Traversing List Program

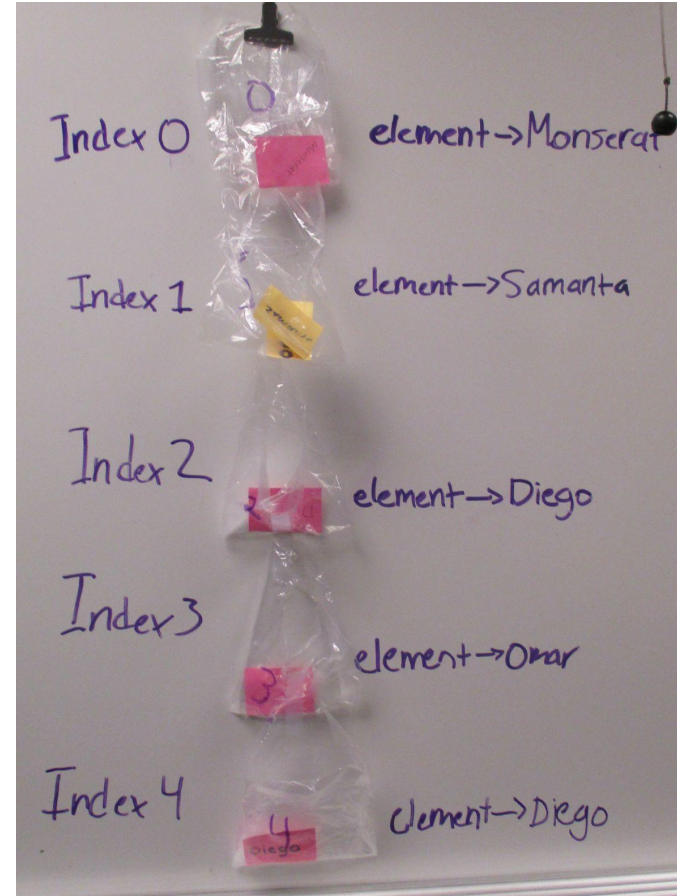An supplemental lesson after Mission 9 and Traversing a List Assignment

Warm-up

Lists

Lists

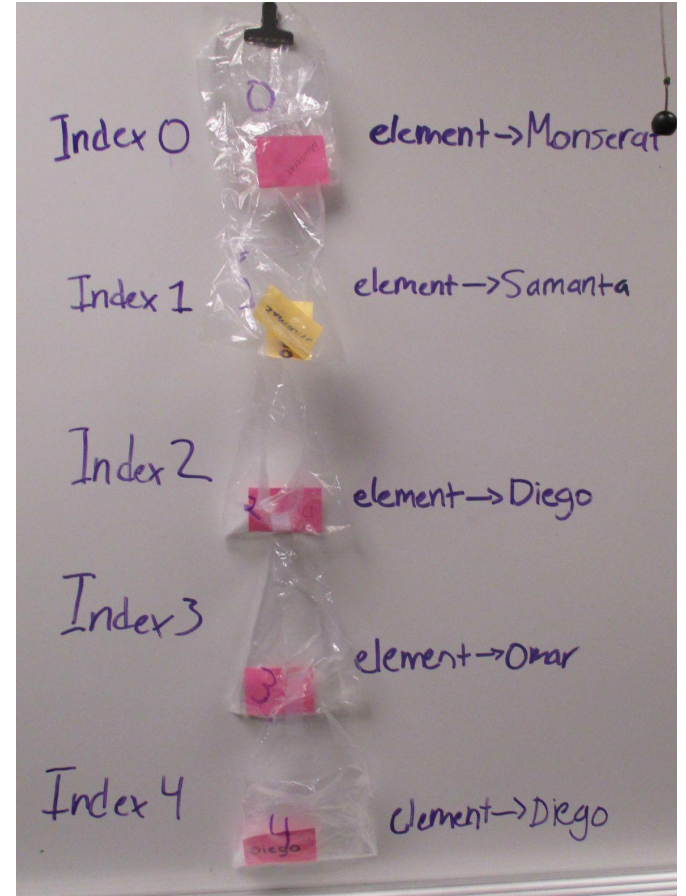In the last lesson, you learned about traversing a list.

What does it mean to traverse a list?

Lists

You also learned how to and practiced traversing a list.

