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
  - + →concatenation
  - * →repetition

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # concatenate two strings
    result = result1 + result2
    print(result)
```

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # repeat a string
    result = result1 * 3
    print(result)
```

Indexing a String

string_name[*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**

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # get lengths of strings
    print(len(result1))
    print(len(result2))
```

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # get lengths of strings
    print(len(result1))
    print(len(result2))

    print(result1[0])
    print(result2[5])
    print(result1[-1])
    print(result2[-3])
```

Slicing Strings

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

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

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # slice strings
    print(result1[2:5])
    print(result2[4:8])
```

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # slice strings
    print(result1[:5])
    print(result2[4:])
```

Comparing Strings

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

- *string1* == *string2*

need to output this or store the result

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # compare strings
    print(result1 == result2)
    print(result1 != result2)
    print(result1 > result2)
    print(result1 < result2)
```

in and *not in* operators

- Is string1 a substring of string2?

string1 in *string2*

string can be a variable or a string literal (e.g., “This is literally an example of a string literal.”)

```
if __name__ == '__main__':
    result1 = "Hey there!"
    result2 = "How are you?"

    # check in/not in tests
    print(result1 in result2)
    print(result1 not in result2)
    print(result1 in result1)

    print("Hey" in "Hey Ya!")
    print("" in "Hey Ya!")
    print("Hey Ya!" not in "Hey Ya!")
```

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

list_name=[*item1*, *item2, …item6*]

****only top-level items in list****

```
if __name__ == '__main__':
    ages = [12, 44, 10, 21]
    names = ["Kim", "Janay", "TJ", "Nia"]
    combo = ["Tim", 13, "Ashanti", [40, "Pink"]]

    # output lists
    print(ages)
    print(names)
    print(combo)
```

List access and length

- Similar to strings

list_name[*index*]

- list_name-your variable name
- index-character element directly accessing
  - leftmost 0 to list_length-1

- What about list_name[-1]?

```
if __name__ == '__main__':
    ages = [12, 44, 10, 21]
    names = ["Kim", "Janay", "TJ", "Nia"]
    combo = ["Tim", 13, "Ashanti", [40, "Pink"]]

    # print list length
    print(len(ages))
    print(len(names))
    print(len(combo))

    # directly access elements
    print(ages[1])
    print(names[3])
    print(combo[-1])
```

Slicing Lists

- Sublist (smaller part) of the larger list

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

```
if __name__ == '__main__':
    ages = [12, 44, 10, 21]
    names = ["Kim", "Janay", "TJ", "Nia"]
    combo = ["Tim", 13, "Ashanti", [40, "Pink"]]

    # slice lists
    print(ages[1:3])
    print(names[:2])
    print(combo[1:])
```

in and *not in* operators

- Is list1 a member of list2?

list1 in *list2*

list1 not in *list2*

```
if __name__ == '__main__':
    ages = [12, 44, 10, 21]
    names = ["Kim", "Janay", "TJ", "Nia"]
    combo = ["Tim", 13, "Ashanti", [40, "Pink"]]

    # check membership
    print(21 in ages)
    print("13" not in combo)
    print("Pink" in combo)
```