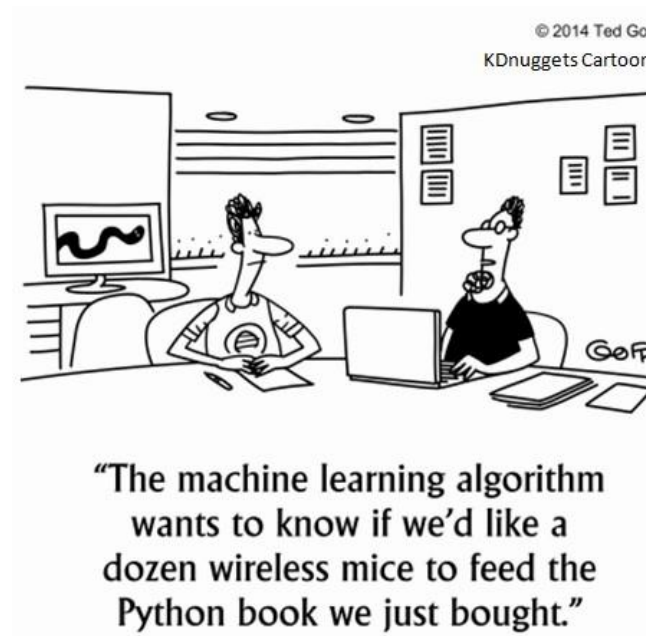


CSc 110, Spring 2017

Lecture 15: lists

Adapted from slides by Marty Stepp and Stuart Reges



Can we solve this problem?

- Consider the following program (input underlined):

How many days' temperatures? 7

Day 1's high temp: 45

Day 2's high temp: 44

Day 3's high temp: 39

Day 4's high temp: 48

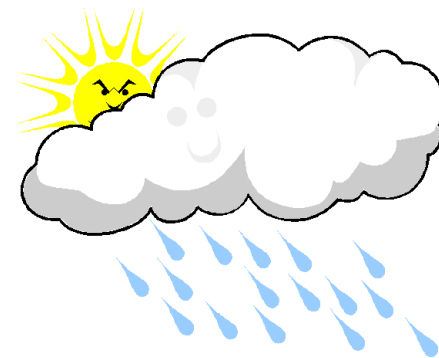
Day 5's high temp: 37

Day 6's high temp: 46

Day 7's high temp: 53

Average temp = 44.6

4 days were above average.

