

# Lesson 1: Lists Explore

## Overview

Students will learn the ways that lists are created, accessed, and changed through a teacher-guided activity using plastic baggies and pieces of paper. The lesson begins with a brief reflection on the value of lists. Students then complete the main activity which introduces the syntax to use lists and the ways they can be used. To wrap up students watch two short videos on lists and record the main concepts in their journals.

## Purpose

In the warmup, students brainstorm different lists of information that they encounter on a daily basis. Then in the activity, students return to baggies and sticky notes from the previous unit to build a concrete model of a list before seeing how lists are programmed in Javascript. Students are exposed to different ways of interacting with a list like accessing, removing, appending, and inserting elements. The wrap up concludes the lesson with a summary video and a journal vocabulary exercise.

## Standards

Full Course Alignment

### CSP Conceptual Framework

- ▶ **AAP-1** - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- ▶ **AAP-2** - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

### CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Lesson Modifications

#### Warm Up (5 minutes)

##### Preview Lists

#### Activity (30 minutes)

##### Lists

## Objectives

Students will be able to:

- Use an index to reference specific elements in a list
- Use appropriate vocabulary to describe lists.

## Preparation

- 7+ sandwich baggies per pair of students
- 1 gallon-sized baggy per pair of students
- packs of red and orange stickies
- pens / pencils
- 1 dry erase marker per four students (pairs can share)
- Review the slides and click through all animations

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

For the teachers

- **CSP Unit 5 - Lists Loops, and Traversals** - Slides

For the students

- **Introduction to Lists - Part 1** - Video (**Download**)
- **Introduction to Lists - Part 2** - Video (**Download**)

## Teaching Guide


### Lesson Modifications



**Attention, teachers!** If you are teaching virtually or in a socially-distanced classroom, please read the full lesson plan below, then click **here** to access the modifications.

### Warm Up (5 minutes)

#### Preview Lists

 **Discuss:** *With a partner, brainstorm lists of information that you encounter on a daily basis. Why are lists useful?*

Have students discuss with a partner and then have a few students share out their responses.

#### Discussion Goal:

- Lists help us group together like information
- We can go through a list item by item to check off things we have completed
- Lists help us stay organized


### Activity (30 minutes)

#### Lists

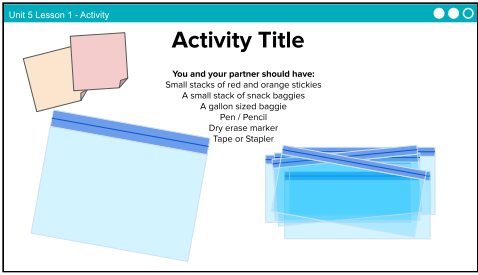
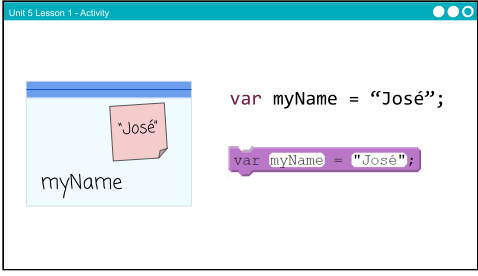
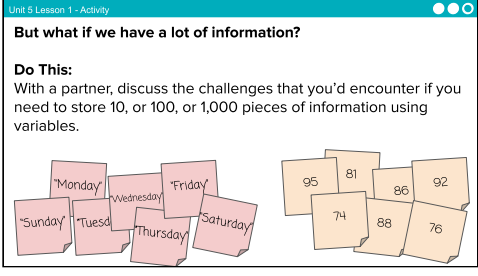
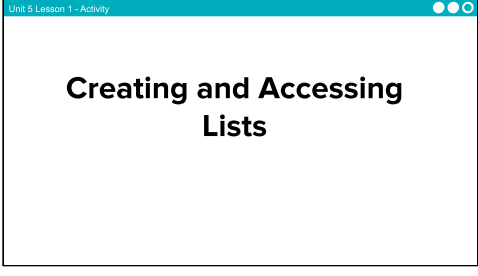
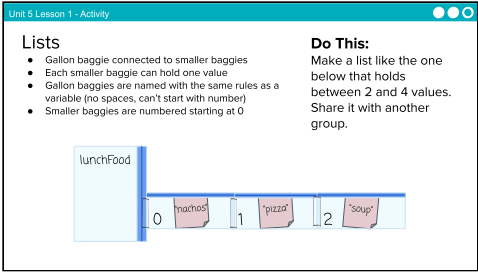
**Group:** Group students in pairs.

