## Start here:

```
taco3.fill("chicken")

taco3.fill("guacamole")
```

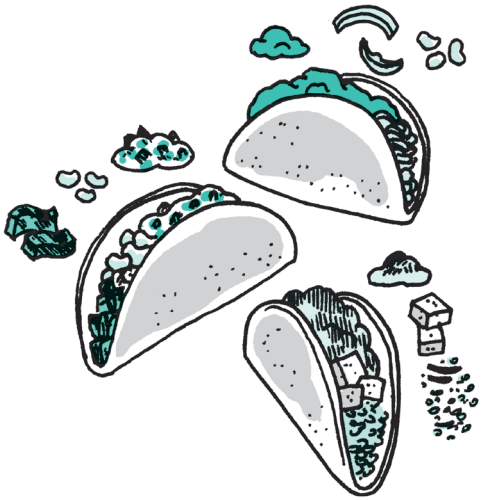
Wow, that's a lot of code and typing! Instead of writing this code over and over again for all of Lucy's tacos, you can write a loop to repeat it for you.

```
for each taco in lucysOrder:

    taco.fill ("chicken")

    taco.fill ("guacamole")
```

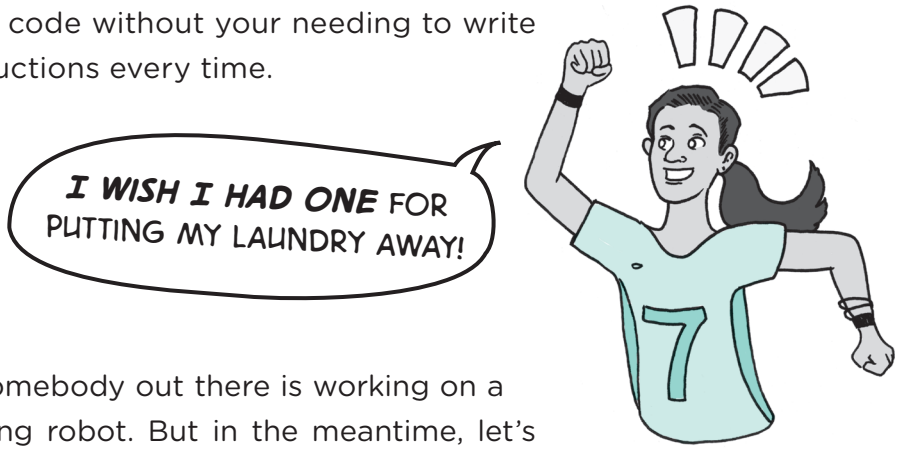
Now all three of your tacos will be filled the same way, and our code is only three lines. But what if we wanted to use different fillings for each taco? Let's take a look at functions!



## FUNCTIONS

Functions, the fourth member of the Core4 concepts along with variables, loops, and conditionals, are little bunches of code that do specific jobs in a larger program. You can create them for lots of reasons, but a

big one is to help you avoid having to write the same code over and over again. All you have to do is write a function and name it, and then any time you want to use it, you call it up in your code by typing in its name, and it will run your bunch of code in that part of your program. It's like creating a button or a shortcut to automatically do a job in your code without your needing to write out the instructions every time.




I'm sure somebody out there is working on a laundry-folding robot. But in the meantime, let's go back to our Lunch Line Algorithm and take a look at how a function can work in a program.

## DEFINING YOUR FUNCTION

The first step to making any function is to define it. Give it a name—otherwise your program won't know how to find that batch of code when it's time to use it. Names should be clear and descriptive. So, in this case, let's name our function the `fill_my_taco` function!

Once you've named it, you have to write the code for it. Part of doing that means creating **PARAMETERS** in your function. Parameters are like little mini variables inside a function. They hold information that may change.

In code it would look like this:

```
def fill_my_taco(taco, item1, item2):

  taco.fill(item1)

  taco.fill(item2)
```

The parameters are the content in parentheses. In this `fill_my_taco` function we can fill our taco with two different items.

## CALLING ALL FUNCTIONS

Once you have your function built, all you have to do is actually use it in the code. This is an important step. It's not enough just to define and build a function; you have to tell the computer when and where you want to use it. You do that by typing the function name in the part of the code where you want it to run. This is known as "calling" the function.

