# Lists III

- Counting all items in a list that match a pattern:

```
total = 0
for pos in range(0, len(L)):
  if <test L[pos] for something>:
    total = total + 1
```

- Filtering all items in a list that match a pattern: (to make a new list of all the items that match)

```
newList = []
for pos in range(0, len(L)):
  if <test L[pos] for something>:
    newList.append(L[pos])
```

- Transforming all items in a list:

```python
newList = []
for pos in range(0, len(L)):
    newList.append(<something that
                    operates on L[pos]>)
```

- Transforming all items in a list without making a copy:

```
for pos in range(0, len(L)):
    L[pos] = <something different>
```

- How could we write code to take a list of integers and add one to all of them?

- How can we write code to take a list of strings and change the first letter of each string to uppercase?

- Write a function called count_initial_vowels that takes a list of names and returns the number of people in class whose name starts with a vowel.

- Write a function called filter_one_vowel that takes a list of names and returns a list of the people in class who only have one vowel in their name.

- Write a function called count_spares that takes two parallel lists, rolls1 and rolls2, as parameters. This function counts the number of times you roll a spare (not a strike) in bowling. (A spare is a sum of 10 from the two rolls, but not all on the first roll.)

- Minimum and maximum for lists:

- Very similar to min and max for reading from a file!

- Find the smallest item in a list (list is called L):

```
smallest = L[0]
for pos in range(0, len(L)):
  if L[pos] < smallest:
    smallest = L[pos]
```

After this loop, **smallest** holds the minimum item in L.

- Suppose I own a store and I have two parallel lists called **products** and **quantities**.
- Every week I check to see which product I have the least of in stock, and I order ten more of those.
  - I want to change **quantities** to reflect this.
- How can I write code for this?

- Why will this code not work perfectly?

```
smallest = quant[0]
for pos in range(0, len(quant)):
  if quant[pos] < smallest:
    smallest = quant[pos]

smallest += 10
```

- Better: save the *position* of the smallest item, not the value of the item itself.

```python
smallest = quant[0]
smallest_pos = 0
for pos in range(0, len(quant)):
  if quant[pos] < smallest:
    smallest = quant[pos]
    smallest_pos = pos

quant[smallest_pos] += 10
```