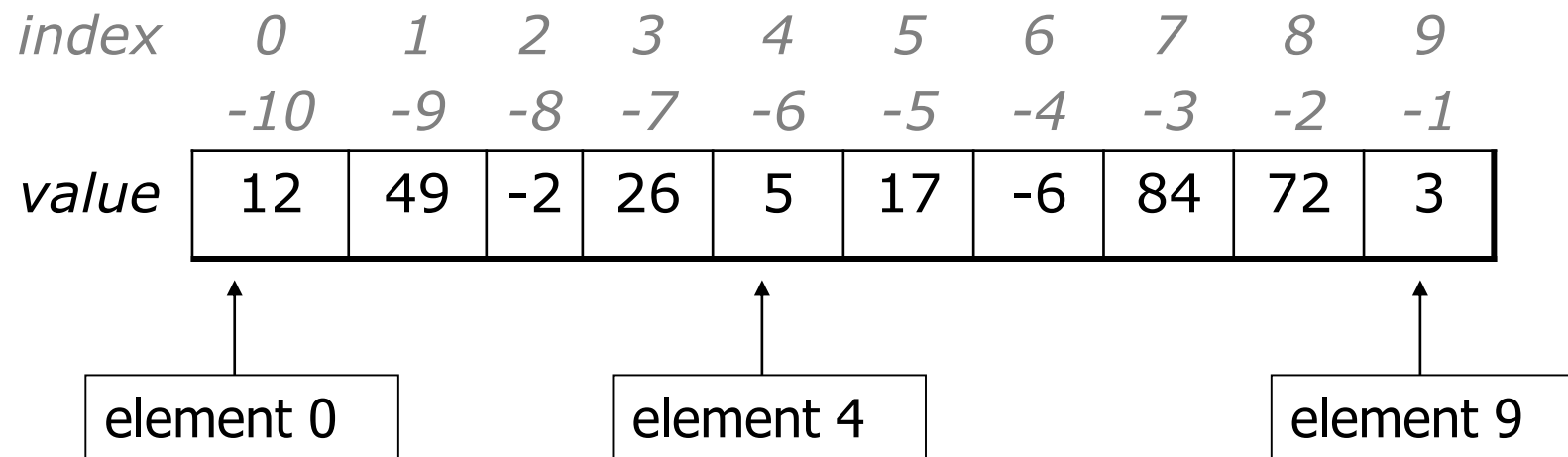


Why the problem is hard

- We need each input value twice:
 - to compute the average (a cumulative sum)
 - to count how many were above average
- We could read each value into a variable... but we:
 - don't know how many days are needed until the program runs
 - don't know how many variables to declare
- We need a way to hold a sequence of values (and of course a way to reference them...)

Lists

- **list:** a type that holds a sequence of zero or more values.
 - **element:** One value in a list.
 - **index:** A 0-based integer used to access an element from an list.



List initialization

name = [**value**, **value**, ... **value**]

- Example:

```
numbers = [12, 49, -2, 26, 5, 17, -6]
```

index 0 1 2 3 4 5 6

value

12	49	-2	26	5	17	-6
----	----	----	----	---	----	----

- An alternate form when the values are the same:

name = [**value**] * **count**

- Example:

```
numbers = [0] * 4
```

index 0 1 2 3

value

0	0	0	0
---	---	---	---

Accessing elements

name [index] # access

name [index] = value # modify

- Example:

```
numbers = [0] * 2
```

```
numbers[0] = 27
```

```
numbers[1] = -6
```

```
print(numbers[0])
```

```
if (numbers[1] < 0):
```

```
    print("Element 1 is negative.")
```

index 0 1

value

27	-6
-----------	-----------

Accessing list elements

```
numbers = [0] * 8
```

```
numbers[0] = 3
```

```
numbers[1] = 99
```

```
numbers[2] = 6
```

x 3

```
x = numbers[0]
```

```
numbers[x] = 42
```

```
numbers[numbers[2]] = 11 # use numbers[2] as index
```

<i>numbers</i>	<i>index</i>	0	1	2	3	4	5	6	7
	<i>value</i>	3	99	6	42	0	0	11	0

Out-of-bounds

- Legal indexes to use []: between **list's length** and the **list's length - 1**.
- Note: this is just like strings
 - Reading or writing any index outside this range with [] will cause an `IndexError: list assignment index out of range`

- **Example:**

```
data = [0] * 10
print(data[0])           # okay
print(data[9])          # okay
print(data[-20])       # error
print(data[10])       # error
```

<i>index</i>	0	1	2	3	4	5	6	7	8	9
<i>value</i>	0	0	0	0	0	0	0	0	0	0

Lists and `for` loops

- It is common to use `for` loops to access list elements.

```
for i in range(0, 8):  
    print(str(numbers[i]) + " ", end='')  
print() # output: 3 99 6 42 0 0 11 0
```

- Sometimes we assign each element a value in a loop.

```
for i in range(0, 8):  
    numbers[i] = 2 * i
```

index 0 1 2 3 4 5 6 7

<i>Value</i>	0	2	4	6	8	10	12	14
--------------	---	---	---	---	---	----	----	----

len

- `len(list)` returns the number of elements in a list.

```
for i in range(0, len(numbers)):  
    print(numbers[i] + " ", end='')  
# output: 0 2 4 6 8 10 12 14
```

- What value is produced by:
 - `len([10, 20, 30])`
 - `len([3, 4] * 2)`

Lists and `for` loops

- You can use the `len` function to loop through a list

```
counts = [1, 3, 6, 23, 43, 12]
for i in range(0, len(counts)):
    print(str(counts[i]) + " ", end='')
print() # output: 1 3 6 23 43 12
```

- Or, you can also loop directly over lists, just as with strings

```
counts = [1, 3, 6, 23, 43, 12]
for number in counts:
    print(str(number) + " ", end='')
print() # output: 1 3 6 23 43 12
```

Weather question

- Use a list to solve the weather problem:

How many days' temperatures? 7

Day 1's high temp: 45

Day 2's high temp: 44

Day 3's high temp: 39

Day 4's high temp: 48

Day 5's high temp: 37

Day 6's high temp: 46

Day 7's high temp: 53

Average temp = 44.6

4 days were above average.

