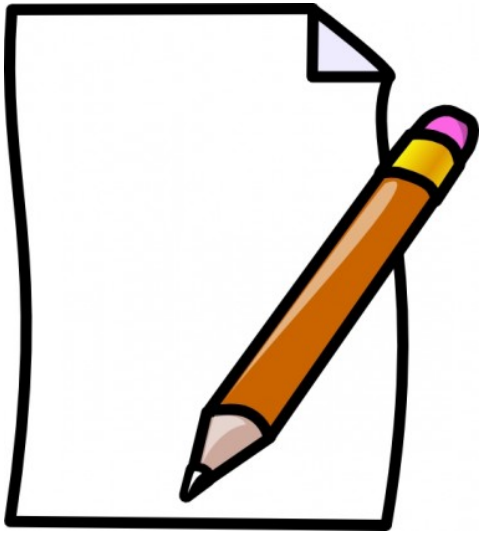
Unit 7 - Lesson 1
Parameters and Return Explore

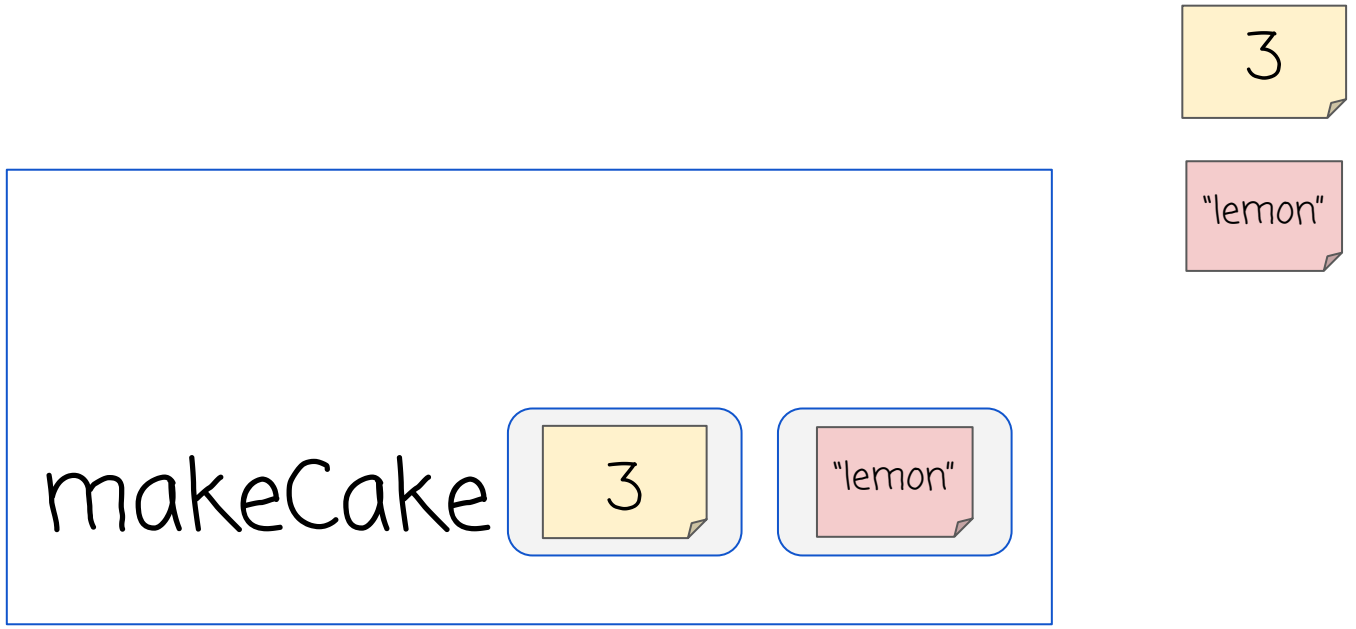
Warm Up

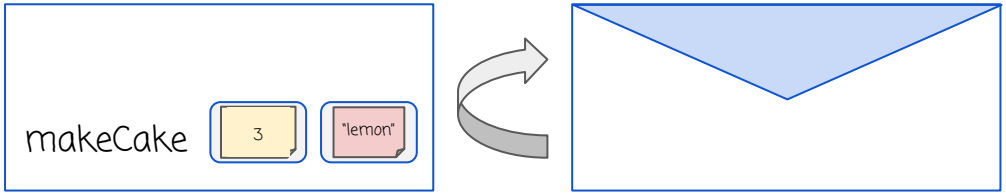
For this lesson, you will need:

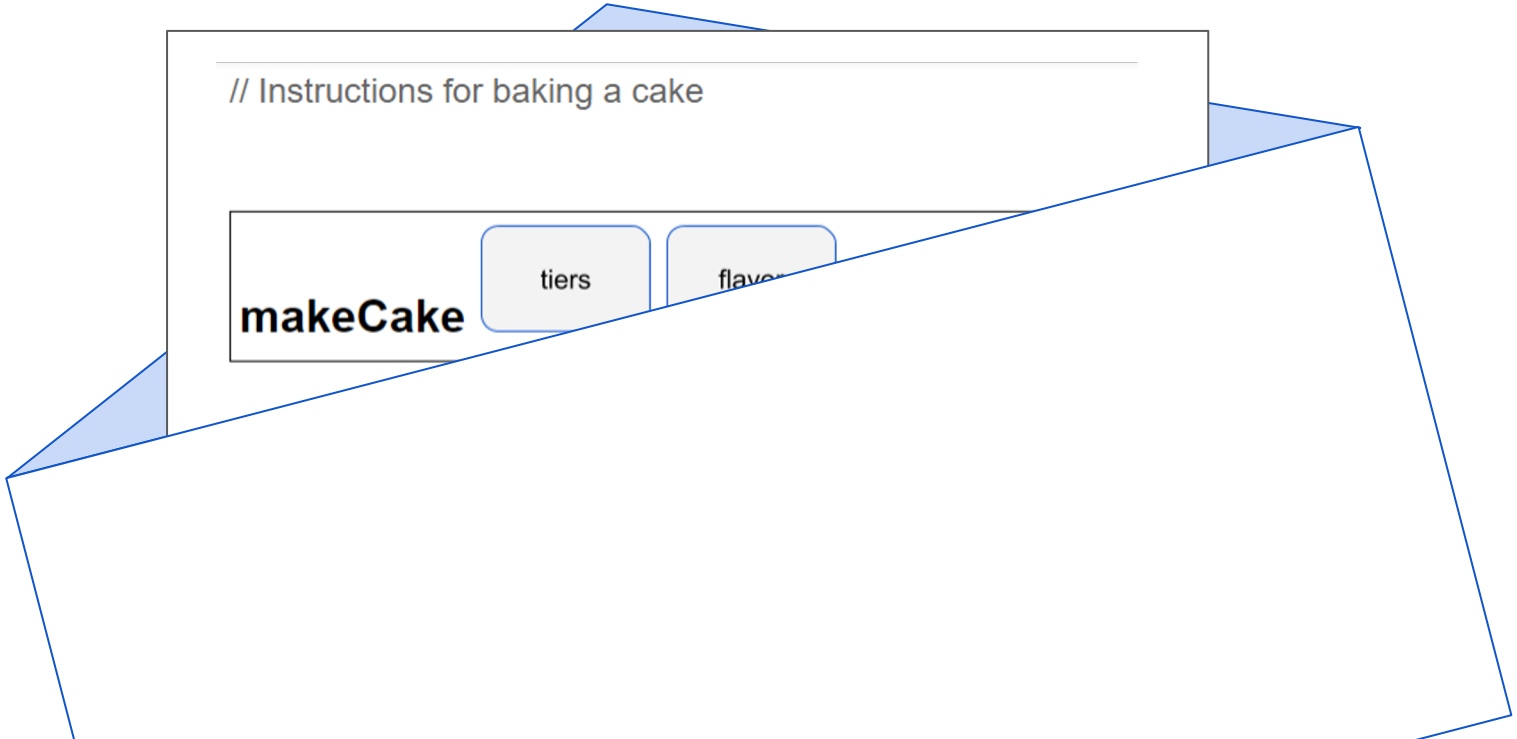
Sheet of paper
Pen / Pencil

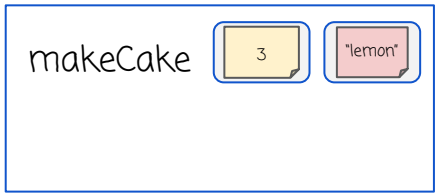


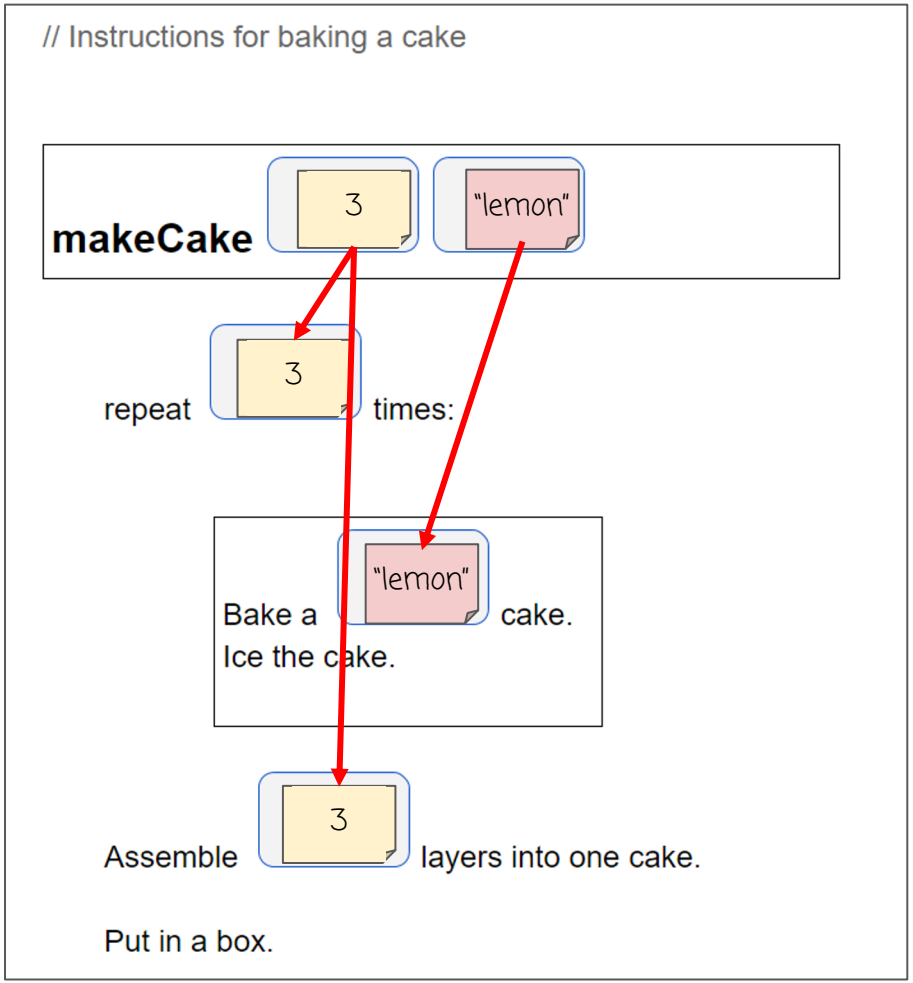
Activity


3
"lemon"
makeCake