**Distribute:** Give each pair of students:

- A small stack of red and orange sticky notes
- A pen/pencil
- 7+ snack-sized plastic baggies
- One gallon-sized plastic baggy
- A dry erase marker to share with another group

 **Display:** Use the activity slides for this lesson to guide the unplugged activity on Conditionals.

Slides	Speaker Notes
--------	---------------

Slides	Speaker Notes
	<p><b>Say:</b> Today, we are going to explore lists.</p> <p><b>Note:</b> Supplies Substitutions</p> <ul style="list-style-type: none"> <li>There's no need to use stickie notes if you have other scraps of colored paper. Also consider cutting stickies in 4 to make them go further. If you don't have dry erase markers handy consider using pieces of masking tape on the baggies.</li> <li>The gallon-sized baggie can be replaced with any baggie that is large enough to hold the snack-sized baggies.</li> </ul>
	<p><b>Say:</b> We've already learned that variables can help us store one piece of information.</p>
	<p><b>Say:</b> But what if we have lots of information?</p> <p><b>Do This:</b> With a partner, discuss the challenges that you'd encounter if you need to store 10, or 100, or 1,000 pieces of information using variables.</p> <p><b>Discuss:</b> Lead a short discussion on the challenges they'd encounter. The main challenges are that you'd need to make a variable for each piece of information and it could get difficult to name them all.</p>
	<p><b>Say:</b> Let's look at how to use lists</p>
	<p><b>Say:</b> Today we're going to practice using a new way to store information called a list. Follow the instructions here to make a list of your own.</p> <p><b>Do This:</b> Have students create their own list. Circulate the room to check on how they're doing. Make sure they share with another group.</p>

## Slides

## Speaker Notes

Unit 5 Lesson 1 - Activity

Your list is made up of **elements**. Each element has its own **index**. Indexes start at 0 and count up. The **length** of the list is how many elements it contains. This list has 3 elements and indexes from 0 to 2.

**Say:** Your list is made up of elements. Each element has its own index. Indexes are just numbers that count up from zero. The length of the list is how many elements it contains. This list has 3 elements and indexes from 0 to 2.

Unit 5 Lesson 1 - Activity

Notice that all the bags can be folded up and be placed inside the big variable baggie. Sometimes we want to think about the whole list, sometimes we want to think about individual elements

**Say:** Notice that all the bags can be folded up and be placed inside the big variable baggie. Sometimes we want to think about the whole list, sometimes we want to think about individual elements

Unit 5 Lesson 1 - Activity

```
var myList = ["pizza", 4]
```

- This command creates a new list and assigns it to a variable
- A list is indicated with square brackets
- Each value in the list is separated by commas

**Do This:** Write out the command that creates the list you just made. Then write the command that makes another group's list. Share both answers with that group.

**Say:** We create a list with this command. It creates a new list and assigns it to a variable. The square brackets indicate we're making a list. Each value is separated by commas.

**Do This:** Have students write the command to make their own list and the list of partner group. By sharing their answers with another group they can check that they're following along.

