Lists and Strings

Many interesting problems involve manipulating sequences of data. You've learned about lists and strings before, but this activity provides a more in-depth look at what they can do.

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Name four methods that lists provide, and describe what each method does.
- Explain the syntax and meaning of slice operations, with and without indexes.
- Name four methods that strings provide, and describe what each method does.

Process Skill Goals

During the activity, students should make progress toward:

- Gaining insight about data structures from many examples. (Information Processing)

Model 1 Working with Lists

Recall that a variable can hold multiple values in the form of a list. The values are separated by commas and wrapped in square brackets.

Lists have ***methods*** (built-in functions) that can be called using dot notation. For example, to add a new element to the end of a list, we can use the `append` method.

Python code	Shell output
`rolls = [4, 6, 6, 2, 6]`	
`len(rolls)`	
`print(rolls[5])`	
`rolls.append(1)`	
`print(rolls)`	
`print(rolls[5])`	
`lucky.append(1)`	
`lucky = []`	
`print(lucky[0])`	
`lucky.append(5)`	
`print(lucky)`	
`print(lucky[0])`	
`rolls.count(6)`	
`rolls.remove(6)`	
`print(rolls)`	
`help(rolls.remove)`	
`help(rolls)`	

Questions (15 min) Start time:

1. What is the result of calling the `append` method on a list?

2. What must be defined prior to using a method like `append`?

3. Explain why two lines in Model 1 caused an `IndexError`.

4. What is the result of calling the `remove` method on a list?

5. Based on the `help` output, name several list methods not shown in Model 1. Do not include methods that begin and end with two underscores (e.g., __add__).

6. Give one example of a list method that requires an argument and one that does not.

7. Describe the similarities and differences between using a list method like `append` and Python built-in functions like `print`.

8. Complete the program below by adding two lines; one will initialize a variable called numbers with an empty list, and the other will build the list by adding one number at a time to the end of numbers. This uses a "while" loop which terminates when the user inputs the number 0.

```
x = 1

while x != 0:
    x = int(input("Enter the next number: "))


print numbers
```

Model 2 Indexing and Slicing

A string is a sequence of characters in single quotes (') or double quotes ("). Depending on the application, we can treat a string as a single value (e.g., `dna`), or we can access individual characters using square brackets (e.g., `dna[0]`). We can also use ***slice notation*** (e.g., `dna[4:8]`) to refer to a range of characters. In fact, all types of sequences (including `list` and `tuple`) support indexing and slicing.

Python code	Shell output
`dna = 'CTGACGACTT'`	
`dna[5]`	
`dna[10]`	
`len(dna)`	
`dna[:5]`	
`dna[5:]`	
`dna[5:10]`	
`triplet = dna[2:5]`	
`print(triplet)`	
`dna[-5]`	
`dna[-10]`	
`dna[:-5]`	
`dna[-5:]`	
`triplet = dna[-4:-1]`	
`print(triplet)`	

Questions (15 min) Start time:

9. What is the *positive* index of each character in the `dna` string? Check your answers above.

Character:	C	T	G	A	C	G	A	C	T	T
Index:										

10. What is the *negative* index of each character in the `dna` string? Check your answers above.

Character:	C	T	G	A	C	G	A	C	T	T
Index:										

11. Based on the previous questions, what are `dna[2]` and `dna[-2]`? Explain your answers.

12. Explain the IndexError you observed. What is the range of indexes for the `dna` string?

13. Consider the notation of the operator `[m:n]` for slicing the string.

a) Is the value at the start of the resulting string the same as the value at index `m` (i.e., `dna[m]`)? If not, describe what it is.

b) Is the value at the end of the resulting string the same as the value at index `n` (i.e., `dna[n]`)? If not, describe what it is.

c) Explain what it means when only a single number is referenced when creating a slice, such as `[m:]` or `[:n]`.

14. What is the simplest way to get the first three characters of `dna`? What is the simplest way to get the last three characters?

15. Write a Python expression that slices `'GACT'` from `dna` using positive indexes. Then write another expression that slices the same string using negative indexes.

16. Write a Python assignment statement that uses the `len` function to assign the last letter of `dna` to the variable `last`.

17. Write a Python assignment statement that uses a negative index to assign the last letter of `dna` to the variable `last`.

Model 3 Common String Methods

Like lists, strings have ***methods*** (built-in functions) that can be called using dot notation. See https://docs.python.org/3/library/stdtypes.html#string-methods for more details.

Python code	Shell output
`dna = 'CTGACGACTT'`	
`dna.lower()`	
`print(dna)`	
`lowercase = dna.lower()`	
`print(lowercase)`	
`dnalist = list(dna)`	
`print(dnalist)`	
`dnalist.reverse()`	
`print(dnalist)`	
`type(dna)`	
`dna = dna.split('A')`	
`print(dna)`	
`type(dna)`	
`dna.replace('C', 'g')`	
`print(dna[0])`	
`type(dna[0])`	
`dna[0].replace('C', 'g')`	
`print(dna)`	

Questions (15 min) Start time:

18. Does the `lower` method change the contents of the `dna` string? Justify your answer.

19. Describe the `list` function—what does `list(dna)` return in Model 3?

20. Why is it possible to call the `replace` method on `dna[0]` but not `dna`?

21. Name several other string methods not shown in Model 3. (Read the documentation.)

22. Consider the application of a method on a variable:

a) Does a string variable change after applying a method? Provide justification.

b) Does a list variable change after applying a method? Provide justification.

c) Identify the data type that is ***immutable*** (i.e., the value never changes).

23. Write a single statement to change the final contents of `dna` to `['CTG', 'cc', 'CTT']`. Confirm that your code works in a Python Shell.

24. Why do you think Python has a `replace` method for strings but not for lists?