3
"lemon"

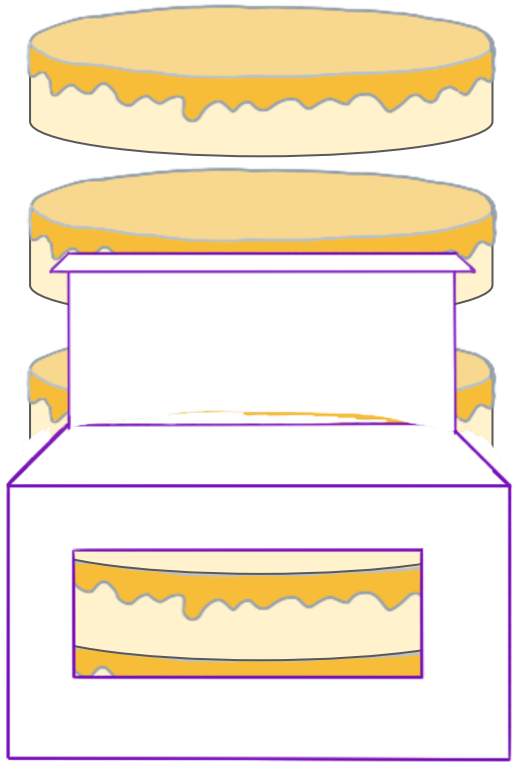

makeCake
3
"lemon"


// Instructions for baking a cake
makeCake
tiers

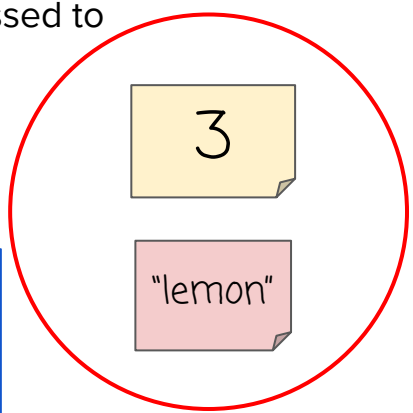

makeCake
3
"lemon"

