

Lesson 3: Lists Practice

Overview

Practice the basics of lists including creating lists and accessing, inserting, and removing elements from lists.

Purpose

This lesson is students primary opportunity to get hands on with lists in code prior to the Make activity in the following lesson. Give students as much class time as you can to work through these. For this lesson it's recommended that you place students in pairs as a support and to encourage discussion about the challenges or concepts they're seeing.

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)

Activity (35 minutes)

Practice Time

Wrap Up (5 minutes)

Assessment: Check For Understanding: AP Practice

Objectives

Students will be able to:

- Accurately use list operations including accessing, inserting, and removing elements
- Correctly set up a list in a program
- Debug programs with lists

Preparation

- Review the slide on string indexes which you'll cover at the beginning of the lesson
- Review the programming progression to understand what students will be asked to do.

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


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)

 **Do This:** Run a quick vocabulary review with students. Click through the animation to see the vocabulary definitions.

Remarks

Today we're going to have a chance to practice programming with a lot of the concepts and patterns we've explored over the last two lessons. I encourage you to go through these with a partner, but pay close attention to what each other is doing. In our next lesson you're going to have to use a lot of these on an independent project, and these activities are good practice for what you'll find there! Alright, let's get to it!

Activity (35 minutes)


Practice Time

Teaching Tip


Providing Support: Circulate around the room through the lesson encouraging students to use the strategies introduced at the beginning of the lesson. Students have a number of supports at their fingertips, so a big part of your role is helping build their independence in using those resources.

Common Errors: The following are common errors students may encounter in this lesson:

- The last index in a list is at `list.length-1` because the first index is 0. One of the most common errors with lists, therefore, is trying to access indexes that are out of bounds, most notably `list.length`,
- Syntax errors are common with lists, as the bracket notation is new. Encourage students to use the blocks to help if they need.
- When using the patterns students may become overly reliant on simply copying and pasting without thinking through what the code from the patterns does. They may reference variables that don't exist in their program, for example, because it is in the example code of the pattern.

 **Display:** Review the slides on the debugging process and remind students of important skills including the following:

- Try to zoom in on precisely where you're getting stuck.
- Talk to your partner! That's what they're there for!
- Hover over blocks to read the documentation about how they work.
- Read the resources in the Help & Tips tab
- Talk to the group next to you. If another group asks for help make sure to take some time to talk it through with them.

 **Display:** Highlight important debugging skills specifically for lists including ways to use the Watch panel and the Debug Console with lists.

Display: Quickly introduce the fact that strings have indexes too. Students will see `string.length` and `string.substring()` during the lesson so this is just calling out a concept that students will need to recognize when they get to it.

Levels 1-2 Setting Up Lists: These levels involve setting up a list and printing the list to the console

- Level 1: a list of numbers
- Level 2: a list of strings



1-2

Setting up Lists

1

2

Levels 3-5 Accessing Elements in a List: Students practice accessing elements in a list using the index. These levels also practice the random list access and list scrolling pattern students saw in the Investigate lesson.

- Levels 3: Students practice printing specific elements to the console using bracket notation, as in `myList[1]`.
- Level 4: Students build a "Magic 8 Ball" app that uses the Random List access pattern.
- Level 5: Students build a "Class Schedule" app that uses the List Scrolling pattern.

Teaching Tip



Extension Opportunities:

- Level 5: Ask students to add the ability to add new classes to the list in the Class Schedule app.
- Level 9: Ask students to add the ability to remove items in the Food Diary app
- Level 11: Ask students to add a button that will "Undo" adding the most recent song to the list. This is pretty tricky and requires keeping track of some new information.



3-5

Accessing Elements in a List

3

4

5

Levels 6 Strings and Indexes: Students practice accessing elements in a list

- Levels 6: Students practice printing the length and first characters of strings
- Level 7: Students debug code that uses `string.substring()` to grab different parts of a date stored as a string.



6-7

Strings and Indexes

6

7

Levels 8-11 List Operations: In these levels, students work with list operations: `appendItem()`, `removeItem()`, and `insertItem()`

- Level 8: Students practice using `appendItem` to increase the size of a list

- Level 9: Students modify a "Food Diary" app so that the user can add new elements to a list
- Level 10: Students practice using `removeItem` and `insertItem` to modify lists
- Level 11: Students modify a "Top Ten Songs" app so that the user can add new items into the middle of a list while keeping the total length of the list to 10.



8-11

List Operations

8

9

10

11

Wrap Up (5 minutes)

Discussion

Discuss: *What aspects of working with lists do you feel like clicked today? What do you still feel like you have trouble with?*

Have students share with one another before sharing with the whole class.

Goal: Use this opportunity to address any lingering questions or misconceptions in the room. You can also use this as a source of discussion topics to kick off the following lesson. As you lead the discussion, call out the many resources students have access to help when they're getting stuck.

Remarks

Working with lists can be tricky - especially working with the index. We will get more practice tomorrow by making an app that uses lists to store information.

Assessment: Check For Understanding: AP Practice

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: What will be displayed after this code segment is run?

```
gems ← "ruby", "sapphire", "garnet"  
gems ← "opal", "emerald"  
DISPLAY gems[2]
```

Question: What will be displayed after this code segment is run?

```
luckyNumbers ← 15, 33, 25  
INSERT luckyNumbers, 2, 13  
APPEND luckyNumbers, 3  
REMOVE luckyNumbers, 1  
DISPLAY LENGTH luckyNumbers
```

☰ 12-13

Check For Understanding: AP Practice

12 ✓

13 ✓