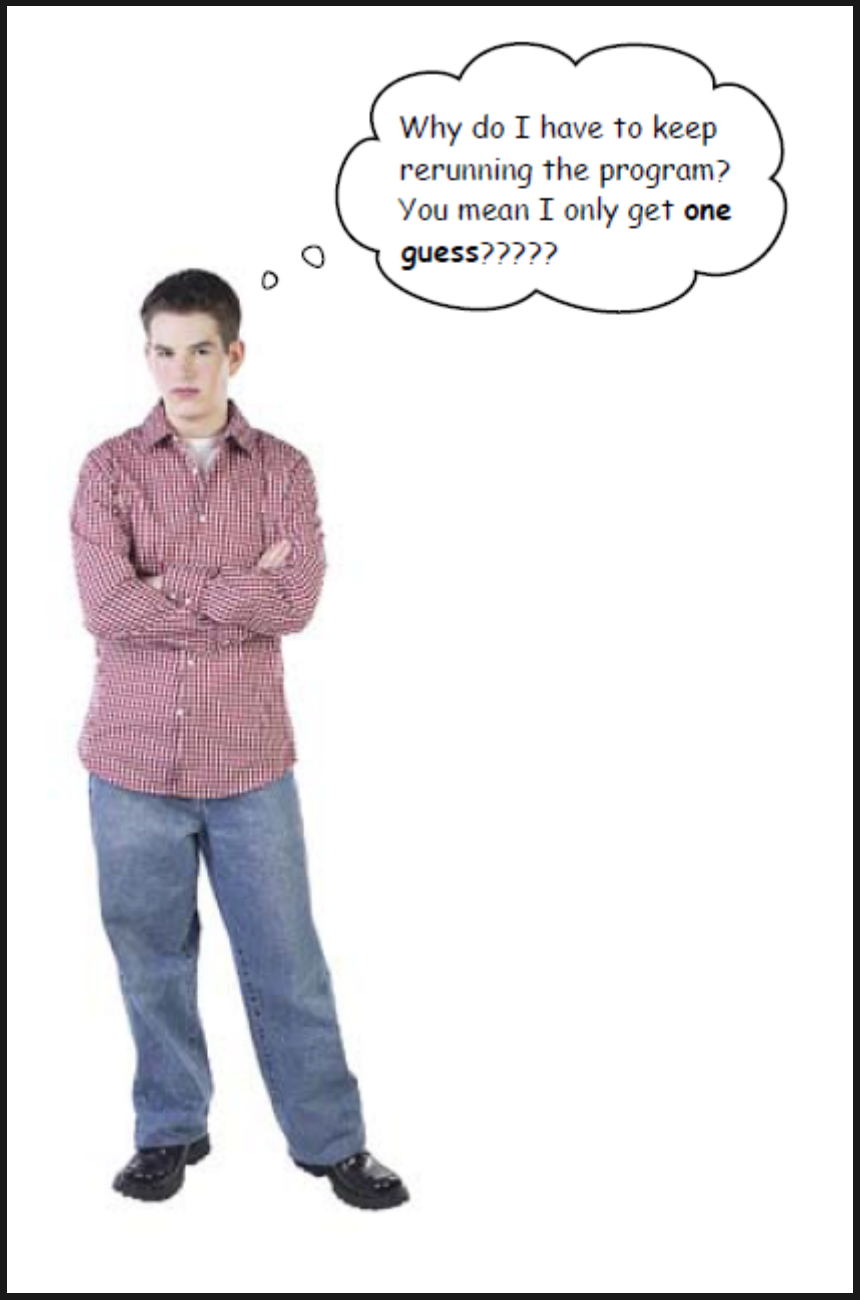
ITERATION

a process wherein a set of instructions or structures are repeated in a sequence a specified number of times or until a condition is met

UH OH: USERS ARE SALTY!


Why do I have to keep rerunning the program? You mean I only get **one guess**?????

