# PART 4: NATURAL LANGUAGE PROCESSING PROJECTS

# By Karen Schwarze

You’ve practiced formulating rules, and you’ve learned some Python syntax. Now you’re ready to dig into natural language processing by building three programs where the computer will perform actions with text you provide. Let’s get started!

**SIMILE GENERATOR**

Remember those rules that you wrote earlier about different literary devices? Now we’re going to use those rules to make a simile generator. When you’ve finished writing this program, you’ll be able to run it and get a different simile every time!

First, let’s revisit the rules you wrote earlier about a simile. They are as follows (your wording may be slightly different, but should contain these basic principles):

*A simile compares two things  
A simile always contains some form of ‘to be’, such as ‘is,’ ‘am’, ‘are’, or their past/future forms  
A simile always contains the word ‘like’ or ‘as’*

For the purposes of our simile generator, we’re going to parse down these requirements a bit. First off, our similes will compare two specific categories of things—abstract nouns and concrete nouns (more on those below). Also, our similes only use the singular, present tense of *to be (is).* And finally, our similes will only contain *like,* not *as.*

Abstract Nouns and Concrete Nouns

An **abstract noun** is an idea or feeling that can’t be seen, touched, felt, or otherwise detected with our five senses (smell, sound, touch, taste, sight). A **concrete noun** can be detected with our senses in some way—just like how concrete, the material used to make sidewalks, can be seen and touched. Abstract nouns include things like love, happiness, anger, disappointment, and victory. Concrete nouns include things like water, the sun, ice cream, puppies…you get the idea.

Before we start coding, we need to map out some ideas on paper. Just as you write an outline before writing an essay, we can write **pseudocode** before we code our program. Our simile generator is going to be very similar to the Exercise 1 you did at the beginning of the section, where you wrote a program that randomly generated a group of words containing a verb, an adjective, and a noun. We’re going to use the import random module again so that our program can randomly select from two lists (our abstract nouns and our concrete nouns). Then, we’re going to tell the program to assign values from those respective lists to two variables. Next, the program will create a simile using a randomly generated abstract noun and concrete noun, joined by the phrase ‘is like’. Finally, we’ll tell the program to display the simile using the print function.

Let’s write out the steps above using some pseudocode. Then we’ll translate it into Python.

Use the import random module.

Create two variables, each containing a list of values.

Randomly assign a value from each list to a variable.

Create a simile using a randomly generated abstract noun and concrete noun, separated by the phrase ‘is like’.

Display the simile using the print function.

Note that all we did to write our pseudocode was to break down the steps. Now, go back and look at the code we wrote for Exercise 1. See if you can use it to help you get close to writing the actual code for your simile generator. Your code will probably be mostly **skeleton code**—code that is more fleshed out than pseudocode but not quite real code yet. Use the space below to write your skeleton code.

(If you what the real code is, you can write that instead of the skeleton code. The goal here is for you to do your best to write the closest thing you can to what the real code will look like before moving on to the answer). Remember that you’ll need to write two lists—one for abstract nouns and one for concrete nouns. Choose any abstract nouns and concrete nouns you want to use—these are your similes!

Write your skeleton code here

After you’ve written some skeleton code, check it against the answers below. As you read the answer, make sure you understand why I wrote the code this way. If you’re not sure, ask for help!

I’ve written out each step followed by the Python code. Remember that your abstract and concrete nouns will probably be different from mine.

1. Use the import random module  
   import random
2. Create two variables, each containing a list of values   
   abstract\_nouns = [‘Love’, ‘Forgiveness’, ‘Impatience’, ‘Penitence’]  
   concrete\_nouns = [‘wind’, ‘light’, ‘the sea’, ‘darkness’]
3. Randomly assign a value from each list to a variable  
   abstract\_noun = random.choice(abstract\_nouns)  
   concrete\_noun = random.choice(concrete\_nouns)
4. Create a simile using a randomly generated abstract noun and concrete noun, separated by the phrase ‘is like’.   
   simile = abstract\_noun + ‘is like’ + concrete\_noun
5. Display the simile using the print function.  
   print(simile)

Ready to see your simile generator in action? Put all of the code above into IDLE, save it as a new file, and select Run > Run Module. Here’s the code again for you:

import random

abstract\_nouns = [‘Love’, ‘Forgiveness’, ‘Impatience’, ‘Penitence’]  
concrete\_nouns = [‘wind’, ‘light’, ‘the sea’, ‘darkness’]

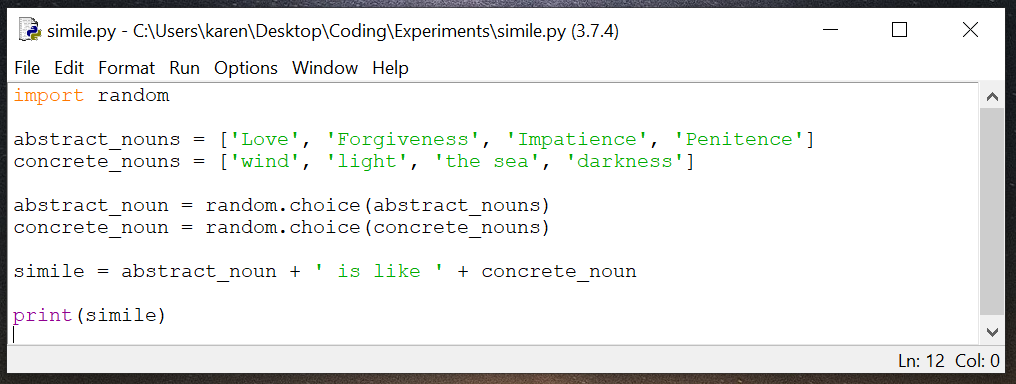
abstract\_noun = random.choice(abstract\_nouns)  
concrete\_noun = random.choice(concrete\_nouns)

simile = abstract\_noun + ‘is like’ + concrete\_noun

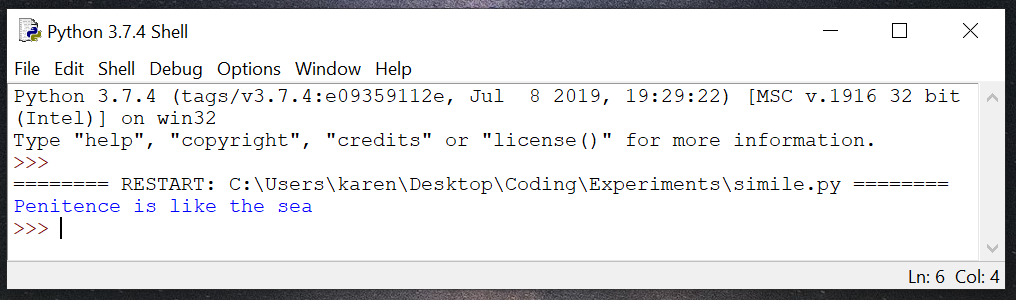
print(simile)

Keep selecting Run Module to generate new similes. Show your classmates your similes! What kind of abstract nouns and concrete nouns did they choose? What kind of similes are their programs generating?

What my program looks like in IDLE before I run it:



What my program looks like after I hit Run > Run Module:



What kind of similes does your program generate?