

Lesson 5.2: 13 Colonies - Variables, Lists and Initialization

Objectives

In this lesson, students will:

- ❖ Practice assessing the status and next steps of a larger programming project
- ❖ Learn to approach a more complex project with planned tasks and algorithms
- ❖ Practice iteratively building upon smaller programming steps to create a large program
- ❖ Review and practice using lists.
- ❖ Review and practice the concept of initialization code
- ❖ Practice adding comments

Agenda

1. Review and Coding Plan	10 mins
2. Student Activity: Create Needed Variables	5 mins
3. Review Lists	5 mins
4. Student Activity: Initializations	20 mins
5. Class Discussion	10 mins

Resources & Links

- ❑ Lesson 2 starter project (in case students were absent or could not finish the prior lesson (lesson 1)): <https://scratch.mit.edu/projects/320757465>

Preparation

- ❑ Projector for class demonstration and instruction
- ❑ Print student activity worksheet, one per student pair

1. Status and Coding Plan



Display your screen and discuss the status of the program and next steps (using the Colonies Coding Plan).

Colonies Coding Plan:	
Task	Algorithm:
Game Setup: Create the Game Instructions	Done
Game Setup: Create the Colony Labels	Done
Create needed variables: score and the list of colonies	<ol style="list-style-type: none"> 1. Create a variable for <i>score</i> 2. Create a list named <i>colonies</i> to store colony names
Initializations	When the green flag is clicked <ol style="list-style-type: none"> 1. Hide the game instructions sprite 2. Initialize <i>score</i> to 0 3. Initialize the <i>colonies</i> list with numbers 1 to 13 corresponding to the 13 colonies

Let's look at the progress we have made so far and what is next. We wrote the game instructions in a new sprite and we created the labels for the colonies.

Next, we will create and initialize our variables. We need a variable for the score. Since we need to display colony names to be located on the map, we want to keep track of the colonies we have not displayed yet so that we don't show the same colony twice. To do that, we are going to create a list for the colonies.

Prompt students to explain what initialization code is and why we need it.

The **initialization** code sets variables and sprite states to the desired initial value. That way when the program starts we ensure everything starts with the correct values, not left over values from a previous run of the program.

2. Student Activity: Create Needed Variables



Instructions to give to students:

1. If students missed the prior lesson or were not able to complete the coding, they can remix this project which is the state at the end of lesson 1.
<https://scratch.mit.edu/projects/320757465>
2. Open your Colonies project and create a variable for **score**.
3. Create a **list** variable named **colonies** to store colony names.

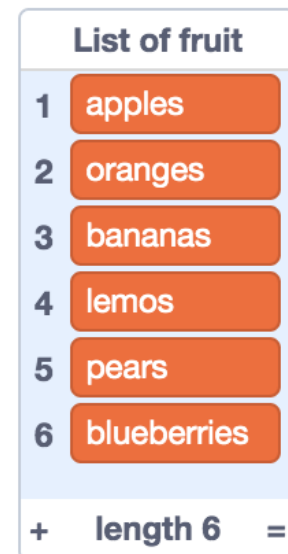
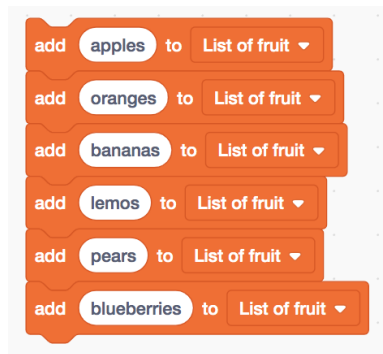
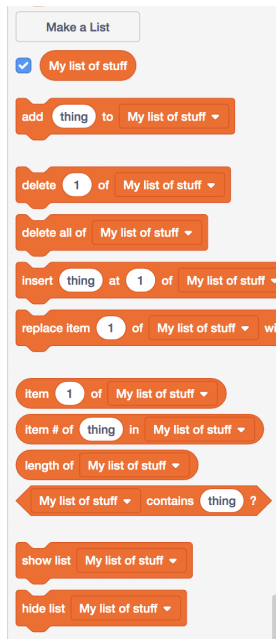
3. Review Lists



Display your screen and engage students in a demonstration and instruction:

A variable stores one value in the computer’s memory. A list stores many values in the computer’s memory. When you create a list, you get a bunch of commands to edit and manipulate your list.

Demo creating the following list or any other of your or student’s choosing. Then demo a few list commands. Exhibit A is useful in displaying and reviewing list commands. Be sure to display the list on the stage so the effect of the list commands are visible.



4. Student Activity: Initializations



Time to initialize our variables.

Ask students which Sprite should contain the initialization code.

Answer: the *Game Instructions* sprite. However, it could be any sprite that contains a When green flag clicked. In this project, the *Game Instructions* sprite makes sense since we want to show the instructions before starting to play the game and the game instructions are applicable to the entire project.

