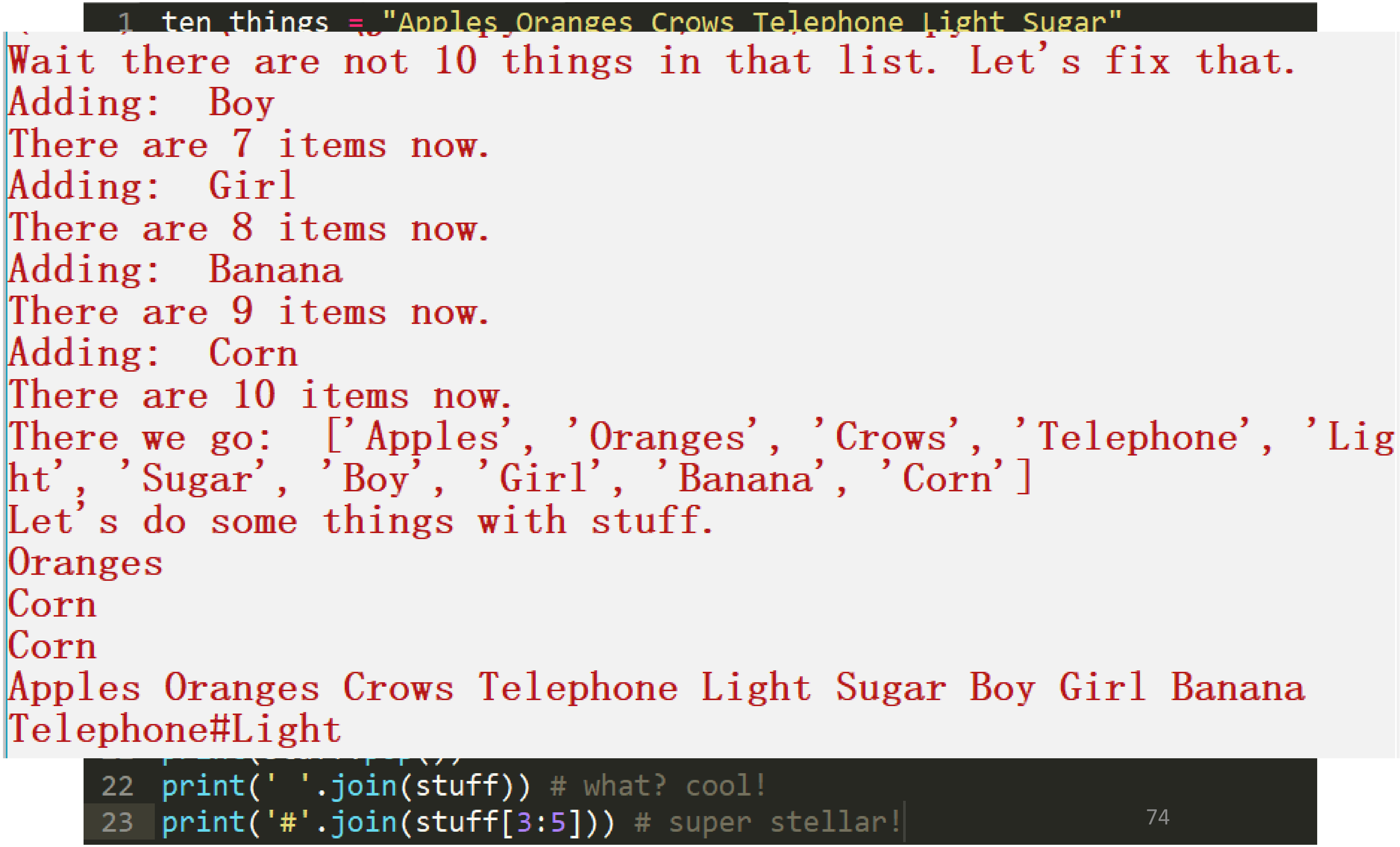


```
from sys import exit
def gold_room():
    print("This room is full of go
    choice = input("> ")
    if "0" in choice or "1" in cho

def

        elif choice == "taunt bear"
            print("The bear has mov
            print("You can go throu
            bear_moved = True
        elif choice == "taunt bear'
            dead("The bear gets pi
        elif choice == "open door"
            gold_room()
        else:
            print("I got no idea wh
```

```
def cthulhu_room():
    print("Here you see the great evil Cthulhu.")
    print("He, it, whatever stares at you and you go insane.")
    print("Do you flee for your life or eat your head?")
    choice = input("> ")

    if choice == "left":
        bear_room()
    elif choice == "right":
        cthulhu_room()
    else:
        dead("You stumble around the room until you starve.")

start()
```

```
You are in a dark room.
There is a door to your right and left.
Which one do you take?
> left
There is a bear here.
The bear has a bunch of honey.
The fat bear is in front of another door.
How are you going to move the bear?
>
I got no idea what that means.
> take honey
The bear looks at you then slaps your face off. Good job!
```

