Notes on List Comprehension

Last updated Fall 2019

Disclaimer: These are not official class notes. They're just meant to be a quick reference. Please let me know if there are any typos or mistakes.

Overview

The idea behind list comprehension is to "do something" to an entire list in one line. It doesn't really improve you code performance wise, but sometimes may help make your code more readable (compared to a for loop).

The structure of a list comprehension is:

```
[<element in new list> for <element in old list> in <old list>]
```

If you wish to add an `if` condition, you can do so with

```
[<element in new list> for <element in old list> in <old list> if <condition>]
```

Examples

I'll go through a couple of examples showing you what you can do and can't do with list comprehension.

1. Let's say I want to square every element of a list. Using a `for` loop, the code could look something like

   ```
   lst = [1, 2, 3, 4, 5]
   lst2 = []
   for elem in lst:
       lst2 += [elem **2]
   ```

   Using list comprehension, we can simplify that down to `python  lst = [1, 2, 3, 4, 5]  lst2 = [elem**2 for elem in lst]`

   Notice how the code looks cleaner and is easier to read?

2. Next, let's try to filter out all the odd numbers of a list. Using a `for` loop, the code could look something like

   ```
   lst = list(range(10)) # creates a list of integers from 0 - 9 (inclusive)
   lst2 = []
   for elem in lst:
       if elem % 2 == 0:
           lst2 += [elem]
   ```

   Using list comprehension, we can simplify that down to `python  lst = list(range(10))  lst2 = [elem for elem in lst if elem % 2 == 0]`

3. Now, let's square only the even elements of a list. Otherwise, keep the original element. Using a `for` loop, the code could look something like

   ```
   lst = list(range(10))
   lst2 = []
   for elem in lst
       if elem % 2 == 0:
           lst2 += [elem ** 2]
       else:
           lst2 += [elem]
   ```

   We can try to do this using list comprehension, but how do we represent the `if/else`?

Turns out there is a way to do `if/elif/else` in list comprehension. This isn't part of this course, but if you're interested, you can find out more by going to this website. To solve our problem, we could do `python  lst = list(range(10))  lst2 = [elem if elem % 2 != 0 else elem**2 for elem in lst]`

4. For the final example, let's try to mutate a list such that the even elements are squared. Using a `for` loop, the code could look something like

```
lst = list(range(10))
for i in range(len(lst)):
    if lst[i] % 2 == 0:
        lst[i] **= 2
```

How can we do this using list comprehension? Turns out you can't. This is because list comprehension will always create a new list. You can't use list comprehension to mutate an existing list.