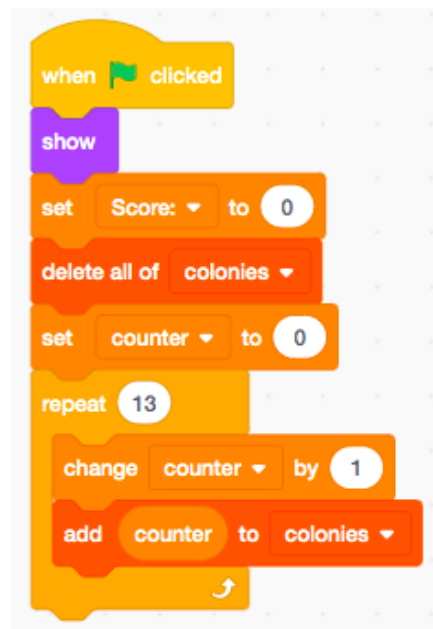
Distribute the activity worksheet and instruct students to follow the activity instructions. Remind them to save and test their projects frequently.

You can use the list of Troubleshooting and Debugging Tips at the end of the lesson to help students during this activity.

Solution to Student Activity:

Review the solution with students.

Prompt students to share their code comments with the class.



```


when green flag clicked
  show
  set Score to 0
  delete all of colonies
  set counter to 0
  repeat 13
    change counter by 1
    add counter to colonies
  
```

3. Wrap Up and Reflections



Reflection Points:

- What was challenging during the initialization activity?
- After creating a variable, you can display it on the stage. How can you hide it?

(Uncheck the check box next to the variable name  in the “Data” section)

- What is a Scratch list?
- Which list commands did you use today?
- What will be the value of counter at the end of this script (25 - 5 is added 4 times to the initial counter value of 5)



```

set counter to 5
repeat 4
  change counter by 5
  
```

Student Activity: Initialization

What to do:

1. Add **initialization** code to your **Game Instructions** Sprite:

When green flag clicked,

- show the game instructions
- set score to 0 (be sure to show the score on the stage)

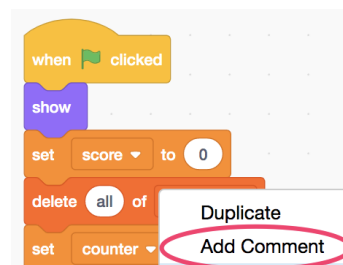
2. Initialize the colonies list with the numbers 1 to 13. To do that we are going to use a helper variable to keep track of the numbers.

- create a variable called **counter**
- after setting score to 0, delete all items from the colonies list (to make sure we start with an empty list)
- set **counter** to 0
- 13 times:
 - add 1 to the value of **counter**
 - add the value of **counter** to the **colonies** list

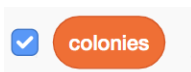
3. Add a comment

Right click on the script to add a comment explaining in a few words what you just did.

Comments are very important so you and others can understand the code.



4. Test your code:



Show the list on your stage so you can easily see the content of the list.

When testing, it helps to display all variables on the stage so you can make sure your code is working. Uncheck them later to hide them.

Your list should contain the numbers 1 to 13 after you run your code.

You can also check the values stored in the variables by clicking on them.

Troubleshooting and Debugging Tips

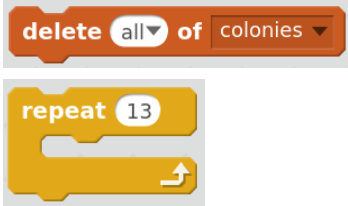

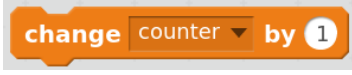


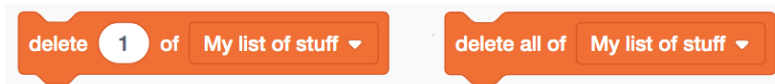
Error	Cause	Solution
The list has more than 13 values	The list was not empty before values were inserted Add delete all of colonies before the repeat 13 loop Or the loop was repeated more than 13 times	
The list has values greater than 13	The counter was not initialised to 0 before the loop	
The list has 13 values but they are all the same	Check if the student used change counter by 1 and not set counter to 1 inside the repeat 13 loop	
The list has value 'counter' 13 times	The student added a text 'counter' instead of the variable counter to the list "colonies"	  Drag the variable "counter" from the data palette to add to "colonies" list

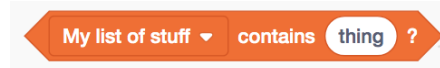
Exhibit A:



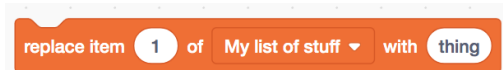
Delete one item from list or all items from the list



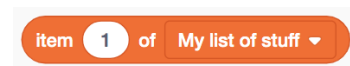
Inserts item at the specified position in the list



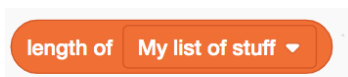
Reports true if the list contains the specified item



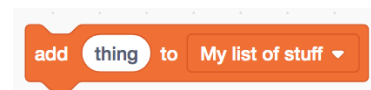
Replaces an item in the list with the specified item



Reports the item at the specified position in the list



Reports how many items are in the list



Adds the specified item to the end of the list