EX29: 列表

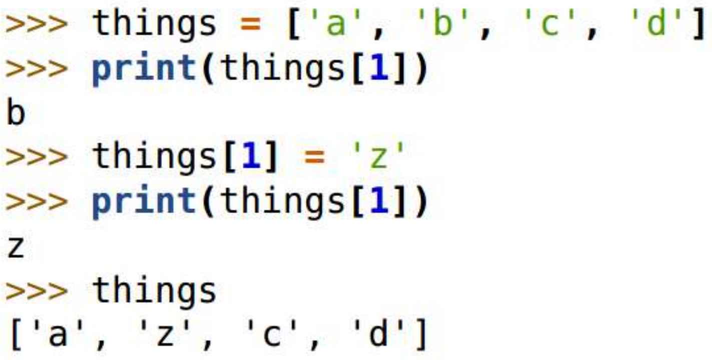


```python
ten_things = "Apples Oranges Crows Telephone Light Sugar"
```

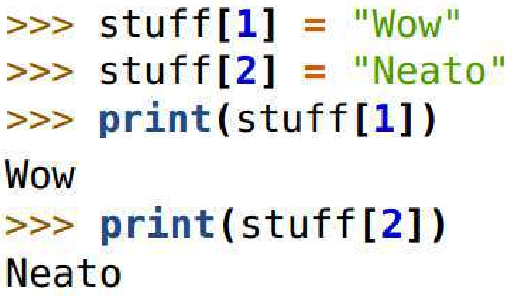
```
Wait there are not 10 things in that list. Let's fix that.
Adding:  Boy
There are 7 items now.
Adding:  Girl
There are 8 items now.
Adding:  Banana
There are 9 items now.
Adding:  Corn
There are 10 items now.
There we go:  ['Apples', 'Oranges', 'Crows', 'Telephone', 'Light', 'Sugar', 'Boy', 'Girl', 'Banana', 'Corn']
Let's do some things with stuff.
Oranges
Corn
Corn
Apples Oranges Crows Telephone Light Sugar Boy Girl Banana
Telephone#Light
```

```python
print(' '.join(stuff)) # what? cool!
print('#'.join(stuff[3:5])) # super stellar!
```