Unit 5 Lesson 1 - Activity

```
myList[1]
```

- This expression "accesses" the value at that index of the list.
- Also uses square brackets

**Do This:** What do lunchFood[0] and lunchFood[2] access? Compare your answers with another group.

**Say:** If we want to access the values in our list we use the square brackets next to the name of our list.

**Do This:** Have students discuss with other groups. They should see the answers are "nachos" and "soup"

Unit 5 Lesson 1 - Activity

### Lists and Expressions

Replace the list access with a copy of the value it holds. Evaluate the expression as normal

**Do This:** Write three expression that include accesses to the list you created. Use the examples above for inspiration. Trade with another group and have them evaluate your expressions

Examples:

- $3 + \text{myNumbers}[1]$  evaluates to 23
- $\text{myNumbers}[0] + \text{myNumbers}[2]$  evaluates to 35

**Say:** We can use list accesses inside of expressions, just like variables.



**Click for animation**

**Say:** This first expression evaluates to 23 because myNumbers[1] contains 20.



**Click for animation**

**Say:** This second expression evaluates to 35 because myNumbers[0] and myNumbers[2] contain 25 and 10.

**Do This:** Have students write three expressions using their own lists. Then have them trade with another group to practice evaluating. Circulate the room to make sure students are following the directions correctly.

## Slides

## Speaker Notes

Unit 5 Lesson 1 - Activity

**myList[1] = "hello"**  
Assigns the value on the right to the index  
Just like variable assignment, the old value is thrown away and replaced.

```
00 var myNumbers = [10,20,25]
01 myNumbers[0] = 5
02 myNumbers[2] = 30
```

**Do This:** Discuss what the list will contain after line 02 runs.

Index	0	1	2
Initial	10	20	25
After Line 01	5	20	25
After Line 02	5	20	30

**Say:** We can assign the index of a list just like a variable.

**Do This:** Have students discuss what the list will contain after line 02 runs.

**Click for animation**

**Say:** This code will assign a new value at index 2 of the list.

Unit 5 Lesson 1 - Activity

**Do This:**  
Run this program. Compare your result with another group.

```
00 var myStuff = [20, "hat", "pow", 5]
01 myStuff[1] = "cat"
02 myStuff[2] = myStuff[1]
03 myStuff[0] = myStuff[3] + 10
04 myStuff[3] = myStuff[0] + myStuff[0]
```

Index	0	1	2	3
Initial	20	hat	pow	5
After Line 01	20	cat	pow	5
After Line 02	20	cat	cat	5
After Line 03	30	cat	cat	5
After Line 04	30	cat	cat	30

**Do This:** Circulate the room while students run this program

**Click for animation**

**Discuss:** Discuss quickly with the class why the program ends with the list shown.

Unit 5 Lesson 1 - Activity

**Do This:**  
You can use expressions in the place of the list index.  
Run this program and compare your result with another group.

```
00 var myStuff = ["dog", "cat", 3, 10]
01 myStuff[2-1] = "tree"
02 myStuff[myStuff[2]] = myStuff[0]
```

Index	0	1	2	3
Initial	dog	cat	3	10
After Line 01	dog	cat	tree	10
After Line 02	dog	cat	tree	dog

**Do This:** Circulate the room while students run this program

**Click for animation**

**Discuss:** Discuss quickly with the class why the program ends with the list shown.

**Note:** Line 02 is tricky. You need to first evaluate myStuff[0] which is "dog" to determine the value being assigned. Then evaluate myStuff[2], which is 3. At this point, the command reads myStuff[3] = "dog". Now it is just a simple assignment.

Unit 5 Lesson 1 - Activity

## Changing Your List

**Say:** Now let's learn about three different commands that can change your list.

Unit 5 Lesson 1 - Activity

**removeItem(list, item)**  
Removes the item in the list given and at the index given  
All items to the right are shifted over  
The last index is removed from the list

```
00 var myNumbers = [10,20,25]
01 removeItem(myNumbers,1)
02 removeItem(myNumbers,0)
```

**Do This:** Discuss what the list will contain after line 02 runs.

Index	0	1	2
Initial	10	20	25
After Line 01	10	25	
After Line 02	25		

**Say:** The removeItem command will remove an item from a list. The item at the index given is removed, items to the right are shifted over, and the last index is removed.

