CompSci 101
Collections and Strings

Reminders

• KISS
• "Good programmers are simply good designers."
  • -Dr. Washington
• Design first and always!

Collection Data Type

• Collection of books, toys, shoes
  • Direct access to each item
• Comprised of smaller pieces
  • Strings and lists
• Strings
  • Smaller strings of size one char
  • Empty string "" or '
• Operations on strings
  • $\rightarrow$ concatenation
  • $\ast \rightarrow$ repetition

Indexing a String

$\text{string\_name}[\text{index}]$

• string\_name-your variable name
• index-character element directly accessing
  • Leftmost 0 to string\_length-1
• What about string\_name[-1]?
  • **Whitespaces in a string count**
Slicing Strings

- Slicing bread, tomatoes, etc.
- Substring (smaller part) of the larger string

```python
string_name[n:m]
```

$n$-index of the first character in the substring
$m$-index of the character that immediately follows the last character in the substring

Comparing Strings

- Compares strings to determine the relationship between them
  - ==, >, <, >=, <=, !=

```python
string1 == string2
```

**need to output this or store the result**

in and not in operators

- Is string1 a substring of string2?

```python
string1 in string2
```

string can be a variable or a string literal (e.g., "This is literally an example of a string literal.")
CompSci 101
Lists

List
• Groceries, errands, names, etc.
• Collection of data values
• Sequential
• Directly access each element
• Elements don't have to be the same type
  \texttt{list\_name=[item1, item2, \ldots item6]}
**only top-level items in list**

List access and length
• Similar to strings
  \texttt{list\_name[index]}
• list\_name-your variable name
• index-character element directly accessing
  • leftmost 0 to list\_length-1
• What about list\_name[-1]?

Slicing Lists
• Sublist (smaller part) of the larger list
  \texttt{list\_name[n:m]}
  \(n\)-index of the first character in the sublist
  \(m\)-index of the character that immediately follows the last character in the sublist
in and not in operators

• Is list1 a member of list2?

```python
list1 in list2
list1 not in list2
```