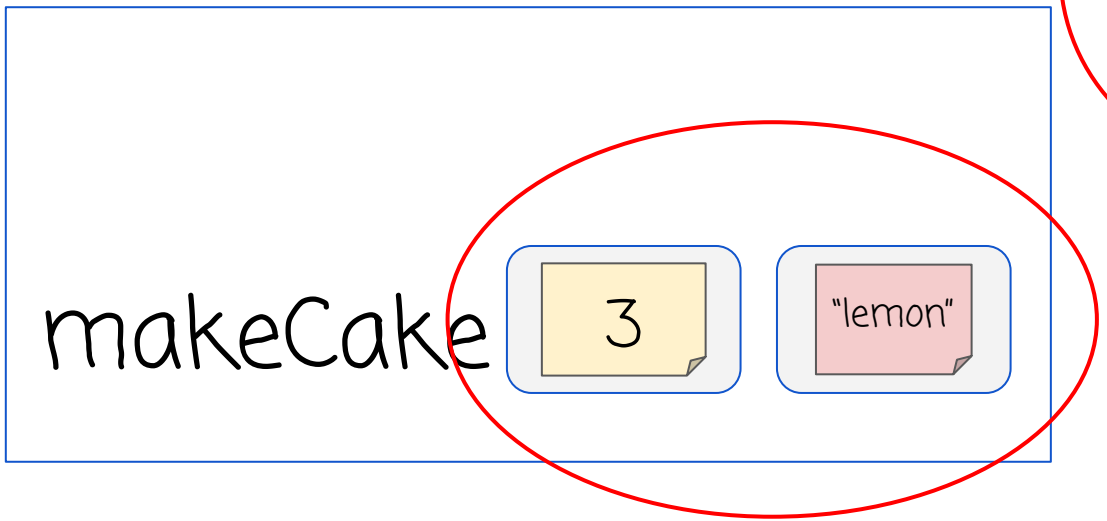

// Instructions for baking a cake
makeCake
3
"lemon"
repeat
3
times:
Bake a
"lemon"
cake.
Ice the cake.
Assemble
3
layers into one cake.
Put in a box.

```
// Instructions for baking a cake

makeCake 3 "lemon"

repeat 3 times:
    Bake a "lemon" cake.
    Ice the cake.
Assemble 3 layers into one cake.

Put in a box.
```



Argument - the value passed to the parameter



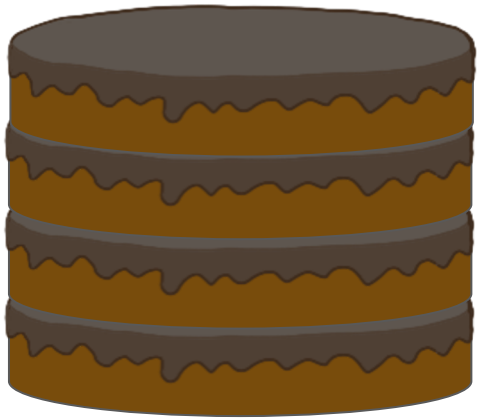


Parameter - a variable in a function definition. Used as a placeholder for values that will be passed through the function.

What if I wanted a four layer chocolate cake? What would that look like?

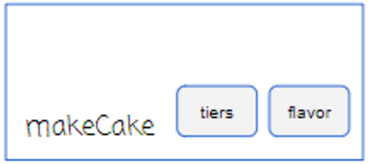
4 "chocolate"

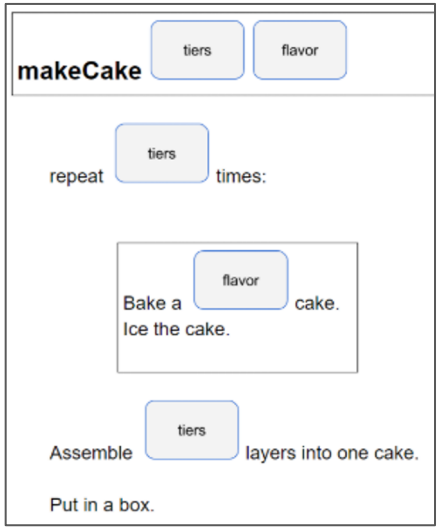
makeCake 4 "chocolate"



