MIT OpenCourseware 6.0001 Introduction to Computer Science and Programming in Python
Lecture 2: Branching and Iteration

Reminder: a computer only does what it is told
Objects and operations

A new object: a String
How do we branch and iterate?

A **STRING**
-sequences of characters
-enclosed in ‘’ or “”
-object bound to the string value

Concatenate: use + to add them together
(in print statements, you can use commas to add objects tg and they can be different types, but python adds a space in between (unlike w the +))
Ex. hi = “hi”
name = “maanya”
print(hi + name)
himaanya
X = (hi + “_” + name)
print(X)
hi_maanya
print(X*3)
hi_maanyahi_maanyahi_maanya

***input** gets input from the user and waits for them to type and enter*
-made as a string (can be bound to diff type of object by CASTING)

Add tests in your code by using comparison operators
EVAL TO BOOLEAN:
i > j
i < j
i >= j
i <= j
i == j (equality)
i != j (inequality)

can use keywords and/or (boolean truth tables)

This adds BRANCHING in our code, because we can perform tests to make decisions…
with key words **if/elif/else (control flow commands)**

DENOTE the flow of control using INDENTATION (matters!)
-nested conditionals should also be intended
-in conditionals, do not compare using = (assignment), instead use == (comparison)

LOOPS
-while
-repeat until a condition is met
-can be infinite
-also uses indentation

-for
-iterates through a sequence/x amount of times
-counter variables (don't need to initialize separately)
-better for exact num of iterations
(starts at a value, increases by increment, ends at some num-1)
**must be integers

for x in range(start, stop, step)

-default start = 0, step = 1
-loop until stop-1

Exit loop early using **break** which exists the innermost loop without executing remaining expressions in the code block that is exited