Python Cookies Engagement Activity – Bake Sale Planning

Introduction

This activity is designed to provide a gentle introduction to functions in Python. Functions are a lot like recipes – reusable processes. In this engagement activity, you will create a bake sale planner by writing your own custom functions and by using built-in String formatting functions.

Preliminaries

a. Navigate to Repl.it and start up a new Python repl

Challenge 1. A cookie recipe…

Functions are a lot like recipes – reusable "chunks of computation". What better way to practice with functions that to develop one for a cookie recipe?

For this engagement activity, we're going to focus on cookies which have two major components – their dough and their fillings/toppings. Cookies can also have a name. That means we can get some normal cookies like:

- Chocolate chip cookie: sugar cookie dough and chocolate chips
- Oatmeal-raisin cookie: oatmeal dough and raisins

And also some really weird cookies, like:

- Why-did-you-let-your-kid-into-the-kitchen-alone? cookies: instant cake mix "dough" and gummy worms

Unfortunately, Python cannot bake you a cookie (yet…). However, we can make a function which simulates cookie baking! Below is a header for the function:

```
def bake_cookie(dough, filling, cookie_name):
```

In this case, the arguments are:

dough – A string, for the dough's type

filling – A string, for the type of topping or filling as appropriate. Filling as chosen as a variable name here

cookie_name – The type of the cookie.

Now it's your turn – complete the function to make and return a string describing the cookie. Here are some test cases to run as an example:

```
1. result = bake_cookie("sugar","chocolate chips", "chocolate chip cookie")
   #result should have the value "A chocolate chip cookie made with sugar
   dough and chocolate chips."
```

```
2. result = bake_cookie("sugar", "cinnamon", "snickerdoodle")
   #result should have the value “A snickerdoodle made with sugar dough and
   cinnamon.”

3. result =  bake_cookie("oatmeal", "raisin", "oatmeal-raisin cookie")
   #result should have the value “A oatmeal-raisin cookie made with oatmeal
   dough and raisin.”
```

Challenge 2. Making some improvements with built-in functions

It is likely that, when you were building the string bake_cookie outputs, that you had to concatenate (or add together) a lot of string literals and variables. This is not an ideal way to make a string with variables. Python's strings have a built-in *format* function which helps simplify things greatly! For your current bake_cookie implementation, try substituting the following code:

```
def bake_cookie(dough, filling, cookie_name):
  message_string = "A {} made with {} dough and {}."
  return message_string.format(cookie_name, dough, filling)
```

The test cases above should still work! This uses Python's built-in formatting method to simplify the message down by dividing responsibilities – message_string is a standard format string that can be reused, and .format fills the placeholders (denoted with {}) with the variables. More handy details can be found here - https://pyformat.info/

Challenge 3. One step further – an actual bake sale program

For the next final step, you're going to use the bake_cookie function and build an actual bake sale inventory generator. For this task, we're providing some starter code:

```
user_selection = input("Please either enter 'cookie' to enter a cookie or 'q' to
stop the program:")
bake_sale_flyer = ""


while user_selection != 'q':



  user_selection = input("Please either enter 'cookie' to enter a cookie or 'q'
to stop the program:")

print(bake_sale_flyer)
```

We suggest approaching this challenge in a few steps, enumerated as follows:

1. **Step 1** – Finish the while loop to get cookie information from the user

   The first task we suggest you complete is actually using the provided while loop to prompt user input. ***If*** the user enters “cookie”, you should then actually bake cookies. Specifically, you will need to prompt for a dough type, a filling type, and the name of the cookie.

Complete the while loop to prompt the user for these inputs and print out the cookie. An example for testing is given. Please note that you may need the newline character, \n, for some of the spacing:

```
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
What dough are you using? sugar
What topping or filling will you use? chocolate chips
And what is this cookie called?  chocolate-chip cookie

A chocolate-chip cookie made with sugar dough and chocolate chips.

Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
What dough are you using? Sugar
What topping or filling will you use? Cinnamon
And what is this cookie called?  Snickerdoodle

A snickerdoodle made with sugar dough and cinnamon.

Please either enter 'cookie' to enter a cookie or 'q' to stop the program: q
```

2. **Step 2** – Adding a quantity

   The current application just prints out a single cookie when the ingredients are entered. This won’t do – we should be able to bake batches of cookies! We need to add a measure of quantity to our Python code.

   We wanted to open up the activity to your creativity here – it’s up to you how you want to add the quantity onto your messages. However, a sample output for what we’d like to see is given:

```
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
What dough are you using? sugar
What topping or filling will you use? cinnamon
What is this cookie called?  snickerdoodle
And how many are you baking? 26

26 count: snickerdoodle made with sugar dough and cinnamon.
```

3. **Step 3** – Producing a bake sale inventory

You may have noticed the ominous `bake_sale_flyer = ""` variable. That's because the last step is to use that variable to store the whole bake sale flyer text and then print it at the end. Accomplishing this is just a series of string concatenations – you'll want to take the message you were printing out before and add it to the `bake_sale_flyer`. The end result should look something like this:

```
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
What dough are you using? sugar
What topping or filling will you use? cinnamon
What is this cookie called?  snickerdoodle
And how many are you baking? 26
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
What dough are you using? peanut-butter and sugar
What topping or filling will you use? sugar crystals
What is this cookie called?  peanutbutter cookie
And how many are you baking? 26
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: q
!!Bake Sale!!
26 count: snickerdoodle made with sugar dough and cinnamon.
26 count: peanutbutter cookie made with peanut-butter and sugar dough and sugar crystals.
```

4. **Step 4** – Cookie Price

The current bake sale flyer is nice, but it might be useful to also add some concept of cookie price to the flyer. For this activity, we'll provide the price for each batch – but calculate it using the price per cookie!

To do this, prompt the user for a price per cookie in the while loop. Then, *write a new custom function* which takes a cookie count and a price per cookie. Here's some hints:

- input() reads in data as a string – **always**. If you want any decimal points in your numbers, you will want to change them to floating point numbers using the function *float*()
- We'd like for the function you write to return the price as a formatted string with a dollar sign and specifically two decimal places. We're giving you a partially filled out format string below, but you'll need to finish it – see https://pyformat.info/#number for how to get the decimal places!
- `currency_format = "${:f}"`

  Provided is one more sample output to show how we included the price in out bake sale flyer

  ```
  Please either enter 'cookie' to enter a cookie or 'q' to stop the program: cookie
  What dough are you using? sugar
  What topping or filling will you use? cinnamon
  What is this cookie called?  snickerdoodle
  And how many are you baking? 26
  How much does a single cookie cost? .50
  ```

```
Please either enter 'cookie' to enter a cookie or 'q' to stop the program: q
!!Bake Sale!!
26 count for $13.00: snickerdoodle made with sugar dough and cinnamon.
```

Spicy Extras

1. Add habaneros to your chocolate-chip cookies. It makes for a zesty kick!
2. We did not dig too far into string format in this challenge. Looking at the provided reference doc, are there any ways you could make the bake sale print out more attractive? Something like:

```
Cookie Type                                         Count        Price
snickerdoodle made with sugar dough and cinnamon    26           $13.00
```

3. Right now, cookie price is presented as a bulk for a given batch. Can you think of other ways to show the price and format the flyer?
4. We did not prescribe custom functions beyond two for this engagement activity. Can you think of any other functional decompositions you can make to further divide up responsibility?