- With a while loop
- With a for loop
- With a specialized for loop

Traversing a List Examples

```python
count = 0
while count < len(my_list):
    display.show(my_list[count])
    sleep(2)
    count = count + 1
```

```python
for index in range(len(my_list)):
    the_image = my_list[index]
    if type(the_image) == tuple:
        display.fill(the_image)
    else:
        display.show(the_image)
    sleep(2)
```

```python
for item in my_list:
    if type(item) == tuple:
        display.fill(item)
    else:
        display.show(item)
    sleep(2)
```

FIRIA LABS

Activity

Part 1: Traversing a list using the simplified for loop

Traversing List Program

For this assignment, you will start a new program. However, you will use the data lists from your Create PT Practice #1

- In CodeSpace, go to Mission 8, objective 6 (or any mission/objective that is not referencing the simulator)
- Browse your files and open your Create PT Practice #1 program (**Practice_PT_1**)
- Start a new project (**Traverse_List**)

Traversing List Program - part 1

You will use some of the same code from
the Create PT Practice program.

- Include a comment block at the
  beginning of your code with your
  names and program description
- Copy the imports and ONE list from
  the **Practice PT** and paste it into the
  Traverse List program

```
'''
Your names
Traversing a list program
'''

from codex import *
from time import sleep
import random

dbacks_pos = ["pitcher", "catcher", "1st Base",
              "2nd Base", "3rd Base", "shortstop"]
```

FIRIA LABS

Traversing List Program - part 1

- Copy the intro() and ending() from the **Practice PT** and paste it into the **Traverse List** program
- Modify the **intro()** to only include an option for **L** and **D**

```python
def intro():
    display.clear()
    display.print("Slideshow program")
    display.print("")
    display.print("L = slideshow")
    display.print("D = quit")

def ending():
    display.clear()
    display.print("Thank You")
    display.print("The End")
```

FIRIA LABS

Traversing List Program - part 1

For part 1, you will use a **specialized for loop**, to display each item in the list. Review the structure of a specialized for loop.

```
for item in my_list:
    if type(item) == tuple:
        display.fill(item)
    else:
        display.show(item)
sleep(2)
```

FIRIA LABS

Traversing List Program - part 1

- This specialized for loop won't need an if statement
- You will only print the item and use a delay

Create a function that will traverse a loop to display each item in the list.

```python
def slideshow():
    for {..........} :
        display.clear()
        display.print( {...} )
        sleep(1)
    display.print("End of list")
```

FIRIA LABS

Traversing List Program - part 1

Write code for the main program that will:

- Call intro()

- Start a while True loop

- If statement for BTN_L that calls slideshow()

- If statement for BTN_D that breaks the loop

- Call ending()

Run the program and make sure it works. You should

see a slideshow of the items when you press BTN_L

FIRIA LABS

Activity

Part 2: Traversing a list using a for loop and two lists

Traversing List Program - part 2

Modify your program by copying the second list to your code

```
dbacks_pos = ["pitcher", "catcher", "1st Base",
              "2nd Base", "3rd Base", "shortstop"]
dbacks_players = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",
                  "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]
```

The simplified for loop works great for a single list. However, it won't work for more than one list. You will need to use a for loop to traverse more than one list and use the index variable to reference the items.

Traversing List Program - part 2

Review the for loop

For loop used with a single list:

```
for index in range(len(my_list)):
    the_image = my_list[index]
    if type(the_image) == tuple:
        display.fill(the_image)
    else:
        display.show(the_image)
sleep(2)
```

For loop used with two lists:

```
for index in range(len(my_pics)):
    display.show(my_pics[index])
    audio.mp3(my_music[index])
    sleep(1)
```

Notice that index is used for both lists

FIRIA LABS

Traversing List Program - part 2

Modify your program by changing the **slideshow()** function so that it prints an item from both lists at the same time in the loop.

Change this specialized for loop to a for loop (with index) and display an item from both lists

```python
def slideshow():
    for item in dbacks_pos:
        display.clear()
        display.print(item)
        sleep(1)
    display.print("End of list")
```

Run the program and make sure it works. You should see a slideshow of the items when you press BTN_L

FIRIA LABS

Activity

Part 3: Adding two team lists and options for BTN_A and BTN_B

Traversing List Program - part 3

Add code to work with either list. From **Practice #1**, copy and paste the other two lists into your **Traversals** program.

```python
dbacks_pos = ["pitcher", "catcher", "1st Base",
              "2nd Base", "3rd Base", "shortstop"]
dbacks_players = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",
                  "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]
rangers_pos = ["cather", "3rd base", "1st base",
               "shortstop", "2nd base", "outfielder"]
rangers_players = ["Mitch Garver", "Josh Jung", "Nathaniel Lowe",
                   "Corey Seager", "Marcus Semian", "Adolis Garcia"]
```

Traversing List Program - part 3

Review your **display_info()** function from **Practice #1**. You added a parameter and used it in an if statement (selection) to assign the lists to display.

