Containers Part 1

Python has a wide variety of built-in types for storing anything from numbers and text (e.g., `int`, `float`, `str`) to common data structures (e.g., `list`, `tuple`).

Unit 1 Lists

A variable can hold multiple values in the form of a ***list***. The values are separated by commas and wrapped in square brackets. For example:

```
primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
```

Each ***element*** of the list can be referenced by an ***index***, which is the value's sequential position starting at 0. For example, `primes[4]` is 11.

index	0	1	2	3	4	5	6	7	8	9
value	2	3	5	7	11	13	17	19	23	29

Questions

1. What is the index of the second element of `primes`? What is the value at that index?

2. How does the index number compare to the position of the element?

3. Type each line of code in a Python Shell and write the corresponding output. If an error occurs, write the type of error.

4. How did you reference the value of the 3rd element of `odd`?

5. What did the output of the len() function tell you about the list?

Python code	Shell output
`odd = [1, 3, 5, 7]`	
`odd`	
`odd[2]`	
`odd[4]`	
`len(odd)`	
`number = odd[1]`	
`number`	
`odd[1] = 2`	
`odd`	
`number`	

6. One of the lines in #3 displayed an error. Explain the reason for the error.

7. Write a statement that assigns a list of three integers to the variable `run`.

8. Write a statement that assigns the value 100 to the last element of `run`.

9. Write a statement that assigns the first value of `run` to a variable named `first`.

Unit 2 Sequences

Lists and strings are examples of ***sequence*** types. Consider the following lines that were entered into a Python Shell. Write an asterisk (*) next to any row your team has questions about.

Python code	Shell output
`seq1 = "one two"`	
`type(seq1)`	<class 'str'>
`len(seq1)`	7
`seq1[1]`	'n'
`seq1[1] = '1'`	TypeError: 'str' object does not support item assignment
`seq2 = ["one", "two"]`	
`type(seq2)`	<class 'list'>
`seq2[1]`	'two'
`seq2[1] = 1`	
`print(seq2)`	['one', 1]
`seq3 = ("one", "two")`	
`type(seq3)`	<class 'tuple'>
`len(seq3)`	2
`seq3[1]`	'two'
`seq3[1] = '1'`	TypeError: 'tuple' object does not support item assignment
`seq4 = "one", 1`	
`type(seq4)`	<class 'tuple'>
`seq4`	('one', 1)

Questions

10. What are the names of the three sequence types introduced in Unit 2?

11. How does the syntax of creating a tuple differ from creating a list?

12. Is there more than one way (syntax) to create a tuple? Justify your answer.

13. Which sequence types allow their elements to be changed? Which do not?

14. Is it possible to store values of different types in a sequence? If yes, give an example from the table; if no, explain why not.

15. Summarize the difference between lists and tuples. How do they look differently, and how do they work differently?

16. Enter the following lines in a Python shell and write the output. What do you learn about converting strings and lists?

```
letters = list("Hello")
letters
str(letters)
```

17. Compare the above example with the following when these lines are entered in the Python shell. What is the difference?

```
other_letters  =  "Bye!"
other_letters
```