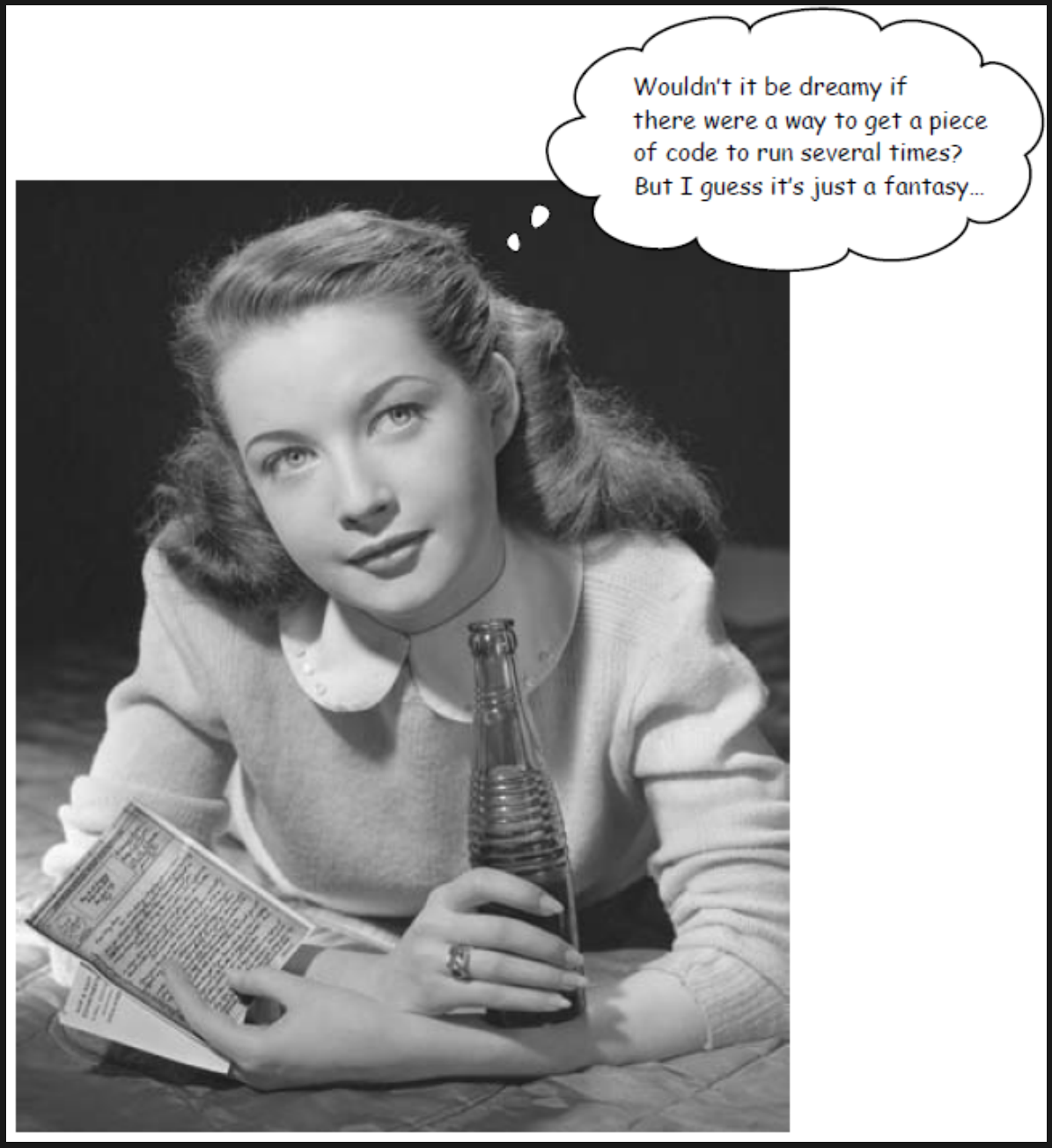
UH OH: USERS ARE SALTY!

Users are never happy!

The program works, and now generates extra feedback, but there's a problem. If the users want to have another guess, they have to run the program again. They really want the program to keep asking them for another guess until they finally get the correct answer.

How do we get the computer to do something repeatedly? Should we just make a copy of the code and paste it at the end of the file? That would make sure the user is asked twice. But what if they need to make 3 guesses? Or 4 guesses? Or 10,000 guesses? What about the case where the guess is correct?

The guessing game program needs to be able to run some code repeatedly.


Wouldn't it be dreamy if there were a way to get a piece of code to run several times? But I guess it's just a fantasy...

