AP CSP Python with CodeX **Traversing a List #3 Activity Guide**	**Name:**
Introduction	
During this assignment, you will create three new CodeX programs that traverse a list.	
Warm-Up	
Discuss how you create a list and access an element in the list.	Answers will vary. Answers can include: • A list is created using square brackets. • The elements in a list are separated with commas. • Access an element by using its index.
Discuss how you traverse a list.	Answers will vary. Answers can include: • Traverse a list using a for loop. • You can avoid the need for an index variable by using a specialized for loop.
Discuss the structure of a for loop.	Answers will vary. Answers can include: • A for loop uses a control variable and has a range. • A for loop has a colon (:) at the end, indicating a block. • Example: for i in range(10):
Examples	
Use this space to take notes about the code for example #6. The program will traverse a list to see if a particular item is in the list.	Notes as needed. The program is Class_Schedule. Students type their own schedule and then see if a specified course is in their list. This program introduces the **if..in** structure, which they will see on the AP exam. This program also introduces using the console panel for input.
Use this space to take notes about the code for example #7. This program will traverse a list to filter it, creating a sub-list.	Notes as needed. The program uses the append() method to create a list. • It will filter the list to create a sub-list. • It adds parameters to the functions. • It uses the str() function when printing integers.
Use this space to take notes about the code for example #8. This program will traverse a list to filter it, by checking to see if a value from one list is in a second list.	Notes as needed. This example combines the first two examples and uses both techniques. Two lists are created first, and then a filtered sub-list is created when traversing the two lists.
Wrap-Up	
Discuss several ways to traverse a list.	Answers will vary. Notes from the slides:

	• Traverse a single list using a specialized for loop. • Traverse multiple lists using a for loop and index. • Repeatedly traverse a list using a for loop. • Traverse a list by checking if a value is in a list. • Traverse a list to filter it.
Discuss several reasons to traverse a list.	Answers will vary. Notes from the slides: • Display all items in a list, one at a time, like a slideshow. • Access all the items in a list, one at a time, like using the data to turn on pixels. • 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 farm animals.
Reflect on your programming experience during this unit. What are some things you learned about programming in general?	Answers will vary. Some notes from the slides are included, but any insight a student has is worthy. • Programs can be different, and each programmer is different. There are many ways to write a program to complete a task. • Sometimes a function needs a parameter, and sometimes it doesn't. They can be written differently and still accomplish the same task. • Variable and function names can be different, but should always be descriptive. • You can use the Console Panel to give input as well as receive output.
During this lesson you created three programs: Class_Schedule, Number_Sorter and Animal_Sorter. Submit your programs to the teacher.	