Variables

This category starts with just one button. 

But that one button will open up a world of possibilities!

The variables you create with this button are called *global variables*. They can be seen and used by all the procedures in your project. That makes them very powerful.

To show you how to create and use variables, we're going to open an ice cream shop. The program will do the following:

- Ask the customer if they want an ice cream cone and respond appropriately to their answer (either Yes or No).
- Tell the customer they can choose two ice cream flavors.
- Have them choose the first flavor and then a second flavor.
- Tell them what they ordered.
- Thank them and tell them the cost.

Let's learn how to create the code.

You will need several variables to store the information. First, set up a list of flavors. Click the **Create variable…** button and enter **flavors** as its name.

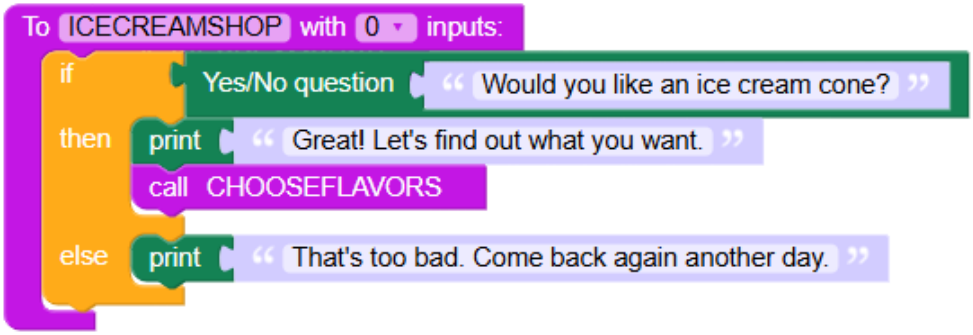
Look at the new blocks in the Variables section:

You can give FLAVORS a value using the ***set*** block.

If FLAVORS were a number, you could make it bigger.

You can use the variable FLAVORS using the last block.

Create a procedure that starts the program. We'll call it ICECREAMSHOP.
It asks a person if they want an ice cream cone. If they do, we start the order process by calling a different procedure, CHOOSEFLAVORS. If they don't, the program ends because there is nothing else for it to do. You don't need to use the ***end program*** block.



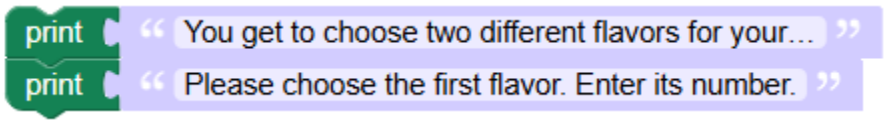
Next, use the ***set*** block to make a list of flavors, like this (choose any flavors you want!):

Each one starts with a number so people don't have to type the words.

Next, we need a CHOOSEFLAVORS procedure. It will ask the customer to choose two flavors. We'll use new variables to store what they enter.

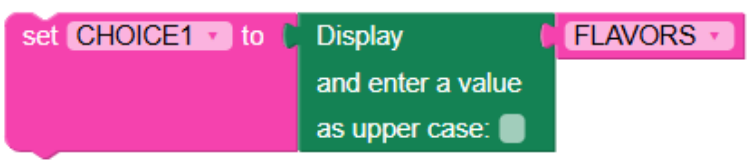
Use the ***Create variable…*** button to make variables called CHOICE1 and CHOICE2.

First, tell the customer what to do:

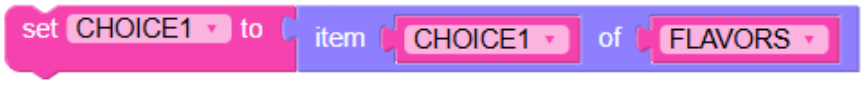


… ice cream cone."

This block displays the list of flavors, stored in the FLAVORS variable, and stores their answer in variable named CHOICE 1.



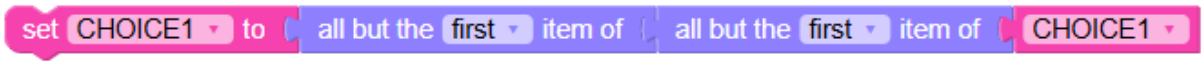
So, when the customer types a number to choose flavor, that number is stored in the CHOICE1 variable. But we need to convert the number to the name of the flavor and not its number. To do this use the ***item*** block, like this:



If they type 2, the value of CHOICE1 changes to become item 2 of FLAVORS. CHOICE1 goes from being 2 to being `2-vanilla`, the second item of the list of FLAVORS.