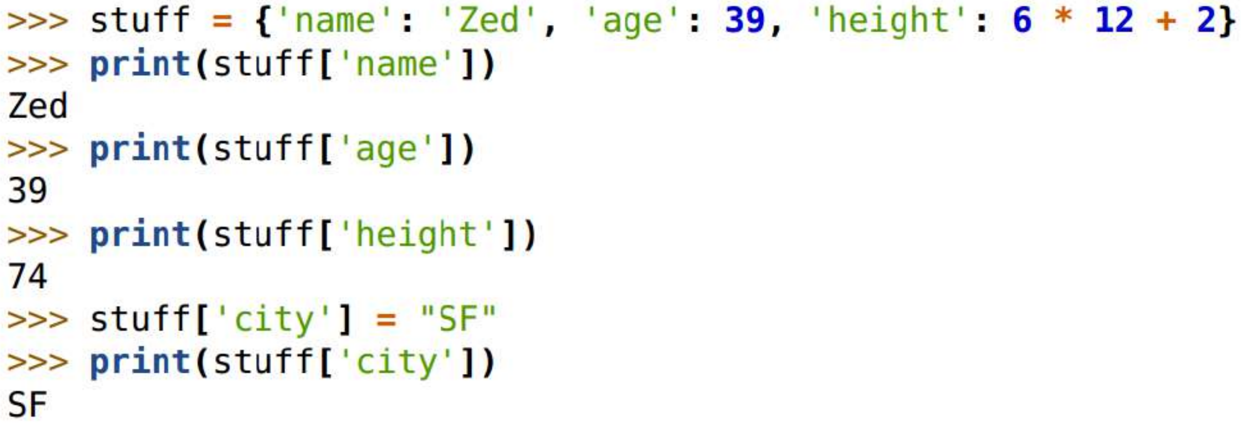
EX30： 字典



```
>>> things = ['a', 'b', 'c', 'd']
>>> print(things[1])
b
>>> things[1] = 'z'
>>> print(things[1])
z
>>> things
['a', 'z', 'c', 'd']
```

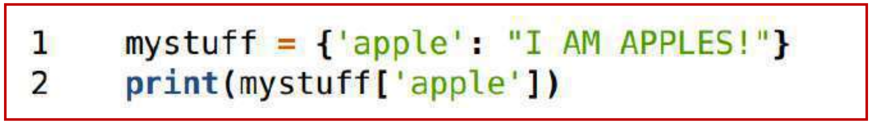


```
>>> stuff[1] = "Wow"
>>> stuff[2] = "Neato"
>>> print(stuff[1])
Wow
>>> print(stuff[2])
Neato
```

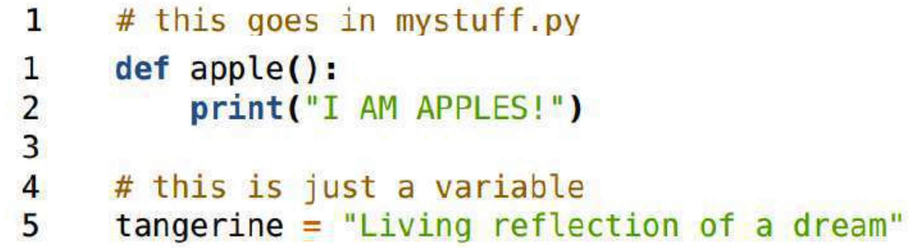


```
>>> stuff = {'name': 'Zed', 'age': 39, 'height': 6 * 12 + 2}
>>> print(stuff['name'])
Zed
>>> print(stuff['age'])
39
>>> print(stuff['height'])
74
>>> stuff['city'] = "SF"
>>> print(stuff['city'])
SF
```

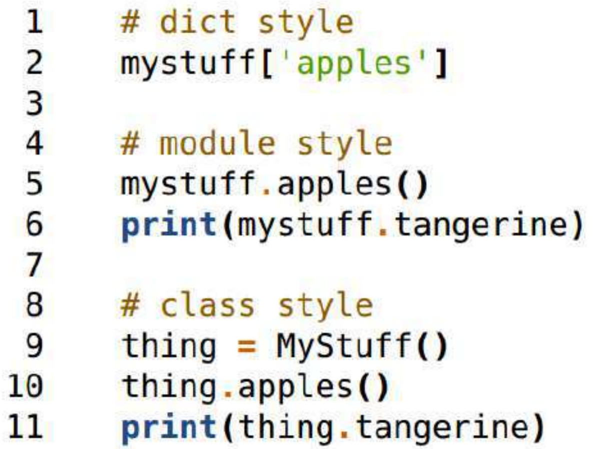
EX31：类



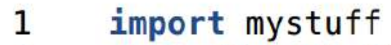
```
mystuff = {'apple': "I AM APPLES!"}
print(mystuff['apple'])
```



