FIRIA LABS

LESSON: List Practice #1

Time: 45 minutes

Overview:

This lesson will help students develop a shared mental model of lists by going through a couple of whole class examples. Students can use baggies, white boards, or any other "container" you choose. Take your time with this lesson and help students understand the basics of lists.

Objectives:

- I can identify parts of a list – element and index
- I can access elements and manipulate values in a list
- I can define list, element and index

Standards:

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

CSP Framework:

Computational Thinking Practices:

3.A Generalize data sources through variables.

3.B Use abstraction to manage complexity in a program.

4.B Determine the result of code segments.

Key Concepts:

- A **list** can be used when you have, or could have, many similar variables.
- A list has **elements**, or items, organized in order by number.
- The number used to organize the elements is called an **index**.
- Data in a **list** can be changed, removed, inserted, and used in calculations.

Preparation:

Decide on the containers you want students to use when developing a mental model of lists. (white boards, baggies, paper cups, etc.)
Make printouts of examples A and B for students to use as group work

Links:

- No assignment for this lesson; group work
- Lists Vocab slide deck
- Examples A & B for printing

Agenda:

- Warm-up (5 minutes)
- Teacher-led mental model (20 minutes)
- Group practice (20 minutes)

Vocabulary:

- **List:** (from Mission 7) A sequence of items you can access with an index.
  (from Lists Vocab) an ordered collection of elements
- **Index**: (from Mission 7) A number that keeps track of what choice should be displayed.
  (from Lists Vocab) a common method for referencing the elements in a list or string using numbers
- **Element:** an individual value in a list that is assigned a unique index
- **List Length:** how many elements it contains. Lists can grow or shrink as elements are added or removed. You can calculate the current length by using the function: ***len(list_name)***

Assessment:

- Daily reflection form
- Demonstrate in a group to create a list and manipulate its data
- Exit ticket or group review

Teaching Guide

Warm-up / Design Process (5 minutes)

This short warm-up is to introduce abstraction and functions. The assignment document is only used for the debugging table, and then at the end of the lesson, as a review.

💡 **Teaching tip – warm-up**

- In the last lesson (Mission 7) you were introduced to lists and how they can increase the functionality of a program. Today we are going to create a mental model of lists and practice working with list elements.
- What are some examples of lists in your every-day life?
    - Song playlists
    - Grocery or shopping lists
    - To-do lists
    - Phone contacts
- What is a variable? What are some limitations of variables? (can only hold one value at a time)

Teacher-led mental model (20 minutes)

💻 Students will NOT work at computers for this lesson. I recommend groups of three, standing at vertical white boards.

💡 **Teaching tip:**

Students can use small whiteboards or baggies (or other containers) to represent individual variables. I will use baggies as an example here, so modify as needed if the students use something else as their variable containers.

📓 **Preparation:**

Have students work in groups of three at vertical white boards. Each student will need one to three baggies and slips of paper to write on. Students at white boards may need tape to connect their baggies together. Students at magnetic boards will need magnets.

🧑‍🤝‍🧑 **Say: (Example #1)**

- Think about your favorite candy. Write it down on a slip of paper and put it in the baggie.
- These baggies represent individual variables. How many variables does your group need to represent all the favorite candies?
- You probably have more than one favorite candy. Write your other favorite candy on more slips of paper, and put each one in a separate baggie. Remember – each baggie, or variable, can only hold one value!
- If you want to write a program that uses all the favorite candies, how many variables would you need? What if you have more favorites? What if you change your mind and have less? How will these changes affect your code?
- All the variables are related. They could be called candy1, candy2, candy3, etc. Instead of creating all these individual variables, wouldn't it be better to have ONE variable that could hold all the values? SURE! And we have this capability with lists.
- Tape your baggies together in a line to create a list.
- The order you taped your baggies is the order of the list. Now that we have a list, how can you access a single candy? We number them! Computers start counting by 0, so starting with 0, number each of your candies.
- The number of each candy is called an index, and the candy (or value) is called an element.
- How many elements are in your list of candies? Notice that the length of the list (number of elements) is one more than the last index.
- What would you call your list? What should its name be? The programming convention is to use a plural noun for lists. This is a visual clue that you are working with a list instead of a variable.

🧑‍🤝‍🧑 **Say: (Example #2)**

Students should continue in their groups. They can use baggies, white boards, etc. for their individual variables.

- Think of your favorite number. Write it on the white board.
- Now perhaps you have another favorite number. Write it on another white board.
- Think about how many variables you need to represent all these numbers.
- … continue this dialog similar to the first example. End with going over the definitions of list, index and element

🧑‍🤝‍🧑 **Show:**

- The slide deck for Lists Vocab. It includes 4 slides. Go over at least the first three

Group Practice (20 minutes)

💻 Students will continue in their groups of three at the vertical white boards. Print the two examples in advance and have them ready for students to work through.

💡 **Teaching tip:**

Students will have two examples to work through in their groups. They can do them in any order. They should compare their answers with other groups and discuss or re-work examples they do not agree on. Teacher can monitor and support as needed.

Answers to Examples:

- Example #A -- by groups Final alist = [4, 6, 1, 5, 6]
- Example #B -- by groups Final blist = [10, 9, 3, 7, 13, 1]

✅ Students worked in groups during this unplugged lesson and will not have an assignment to turn in. There will be a review assignment for the next lists practice lesson.

Formative Assessment:

- Daily reflection form
- Class discussion on what they learned about lists
- Programming journal (add notes)
- Exit ticket

SUCCESS CRITERIA:

- ❑ Work in a group to complete example A and B
- ❑ Define list, index and element
- ❑ Access a single element in a list
- ❑ Manipulate values in a list by changing an element, adding elements, and removing elements