AP CSP Python with CodeX **Traversing a List #2 Activity Guide**	**Name:**
Introduction	
During this assignment, you will modify two CodeX programs to use a for loop.	
Warm-Up	
What do you remember about a for loop?	Answers will vary. They could include: ● A for loop simplifies a while loop. ● A for loop builds-in the loop control variable. ● A for loop automatically increments the control variable. ● A for loop is shorter than a while loop. ● A for loop is a control structure. ● A for loop is a form of iteration.
What do you remember about traversing a list?	Answers will vary. They could include: ● Traversing a list means accessing each element one at a time in order. ● The easiest way to traverse a list is using a for loop. ● A specialized for loop eliminates the need for an index variable.
Examples and Challenges	
For loop example. Use this space to take notes about the for loop example.	Notes as needed. The example uses the Answer_Bot program and changes the pixel function to a for loop. It compares a “before” and “after”.
Slides 7 & 8 review information about traversing a list. Use this space to take notes.	Notes as needed. Information from the slides: ● Traversing means to travel, or **traverse**, through a list one element at a time in order. ● To traverse a list you use a for loop to start with the first index (0) and access that item. ● Continue accessing each item in order until the last index (length of list - 1) is reached. Traversing a list can be very useful in many ways. ● Display all items in a list, one at a time, like a slideshow. ● Looking through a list to see if a particular value is an item in the list. ● Create a sub-list from the complete list, like all numbers less than 10.
Use this space to take notes about the code for example #4. During this project you traverse two lists.	Notes as needed. The program makes significant modifications to Pixels1 by adding a list for the colors and a list for the brightness. It also uses a for loop in the pixels function.

Slide 16 reviews abstraction. Explain procedural abstraction and data abstraction.	Answers will vary. Information from the slides: ● The turn_pixel() function with parameters is procedural abstraction. ● It allows you to use the same function for any color and brightness without duplicating code, and you can call the function as many times as you want. ● The lists are data abstraction. ● The lists allow you to easily modify the data (add, move, change, etc.) and you will not have to change the code at all!
Use this space to take notes about the code for example #5. During this project you traverse a 2D list.	Notes as needed. The program changes the Pixels1 program by combining the two lists into one list of lists, or a matrix.
Challenge #1 Were you able to complete the challenge? If so, which method did you use: a third list or a matrix?	Students may or may not get to this point. It is optional. Students either add a third list to Pixels1_traversals or add a third item to the list of lists to Pixesl1_matrix. The added data is for the delay.
Challenge #2 Were you able to complete the challenge? If so, explain which program you used and the changes you made.	Students may or may not get to this point. It is optional. Each answer will vary, depending on the project picked and the changes made.
Wrap-Up	
Explain how to traverse a list.	Answers will vary. A list can be traversed using a for loop or specialized for loop. Each item is accessed using an index.
Explain what a matrix is.	Answers will vary. A simple answer is that a matrix is a list of lists.
During this lesson you modified Answer_Bot_traversals and completed Pixels1_traversals and Pixels1_matrix, plus any challenges. Submit your modified programs to the teacher.	