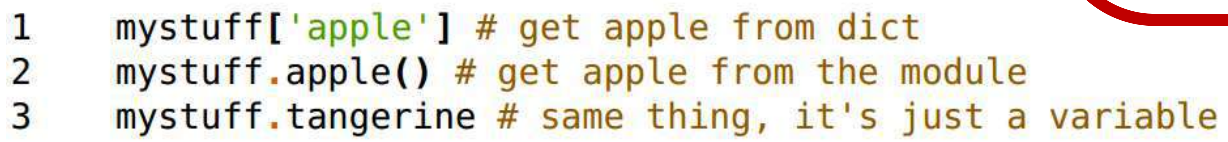
```
# this goes in mystuff.py
def apple():
    print("I AM APPLES!")

# this is just a variable
tangerine = "Living reflection of a dream"
```



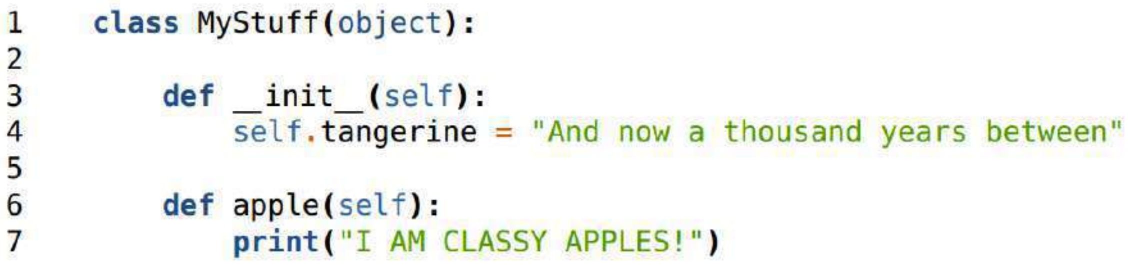
```
# dict style
mystuff['apples']

# module style
mystuff.apples()
print(mystuff.tangerine)

# class style
thing = MyStuff()
thing.apples()
print(thing.tangerine)
```



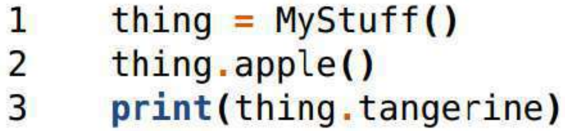
```
import mystuff
```



```
mystuff['apple'] # get apple from dict
mystuff.apple() # get apple from the module
mystuff.tangerine # same thing, it's just a variable
```



```
class MyStuff(object):

    def __init__(self):
        self.tangerine = "And now a thousand years between"

    def apple(self):
        print("I AM CLASSY APPLES!")
```



```
thing = MyStuff()
thing.apple()
print(thing.tangerine)
```

EX32 面向对象

class **告诉**Python定义一个新类型new type.

object 对象 最基本的类型/任何类型的实例。

Instance 实例 What you get when you tell Python to create a class.

def 定义类函数.

self 自索引/自访问

inheritance 继承 .

composition 组合车有四个轮子

attribute 属性：组成，变量

is-a 一种东西继承与另一种东西，企鹅是鸟

has-a 一种东西有个啥，例人有一只嘴巴

EX32 面向对象

- class X(Y) “定义 X类型是Y子类”
- class X(object): def __init__(self, J) “构造函数”
- class X(object): def M(self, J) “成员函数”
- foo = X() “X实例化.”
- foo.M(J) “调用成员函数”
- foo.K = Q “调用成员变量”