Weather answer

```
# Reads temperatures from the user, computes average and # days above average.
```

```
def main():
```

```
    days = int(input("How many days' temperatures? "))
```

```
    temps = [0] * days                # list to store days' temperatures
```

```
    sum = 0
```

```
    for i in range(0, days):          # read/store each day's temperature
```

```
        temps[i] = int(input("Day " + (i + 1) + "'s high temp: "))
```

```
        sum = sum + temps[i]
```

```
    average = sum / days
```

```
    count = 0                          # see if each day is above average
```

```
    for i in range(0, days):
```

```
        if (temps[i] > average):
```

```
            count = count + 1
```

```
    # report results
```

```
    print("Average temp = " + str(average))
```

```
    print(str(count) + " days above average")
```

Weather question 2

- Modify the weather program to print the following output:

```
Type in a temperature or "done" to finish
```

```
Day 1's high temp: 45
```

```
Day 2's high temp: 44
```

```
Day 3's high temp: 39
```

```
Day 4's high temp: 48
```

```
Day 5's high temp: 37
```

```
Day 6's high temp: 46
```

```
Day 7's high temp: 53
```

```
Day 7's high temp: done
```

```
Average temp = 44.6
```

```
4 days were above average.
```

List functions

Function	Description
<code>append(x)</code>	Add an item to the end of the list. Equivalent to <code>a[len(a):] = [x]</code> .
<code>extend(L)</code>	Extend the list by appending all the items in the given list. Equivalent to <code>a[len(a):] = L</code>
<code>insert(i, x)</code>	Inserts an item at a given position. <code>i</code> is the index of the element before which to insert, so <code>a.insert(0, x)</code> inserts at the front of the list.
<code>remove(x)</code>	Removes the first item from the list whose value is <code>x</code> . Errs if there is no such item.
<code>pop(i)</code>	Removes the item at the given position in the list, and returns it. <code>a.pop()</code> removes and returns the last item in the list.
<code>clear()</code>	Remove all items from the list.
<code>index(x)</code>	Returns the index in the list of the first item whose value is <code>x</code> . Errs if there is no such item.
<code>count(x)</code>	Returns the number of times <code>x</code> appears in the list.
<code>sort()</code>	Sort the items of the list
<code>reverse()</code>	Reverses the elements of the list
<code>copy()</code>	Return a copy of the list.

Weather 2 answer

```
# Reads temperatures from the user, computes average and # days above average.
```

```
def main():  
    print('Type in a temperature or "done" to finish')  
  
    temps = []                # list to store days' temperatures  
    sum = 0  
    done = input("Day 1's high temp: ")  
    day = 1  
  
    while(done != "done"):    # read/store each day's temperature  
        done = int(done)  
        sum = sum + done  
        temps.append(done)  
        done = input(("Day " + str(day + 1) + "'s high temp: "))  
        day = day + 1  
    average = sum / day  
  
    count = 0                 # see if each day is above average  
    for i in range(0, day - 1):  
        if (temps[i] > average):  
            count = count + 1  
  
    # report results  
    print("Average temp = " + str(average))  
    print(str(count) + " days above average")
```


Weather question 3

- Modify the weather program to print the following output:

```
How many days' temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.
```

```
Temperatures: [45, 44, 39, 48, 37, 46, 53]
Two coldest days: 37, 39
Two hottest days: 53, 48
```

Weather answer 3

```
# Reads temperatures from the user, computes average and # days above average.
```

```
def main():
    days = int(input("How many days' temperatures? "))

    temps = [0] * days                # list to store days' temperatures
    sum = 0

    for i in range(0, days):        # read/store each day's temperature
        temps[i] = int(input("Day " + (i + 1) + "'s high temp: "))
        sum = sum + temps[i]
    average = sum / days

    count = 0                        # see if each day is above average
    for i in range(0, days):
        if (temps[i] > average):
            count += 1

    # report results
    print("Average temp = " + str(average))
    print(str(count) + " days above average")

    print("Temperatures: " + str(temps))
    temps.sort()
    print("Two coldest days: " + str(temps[0]) + ", " + str(temps[1]))
    print("Two hottest days: " + str(temps[-1]) + ", " + str(temps[-2]))
```

"list mystery" problem

- **traversal:** A sequential processing of the elements of a list.
- What element values are stored in the following list?

```
a = [1, 7, 5, 6, 4, 14, 11]
for i in range(0, len(a) - 1):
    if (a[i] > a[i + 1]):
        a[i + 1] = a[i + 1] * 2
```

<i>index</i>	0	1	2	3	4	5	6
<i>value</i>	1	7	10	12	8	14	22