PYTHON TO THE RESCUE

Loops

Loops let you run the same piece of code over and over again

Programs often need to keep running the same piece of code many times. In addition to branches, programming languages also provide loops.

Loops are a little like branches. Just like branches, loops have a condition (the loop condition) that is either true or false. Also, like the if part of branches, if the loop condition is true, then a loop will run a given piece of code. For a branch, this code is called the body. For a loop, it’s called the loop body.

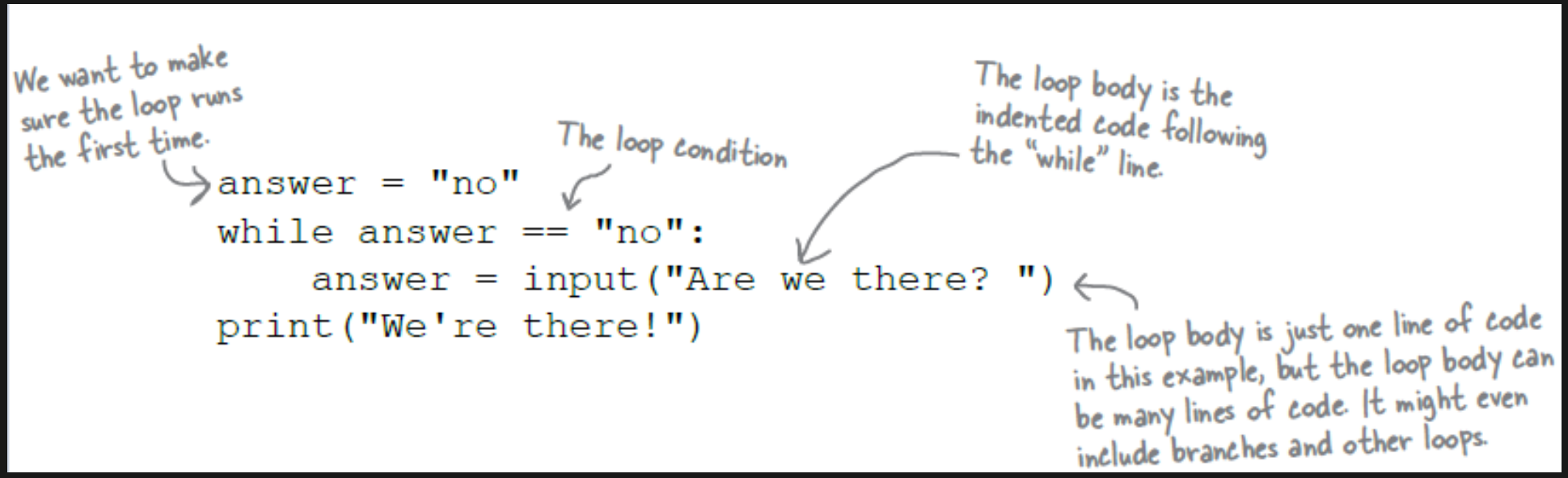
LOOPS

PYTHON'S WHILE LOOP

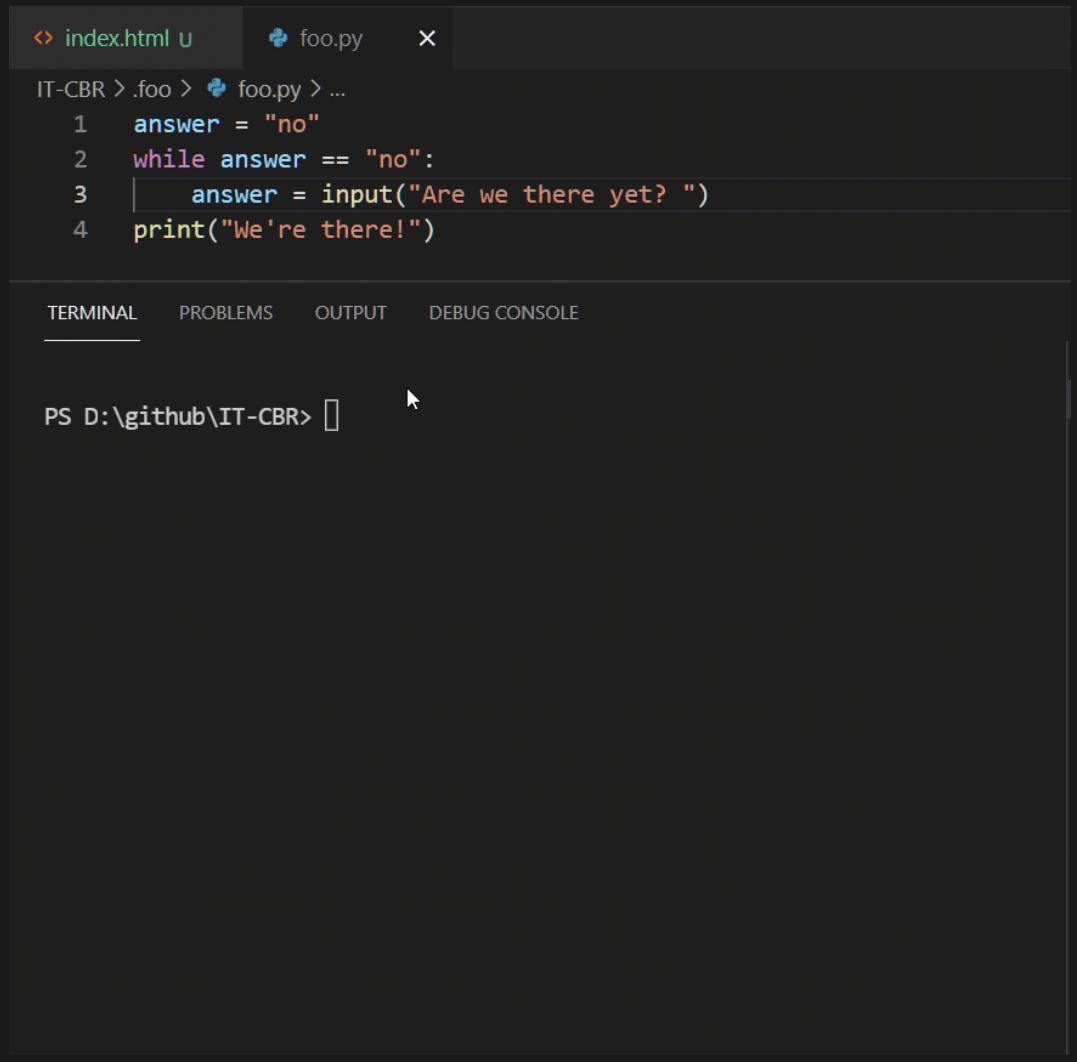
Programming languages have lots of different ways of creating loops, but one of the simplest ways in Python is to use a while loop.

```
1 answer = "no"
2 while answer = "no":
3     answer = input("Are we there yet?")
4 print("We are there!")
```

PYTHON'S WHILE LOOP



This is what the loop looks like when you write it as a Python while loop. The code keeps asking the question “Are we there?” until the user types something other than no.



Did you notice that you had to set the value of the answer variable to something sensible before you started the loop? This is important, because if the answer variable doesn't already have the value no, the loop condition would have been false and the code in the loop body would never have run at all.

EXERCISE!

Now, it's time to apply your programming mojo. Be warned: this exercise is kind of tricky. You need to rewrite your game program so it keeps running until the user guesses the correct answer. You will need to use all of the things you've learned in this chapter. You will need to work out the conditions for each of the branches and loops that are required.

Remember: the program needs to keep asking the user for an answer while the current guess is wrong.

Hint: If you need to test that two things have different values, use the != operator.

```
1 if "ada" != "bob":
2     print("Ada is not equivalent to bob!")
```

BUT NOW I KNOW THE ANSWER!

HOW CAN WE GET A RANDOM NUMBER BETWEEN 1 AND 10?

google it: python random number between 1 and 10

LIBRARIES

```
1 import random
2 foo = random.randint(1, 10)
3 print(foo)
```

output:

```
1 > 3
```

```
 1 from random import randint
 2 guess = 0
 3 answer = randint(1, 10)
 4 print("Welcome")
 5 while guess != answer:
 6     guess = int(input("Guess the number "))
 7     if guess > answer:
 8         print("Your guess is too high!")
 9     elif guess < answer:
10         print("Your guess is too low!")
11 print(f"Good guess! {guess} was the correct answer!")
12 print("Game over")
```

EVERYBODY LOVES YOUR GAME!