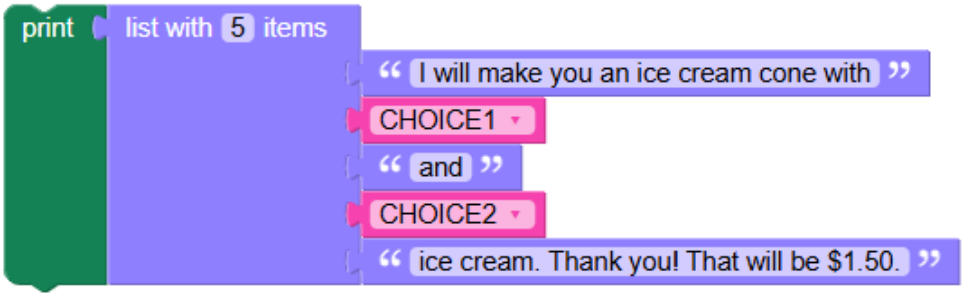
But how do we get rid of the number and hyphen before the name of the flavor? Do you remember the ***all but the first item of*** block? That is just what we need! But we have to use it two times. The first time removes the number, and we're left with `-vanilla`. There is still a hyphen before vanilla, so the second use of the ***all but the first item of*** block removes that.

So, this code turns the value of CHOICE1 from `2-vanilla` into just plain `vanilla`.

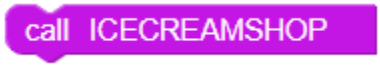


Now, we can use the same process to collect the second flavor choice. Use the same code, but use the CHOICE2 variable you created instead of CHOICE1. That lets us keep the two flavors separate, each with its own variable name.

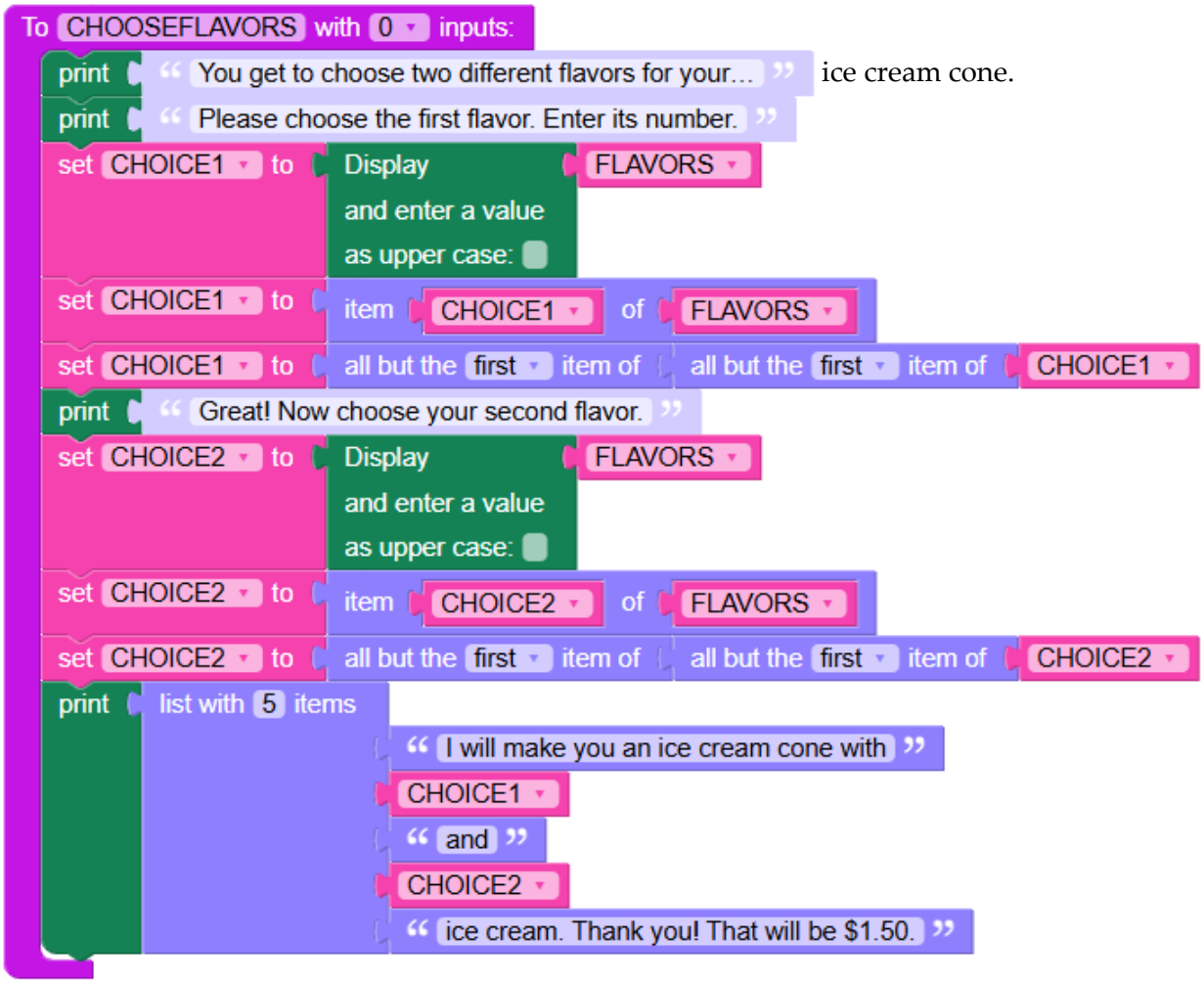
Finally, tell them what you will make for them and how much it will cost.



