Introduction to Programming

Spring 2022

Objects and Graphics

- **The String Data Type**
- **Simple String Processing**
- **List as Sequences**
- String Representation and Message Encoding
- String Methods
- Lists Have Methods too
- From Encoding to Encryption
- Input/Output as String Manipulation
- File Processing

The String Data Type

- Text is represented in programs by the *string* data type.
- A string is a sequence of characters enclosed within quotation marks (") or apostrophes (').
- Example

The String Data Type

- We can access the individual characters in a string through indexing.
- The positions in a string are numbered from the left, starting with 0.
- The general form is `string[<expr>]`, where the value of `expr` determines which character is selected from the string.
- In a string of n characters, the last character is at position n-1 since we start counting with 0.
- We can index from the right side using negative indexes.

The String Data Type

- Indexing returns a string containing a single character from a larger string.
- We can also access a contiguous sequence of characters, called a substring, through a process called slicing.
- Slicing:
  - `string[<start>:<end>]`
- start and end should both be ints
- The slice contains the substring beginning at position start and runs up to but doesn't include the position end.

The String Data Type

- If either expression is missing, then the start or the end of the string are used.

The String Data Type

- Repetition builds up a string by multiple concatenations of a string with itself (*)
- Example
  - `3 * "spam"`
  - `"spam" * 5`
  - `(3 * "spam") + ("eggs" * 5)`

The String Data Type

- `len` function will return the length of the string
- `len('spam')`
- We can also use it as a sequence in a for loop:

```
for ch in 'Spam!':
    print (ch, end= " ")
```

Simple String Processing

Operator	Meaning
+	Concatenation
*	Repetition
string[]	Indexing
string[:]	Slicing
len(string)	Length
for var in string	Iteration through characters

Lists as Sequences

- It turns out that strings are really a special kind of sequence, so these operations also apply to sequences!

Operator	Meaning
+	Concatenation
*	Repetition
string[]	Indexing
string[:]	Slicing
len(string)	Length
for var in string	Iteration through characters

Lists as Sequences

- Strings are always sequences of characters, but lists can be sequences of arbitrary values.
  - Lists can have numbers, strings, or both!
- Lists are mutable, meaning they can be changed. Strings can not be changed.

Class Work 05

- Given the initial statement:
- `s1 = "spam"`
- `s2 = "ni!"`
- Show the result of evaluating each of the following string expressions

1) `s1 * 3 + 2 * s2`
2) `s1[1]`
3) `s1 + s2[-1]`

Class Work 05

- Show the output that would be generated by each of the following program fragments:

1)
```
for ch in "aardvark":
    print (ch)
```

- A CS professor gives 5 point quizzes that are graded on the scale 5-A, 4-B, 3-C, 2-D, 1-F, 0-F. Write a program that accepts quiz score as an input and prints out the corresponding grade.