We are going to write a Python program that relates to the four fundamental operations that are needed for even the simplest applications and softwares. These functions -- create, read, update, and destroy -- are generally necessary for any software to be considered complete.

These four functions are so fundamental that they're often referred to by the acronym CRUD.

To do this, we will be creating a sort of “Cookbook” checklist Python Program, making sure that we can create, read, update, or destroy recipes from this cookbook. We can be sure that at the very least our most basic operations can be met. Though we will need some more tools in addition to these four here. You will be creating those as well!

Getting Started

- **STEP I :** To begin with we'll create our new file cookbook.py file. You should be able to verify that a new empty file was created named cookbook.py

  By focusing on our desired results, we're more likely to stay on track with the features that we need instead of creating features for the sake of having features. An easy way to do this is with user stories. You have a user that desires a feature so that they can do something.

  As [a type of user], I want [some feature] so that I can [do something].

  With user stories you want to approach a problem from the perspective of the user. Different types of users will have different experiences with the application. Here we only have one type of user in mind so we won't have to worry about what different types of users will or won't be able to see.

  In our case our user stories will look something like this:

  As a user, I want to be able to create, read, update, and destroy items in a checklist.

  As a user, I want to be able to mark or check off items so I can know that it's already represented.

  As a user, I want to be able to see everything in my list at once so I know what is in my list.

  From these above user stories we can generate a list of important features and functionalities that our users will expect from our application.

Create, Read, Update, and Destroy

- **STEP I :** Create a list object in memory that we can refer back to by the name cookbook.

  Now we have our cookbook but we can't use it yet. Ideally you will want to be able to re-use as much code as possible. This means breaking it up into small chunks that can be called on when needed. These chunks of code are called functions and are how we will structure most of our program. We will create functions to help with our functional needs.

- **STEP II :** Define the **create function**, which uses the appropriate list class method to **add** the parameter *recipe* to our checklist. Using this *recipe* parameter allows us to add any item we desire later when calling the function. Include a print statement that shows that a recipe has been added to the cookbook.

- **STEP III :** Define the **read function**, which uses the **index** of our list to access any item in the list and returns it. Since we need the index in order to retrieve a value, we are using *index* as a parameter. Be sure to check that the index is in range!

- **STEP IV :** Define the **update function**, which uses the index again not only to access the value at an index, but **change** or **update** that value as well. Just like our last function we'll need the index of the desired item, but we'll also need to know what we want to update it with. This function will need an additional parameter, *recipe*. Print a statement that shows what the recipe has been updated to. Be sure to check that the index is in range!

  So that takes care of the first three letters, C, R, and U. Now we just need the last one - Destroy.

- **STEP V :** Define the **destroy function**, which uses the appropriate list class method to remove an item at a given index. Print a statement that shows what recipe has been removed. Be sure to check that the index is in range!

  We have finished with our basic CRUD functions. However, there are some more things we should do before we can really say this cookbook is as helpful as it can be! For example, right now we can see a single item if we know its index, but we should be able to see the entire list.

- **STEP VI :** Define a **list_all_recipes function** that allows you to see every recipe that is in the cookbook. In order to view the entire list at once we'll need to get every individual item in the list.