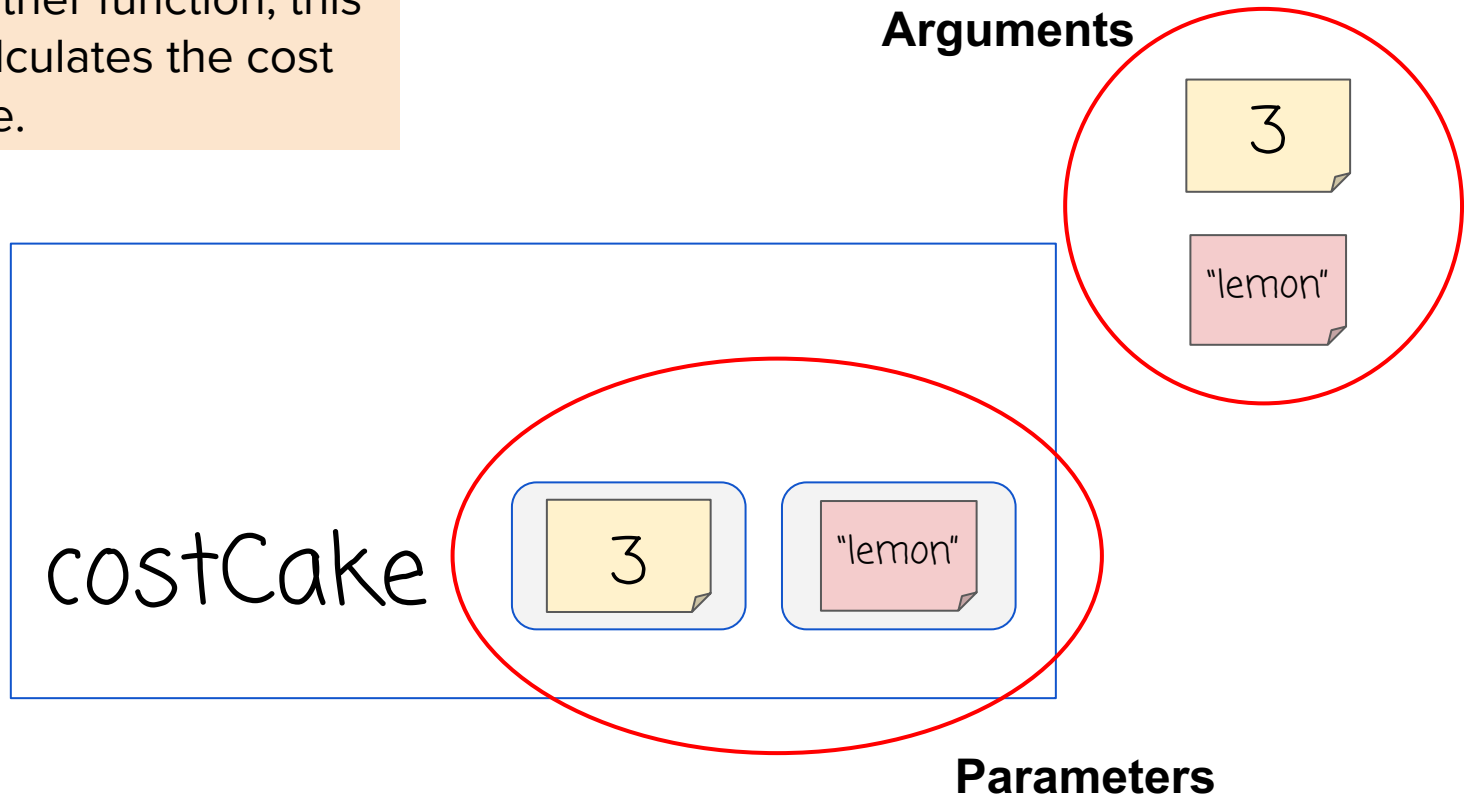
Do This: Time to create your own function with parameters!

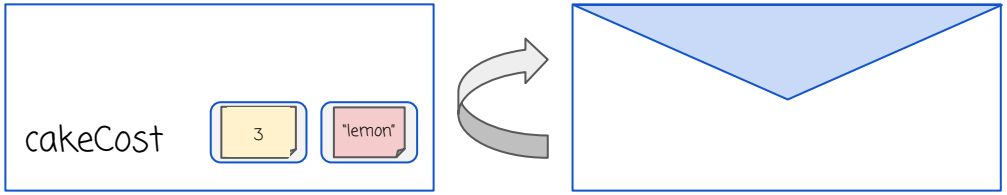
- Complete Challenge #1 in your Activity Guide