Calling the function in the code would look like this:

```
fill_my_taco(taco1, "chicken", "guacamole")
```

And we can try using different fillings for our other tacos as well!

```
fill_my_taco(taco2, "tofu", "salsa")
```

```
fill_my_taco(taco3, "beans", "vegetables")
```

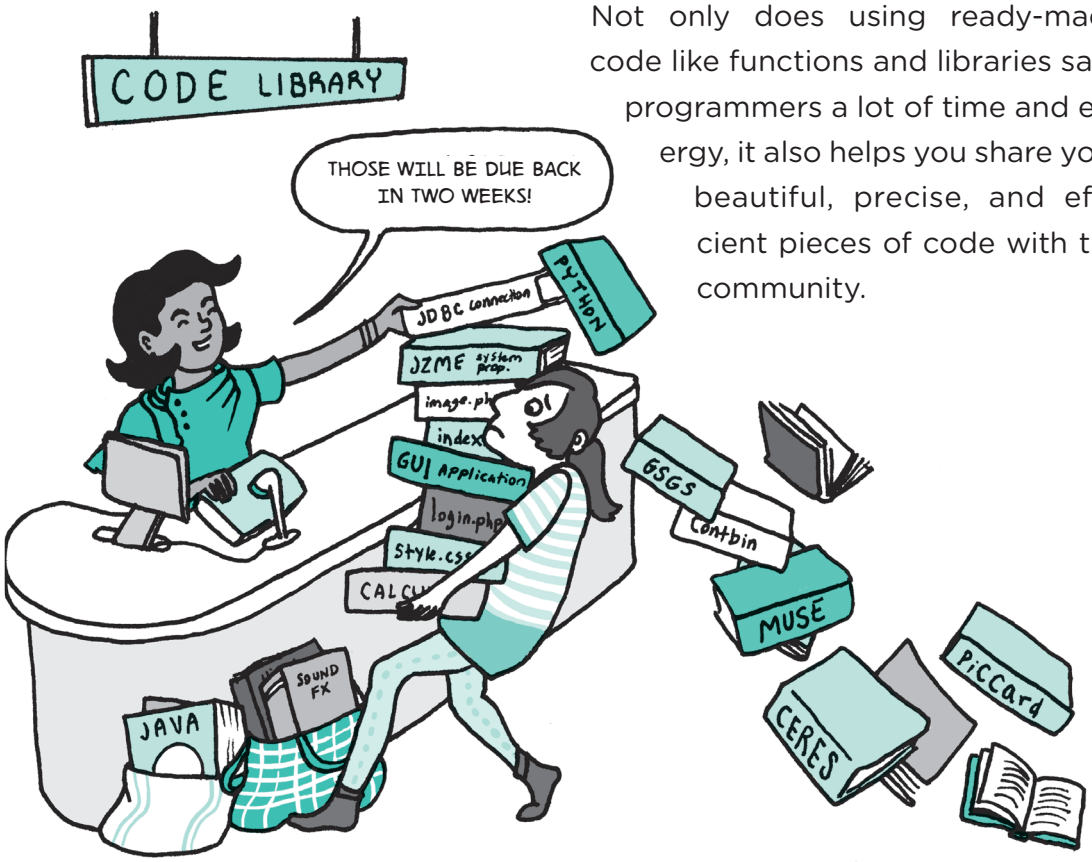
Now we have three different tacos. Mmmmm . . . variety!

Functions can also be represented as buttons or shortcuts on the user side of a program, like the print button in a word processing app or the back button on a web browser. Click any one of those buttons when you are using that program, and it will call up the function in the code and activate it.

## LIBRARIES

Functions are huge time-savers when you really start coding. And they aren't the only shortcut in coding. Libraries are also an important resource for coders. We're not talking about your local library, where you go to check out books. (Yay, we ♥ libraries!) Code libraries store useful algorithms and pieces of code that someone has already written so that others can use them in their programs. These libraries let programmers access all kinds of handy code:

- ★ Algorithms for searching, like the kind that power Google or Microsoft's search engine.
- ★ Algorithms for sorting that help you categorize data different ways—alphabetically, by number, by name, etc.
- ★ Recommendation algorithms, which take input or search patterns and use them to make suggestions for a certain user, just like the "movies for you" on Netflix or products on Amazon.




Not only does using ready-made code like functions and libraries save programmers a lot of time and energy, it also helps you share your beautiful, precise, and efficient pieces of code with the community.