CSE 6040/x Bootcamp

Loops and Comprehensions

Loops and Comprehensions

- A for loop can iterate over every item in a list or go through every single character in a string and won't stop until it has gone through every character.

- Writing for loops helps reduce repetitiveness in your code, following the DRY (Don't Repeat Yourself) principle. You don't write the same block of code more than once.

- We can create new sequences using a given python sequence.
  - Can be done using loops
  - Can be done using Comprehensions (shorter way , but may take some time to get used to)

- Comprehension is basically a way of writing a concise code block to generate a sequence which can be a list, dictionary, set by using another sequence

Note - Examples are in the accompanying notebook

Comprehensions pros and cons

- Using loops is **okay.** However, it is worth understanding comprehensions because they have their own benefits.

- Pros:
  - Generally faster than for loops, especially for large datasets.
  - Takes less code to write and fits in a smaller space than a for loop.

- Cons:
  - Can be less legible in certain situations.
  - Can be harder to implement for complicated operations in for loops.

- As you progress through the course, you will see problem solutions that use comprehensions whenever possible which will enable you to wrap your head around them.

- Advice – Always pause and think if you can use a comprehension instead of a loop. Practice, Practice, Practice!!!