6.100L Recitation 2 – September 16, 2022

Reminders:

- MQ 2 next Wednesday
- PS1 half way hand in due next Wednesday
- Finger exercises before each lecture

Lecture 2 Recap: Strings, Input/Output, Branching

Strings

- New data type – it is a sequence of characters
  - `my_string = "Hello world!"`
- They can be indexed and sliced:
  - `my_string[0]  # outputs "H"`
  - `my_string[2]  # outputs "l"`
  - `my_string[-1]  # outputs "!"`
  - `my_string[-2]  # outputs "d"`
  - `my_string[1:3]  # outputs "el"`
- We can concatenate strings
  - `my_new_string = my_string + ' ' +  my_string`

Input

- Done with the *input* command
- Anything the user inputs is read as a string object!
  - `x = input("Enter a string:  ")  # what the user inputs is assigned to x as a string`
- Can cast a user input as an integer
  - `x_as_int = int(input("enter and int: "))  # here x will be an integer`

Output

- Done with the *print* command
  - `print(x)`
  - `print(("x = ", x)` -> (comma concatenates with a space between)
  - print statements are super useful for debugging! especially to see what is happening in loops

Branching

- Idea that we only want to execute certain blocks if specific conditions are satisfied
- We create a code structure to satisfy out requirement.

```
Example branching:
x = 2
if x == 3:
   print("x is 3!")
elif x == 2:
   print("x is 2!")
else:
   print("x is neither 2 or 3")
```

Lecture 3 Recap: Loops & Iteration Methods

Looping Mechanisms

- Loop over ranges of numbers
- Loop over elements of a string
- Main idea – want to repeat things multiple times → reuse code.

For loops

`for` loops have pre-specified range over which they run.

`for i in range(x):`
- i goes from 0 to x-1

`for char in s:`
- char is string that takes on the value of each character in s

While loops

`while` loops have a condition that they check to determine if they should keep running. They run until the condition no longer evaluates to True.

```
counter = 0
while counter < 3:
      print(counter)
      counter += 1
```