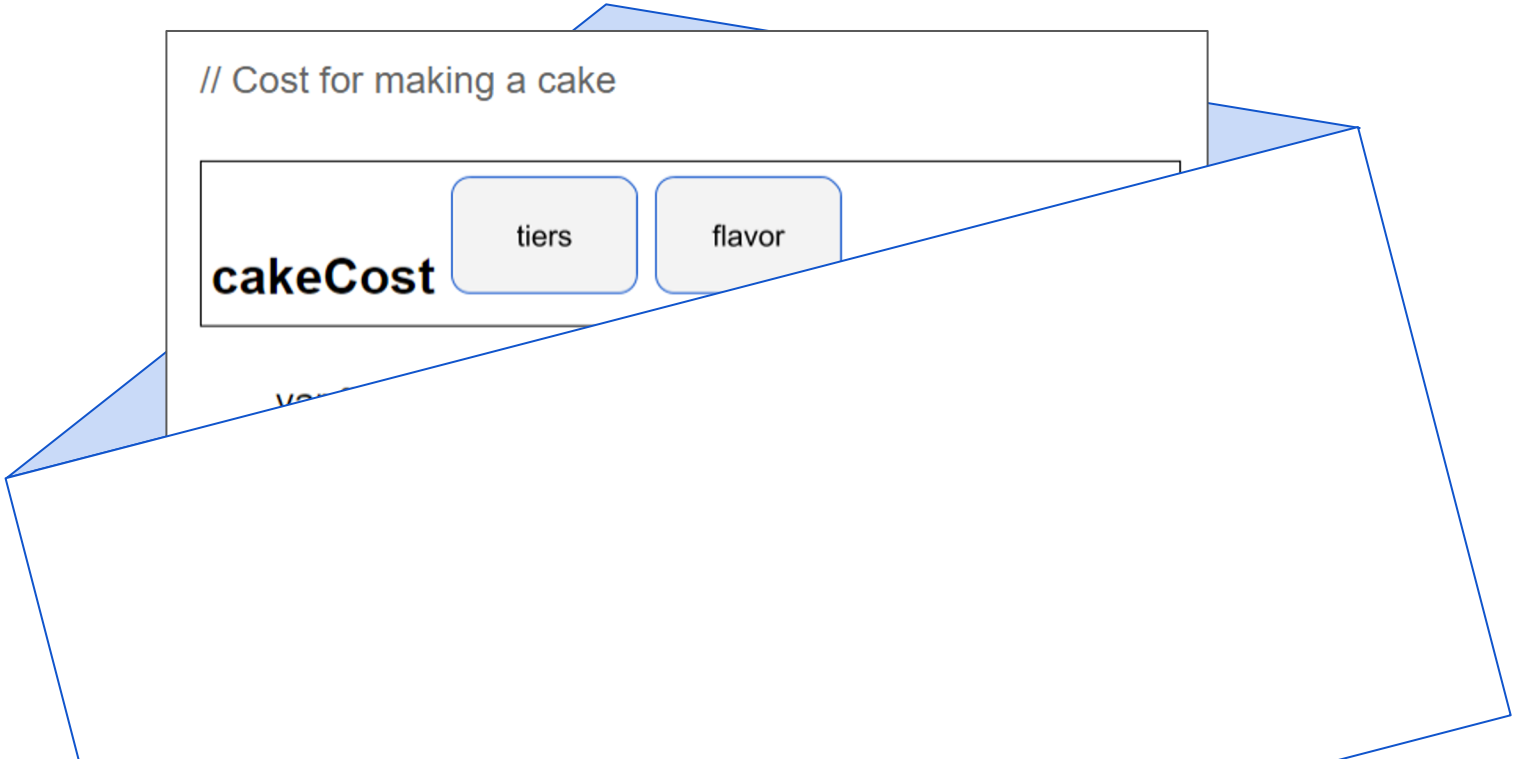


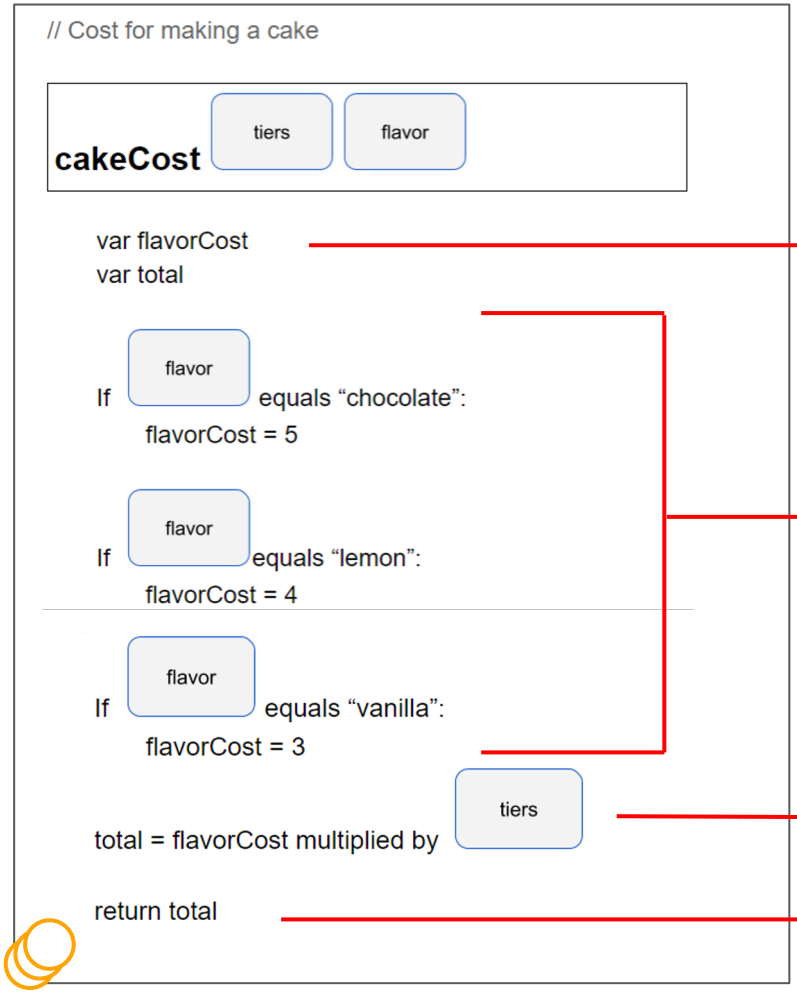


Let's look at another function, this time one that calculates the cost of making a cake.




cakeCost
3
"lemon"


// Cost for making a cake
cakeCost
tiers
flavor



Create two new local variables, **flavorCost** and **total.**

Determine the value of **flavorCost** based on the argument passed through the flavor parameter

Calculate **total** using **flavorCost** and the argument passed through the tiers parameter

???????

```
// Cost for making a cake

cakeCost  3  "lemon"

    var flavorCost
    var total

    If [flavor] equals “chocolate”:
        flavorCost = 5

