**LOOPS**

**What are Loops?**

In our everyday lives, we tend to repeat a lot of processes without noticing.

For instance, if we want to cook a delicious recipe, we might have to prepare our ingredients by chopping them up. We chop and chop and chop until all of our ingredients are the right size. At this point, we stop chopping.

If we break down our chopping task into a series of three smaller steps, we have:

1. An initialization: We’re ready to cook and have a collection of ingredients we want to chop. We will start at the first ingredient.
2. A repetition: We’re chopping away. We are performing the action of chopping over and over on each of our ingredients, one ingredient at a time.
3. An end condition: We see that we have run out of ingredients to chop and so we stop.

In programming, this process of using an initialization, repetitions, and an ending condition is called a [*loop*](https://www.codecademy.com/resources/docs/python/loops?page_ref=catalog). In a loop, we perform a process of *iteration* (repeating tasks).

Programming languages like Python implement two types of iteration:

1. *Indefinite iteration*, where the number of times the loop is executed depends on how many times a condition is met.
2. *Definite iteration*, where the number of times the loop will be executed is defined in advance (usually based on the collection size).

Typically we will find loops being used to iterate a collection of items. In the above example, we can think of our ingredients we want to chop as our collection. This is a form of definite iteration since we know how long our collection is in advance and thus know how many times we need to iterate over the collection of ingredients.

Some collections might be small — like a short string, while other collections might be massive like a range of numbers from 1 to 10,000,000! But don’t worry, loops give us the ability to masterfully handle both ends of the spectrum. This simple, but powerful, concept saves us a lot of time and makes it easier for us to work with large amounts of data.

In this lesson, we’ll learn how to use Python to implement both definite and indefinite iteration in our own programs.

**Instructions**

Look over (and over) the provided diagram. Then, go to the next exercise to get looped in!

