Sierra Bettis
401-HasTools
September 15, 2021

Cheat Sheet 3

Lists in Python: stores values or objects in python that you create that can be changed and stored/re-used in code → created by square brackets by commas that can either be string, int, or float objects

0 1 2 3 4 → index numbers

Ex: examplelist = [1, 2, 3, 4, 5]

-5 -4 -3 -2 -1

examplelist [1:3] = 2,3 → "start to stop, not including 3"

examplelist [0:4:2] = 1,3 → "start at 0, stop at 4, and count by 2"

1. You can access the list by the negative index numbers and the items in the list instead of the index numbers
2. You can use: len(list_name) to find the length of the list
3. You can use: list_name(index) = value to replace that index value with a new variable **or** list_name.insert(0,value) that will insert a value at the index value of 0 in the list you created
4. To delete a value in the list file use: del list_name[index]
5. To append a value in the list file use: list_name.append (value)
6. To add items to the list file use: list_name = value + list_name → this will add a value to the beginning of the list file

Operators in Python: symbols that carry out specific computations or operations in python

1. Arithmetic: mathematic computations (+, -, *, /, **)
2. Comparison: used when trying to find if values are greater than or equal to a value you are trying to find → **true or false** (=, !=, >, >=, <, <=)
3. Logical: used when trying to determine whether values are within a ven diagram or not → **true or false** (and, or, not)
4. Membership: used to check whether a value is included in another operand or not → **true or false** (in, not in)
5. Identity: used to check whether an operand is the same or not → **true or false** (is, is not)

6. Assignment: used to assign new values (ex: var = 10, var += 2 → 12, which is the same as: var = var + 2→ 12)
    a. You use print(var) = 12

Conditional Statements: used to determine whether you already ran something else in your code that is the same (if: or else:)

DRY (Don't repeat yourself): make easy code to work with→ easier for you and easier for people to add and follow along with code you made

- Elif statement: use this to check for an alternative condition (I don't really understand this?) → use this after the if statement is not satisfied
- If → elif → else

Conditional statements with a combination of conditions:

1. And: executing a code when all specifications have been made
2. Or: executing a code when at least one specified condition has been made
3. Not: executing a code when no specified condition has been made

Loops: sequence of operations performed repeatedly in a specific order → helps make code cleaner and easier to read (DRY) → "for" and "while" loops

Ex: "for avalue in list_values:" then "print avalue" → should output all the values instead of printing all the values in brackets

1. For loops are used to execute values from a list that was already defined
    a. You can also add a number with placeholder, and it will add that number to each of the numbers in the already defined list
    b. You can also create for loops with text strings (fname)
    c. You can also create for loops on data structures (dlist)