    If [flavor] equals “lemon”:
        flavorCost = 4

    If [flavor] equals “vanilla”:
        flavorCost = 3

    total = flavorCost multiplied by [tiers]

    return total
```

After running this, what does **flavorCost** equal?

4

What does **total** equal?

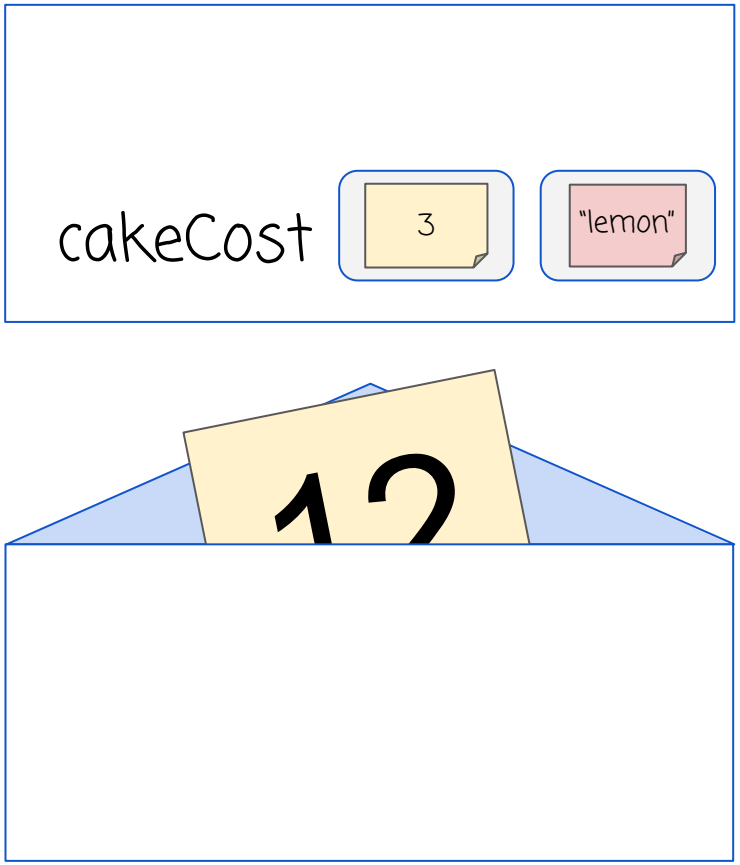
12

```
// Cost for making a cake

cakeCost  3  "lemon"

    var flavorCost
    var total

    If [flavor] equals “chocolate”:
        flavorCost = 5

    If [flavor] equals “lemon”:
        flavorCost = 4

    If [flavor] equals “vanilla”:
        flavorCost = 3

    total = flavorCost multiplied by [tiers]

    return total
```

What does it mean to **return total**?