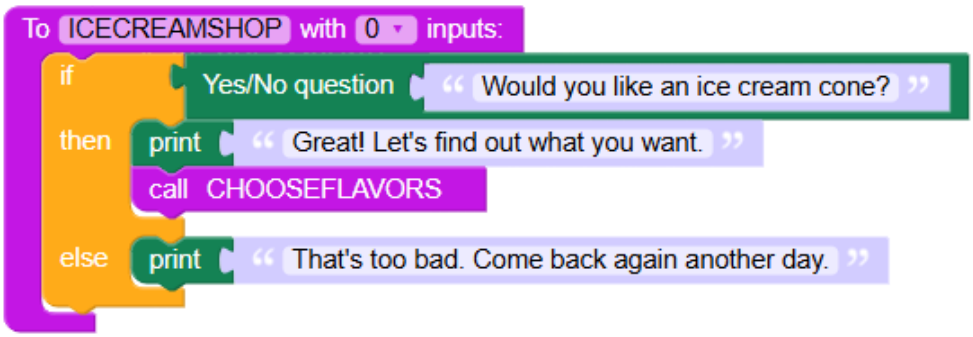
Don't forget to call the main procedure to start the program!

Put this block last in your code. 

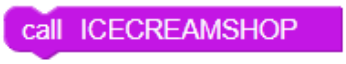
Let's look at the complete program now (on the next page).



ice cream cone.



The ICECREAMSHOP procedure has to come last because it refers to the CHOOSEFLAVORS procedure, which has to be defined or stored as a procedure before it can be used.



The Logo code looks like this:

```
MAKE "FLAVORS (LIST `1-chocolate` `2-vanilla` `3-strawberry` `4-rocky
road` `5-mint chocolate chip`)

TO CHOOSEFLAVORS
  IGNORE ALERT `You get to choose two different flavors for your ice
cream cone.`
  IGNORE ALERT `Please choose the first flavor. Enter its number.`
  MAKE "CHOICE1 READPROMPT :FLAVORS
  MAKE "CHOICE1 ITEM :CHOICE1 :FLAVORS
  MAKE "CHOICE1 BUTFIRST BUTFIRST :CHOICE1
  IGNORE ALERT `Great! Now choose your second flavor.`
  MAKE "CHOICE2 READPROMPT :FLAVORS
  MAKE "CHOICE2 ITEM :CHOICE2 :FLAVORS
  MAKE "CHOICE2 BUTFIRST BUTFIRST :CHOICE2
  IGNORE ALERT (LIST `I will make you an ice cream cone with` :CHOICE1
`and` :CHOICE2 `ice cream. Thank you! That will be $1.50.`)
END

TO ICECREAMSHOP
  IF CONFIRM `Would you like an ice cream cone?` [
    IGNORE ALERT `Great! Let's find out what you want.`
    CHOOSEFLAVORS
  ] [
    IGNORE ALERT `That's too bad. Come back again another day.`
  ]
END

ICECREAMSHOP
```

Modify the program!

What could you add to this program? Try one or more of these ideas!

- Ask the customer if they want the ice cream in a cone or a cup.
- Allow the customer to choose 3 flavors.
- Add more flavors to the list.
- Change the program to offer just one-scoop ice cream cones.

Remember to save your program frequently!

Now you can create any project you can imagine! And send us your code for our Gallery.