You will be doing the same thing to your **Traversal** program, but you won't need the index as a global variable.

```python
def display_info(state):
    global index
    if state == 1:
        the_list1 = dbacks_pos
        the_list2 = dbacks_players
    else:
        the_list1 = rangers_pos
        the_list2 = rangers_players
```

FIRIA LABS

Traversing List Program - part 3

Modify your **slideshow()** function

- Use a parameter (you can call it state, or topic, or choice, etc.)
- Use the parameter in an if statement to assign the lists to two local variables
- Use the two local variables in the for loop to display the data

```
def slideshow():    #Add parameter
    # If statement here
    for index in range(len(dbacks_pos)):
        display.clear()
        display.print(dbacks_pos[index])
        display.print(dbacks_players[index])
        sleep(1)
    display.print("End of list")
```

Change the specific lists to the local variables

FIRIA LABS

Traversing List Program - part 3

Modify your **intro()** function

- Include an option for BTN_A
- Include an option for BTN_B

```python
def intro():
    display.clear()
    display.print("Welcome to the slideshow")
    display.print("")
    display.print("A = Diamondbacks")
    display.print("B = Rangers")
    display.print("L = slideshow")
    display.print("D = quit")
```

FIRIA LABS

Traversing List Program - part 3

Review your **Practice #1** main program and the while True loop.

You programmed an if statement for the button for each topic and set the state.

You will do this to your **Traversal** program.

```python
# Main Program
intro()
index = 0
state = 1

while True:
    display_info(state)

    if buttons.was_pressed(BTN_A):
        state = 1

    if buttons.was_pressed(BTN_B):
        state = 2

    if buttons.was_pressed(BTN_D):
        break

ending()
```

FIRIA LABS

Traversing List Program - part 3

Modify your traversals main program:

- Call the **intro()** and assign **state** (or whatever you are calling the variable)
- You do not need index for this program
- Add if statements for BTN_A and BTN_B
- The slideshow() function is called in the if statement for BTN_L

Run the program and make sure it works. You should see a slideshow of the items when you press BTN_L

FIRIA LABS

Traversing List Program - part 3

Run the program and make sure it works.

- The slideshow should run every time you press L
- The lists being used should change depending on if you press A or B
- The program should end when you press D

FIRIA LABS

Traversing List Program - part 3 challenge

You may have completed the challenge from **Practice #1** – add the team name.

- Add a local variable in slideshow() that is assigned the team name and display it in the for loop with the items from the list.
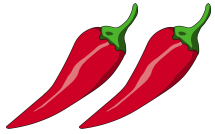
```python
def slideshow(topic):
    if topic == 1:
        the_list1 = dbacks_pos
        the_list2 = dbacks_players
        team = "Diamondbacks"
    else:
        the_list1 = rangers_pos
        the_list2 = rangers_players
        team = "Rangers"
```

Challenges

Two challenges are given, with different levels of difficulty, if you have time to try one (or both).
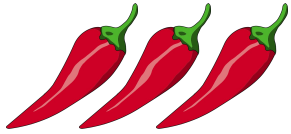
🌶️🌶️ **Medium Challenge**

In CodeSpace, open a program you completed that has a list.

Modify or add code that will traverse the list.

Possible programs

- Mission 7 - Personal Billboard
- Mission 8 - Answer Bot
- Remix #2

FIRIA LABS

🌶️🌶️🌶️ **Spicy Challenge**

Mission 9 is the Game Spinner. It uses a list for the arrows. The list is traversed many times in a loop.

- Modify your code to use a list to traverse the list of arrows inside the loop.
- The easiest way to accomplish this is if you create your own list of arrows. You can use the list to the right.
- Since there is only one list to reference, you can use a simplified for loop.

Copy and paste this list into your code, and use it to traverse the list of arrows:

```
MY_ARROW_LIST = [
    pics.ARROW_N,
    pics.ARROW_NE,
    pics.ARROW_E,
    pics.ARROW_SE,
    pics.ARROW_S,
    pics.ARROW_SW,
    pics.ARROW_W,
    pics.ARROW_NW
]
```

Wrap-up

Traversing a list

Traversing a list

- Now you have had some experience writing code to traverse a list:
  - Specialized for loop
  - For loop
  - Using a parameter in an if statement
- Open your assignment document and answer the questions in the reflection.