A return does two things:

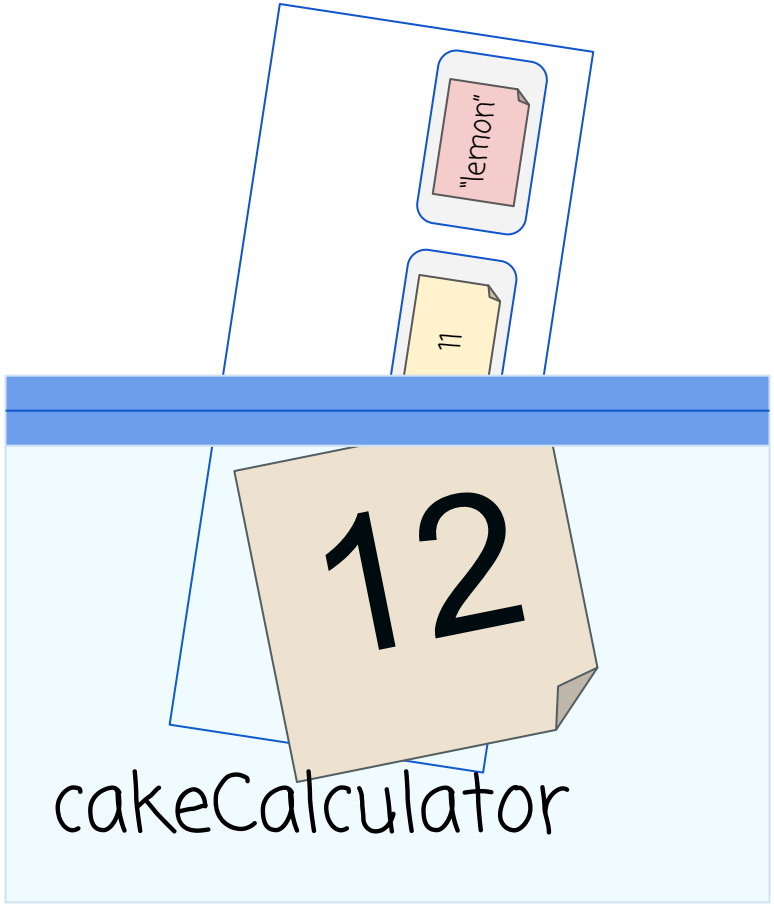
- It stops the flow of the function. If a return is inside of a conditional, if that condition is met the function ends there.
- It **returns** a value to the place where the function was called.



We've called the cakeCost function. It has returned the value **12.**

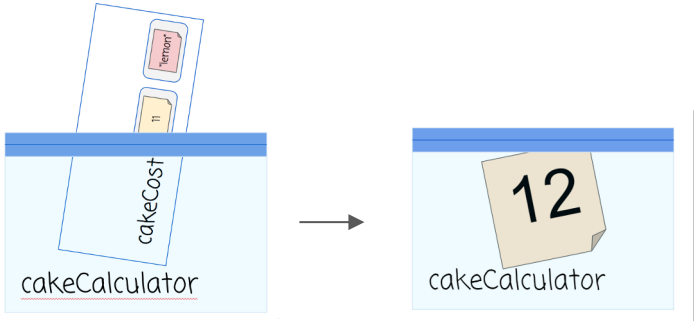
But what happens to that value?

How is it stored?



Let's return to variable baggies!

A function **return** value can be stored in a variable.



Here's how this looks in Javascript:

```
var cakeCalculator = cakeCost(3, "lemon");
```

After the expression is evaluated,
cakeCaculator stores the value **12.**

We can also print to the console like so:

```
console.log(“Cake cost: ” + cakeCost(3, “lemon”));
```

Console

```
Cake cost: 12
```

Wrap Up

Takeaways

- Functions with parameters and return values help us simplify our code
- Functions can only return one value at a time
- A function can have:
    - No parameters and no return values
    - Parameters, but no return values
    - Return values, but no parameters
    - Parameters and return values

Vocabulary

- **Parameter -** a variable in a function definition. Used as a placeholder for values that will be passed through the function.
- **Argument -** the value passed to the parameter
- **Return -** used to return the flow of control to the point where the procedure (also known as a function) was called and to return the value of expression.