**Do This:** Have students discuss what the list will show after this command.

**Click for animation**

**Discuss:** Discuss with the class any questions.

Slides	Speaker Notes
<p><b>Unit 5 Lesson 1 - Activity</b></p> <p><b>appendItem(list, item)</b> Adds an item to the end of the list A new index is added to the list to create a place for the item The last index is removed from the list</p> <pre> 00 var myNumbers = [10] 01 appendItem(myNumbers,50) 02 appendItem(myNumbers,100) </pre> <p><b>Do This:</b> Discuss what the list will contain after line 02 runs.</p>	<p><b>Say:</b> The appendItem command will add an item to a list. A new index is added to the end of the list and the new item is place in it.</p> <p><b>Do This:</b> Have students discuss what the list will show after this command.</p> <p> <b>Click for animation</b></p> <p><b>Discuss:</b> Discuss with the class any questions.</p>
<p><b>Unit 5 Lesson 1 - Activity</b></p> <p><b>insertItem(list, index, item)</b> Inserts an item into a list at the index given A new index is added to the list so there's space for the new item The new item is placed at the index given, all other items move right</p> <pre> 00 var nums = [10,50] 01 insertItem(nums,1,20) 02 insertItem(nums,1,100) </pre> <p><b>Do This:</b> Discuss what the list will contain after line 02 runs.</p>	<p><b>Say:</b> The insertItem command will add an item to the middle of a list. The item is placed in the index given. Then every item is moved over one space to make room. A new index is added to the end of the list to make room.</p> <p><b>Do This:</b> Have students discuss what the list will show after this command.</p> <p> <b>Click for animation</b></p> <p><b>Discuss:</b> Discuss with the class any questions.</p>
<p><b>Unit 5 Lesson 1 - Activity</b></p> <p><b>Do This:</b> Run this program. Compare your result with another group. What is the length of the list at the end?</p> <pre> 00 var alist = [20, "hat", "pow"] 01 appendItem(alist, 5) 02 appendItem(alist, 10) 03 removeItem(alist, 1) 04 insertItem(alist, 2, "bang") </pre> <p><b>Command Reference</b> removeItem(list, index) appendItem(list, item) insertItem(list, index, item)</p> <p>The length of the list is 5 (one less than the last index)</p>	<p><b>Do This:</b> Have students run this program. Circulate the room to support them with any questions.</p> <p> <b>Click for animation</b></p> <p><b>Discuss:</b> Discuss with the class any questions.</p>
<p><b>Unit 5 Lesson 1 - Activity</b></p> <p><b>Do This:</b> Run this program. Compare your result with another group. What is the length of the list at the end?</p> <pre> 00 var bList = ["to", 5, "po"] 01 appendItem(bList, bList[2] + "ta" + bList[0]) 02 insertItem(bList, 2-1, "go") 03 deleteItem(bList, 2) </pre> <p><b>Command Reference</b> removeItem(list, index) appendItem(list, item) insertItem(list, index, item) deleteItem(list, index)</p> <p>The length of the list is 4 (one less than the last index)</p>	<p><b>Do This:</b> Have students run this program. Circulate the room to support them with any questions.</p> <p> <b>Click for animation</b></p> <p><b>Discuss:</b> Discuss with the class any questions.</p>

## Wrap Up (10 minutes)

### Video and Vocabulary

**Video:** As a class watch both videos on lists.

**Do This:** Review Key Takeaways and have students add to their journals: lists, element, and index.

---

## Assessment: Check For Understanding

*Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. These questions can be used for an exit ticket.*

**Question:** Match the index with the value. `var myNumbers = [32, 64, 33, 0, 15, 26, 3]`



Check For Understanding