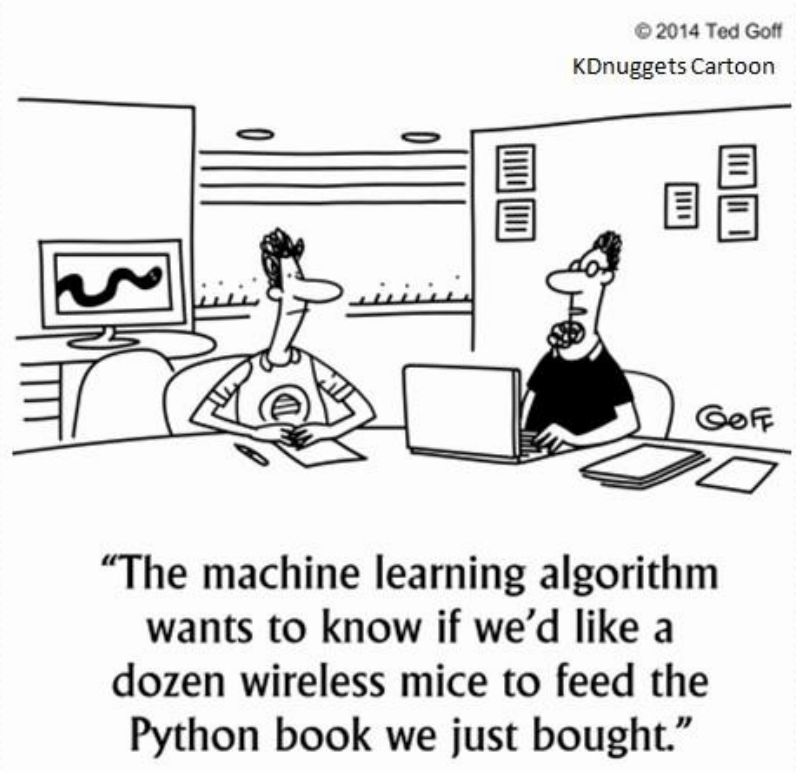
CS 115, Autumn 2021

Lecture 27: returns; lists



Thanks to Marty Stepp and Stuart Reges for parts of these slides

Exercise: Computing quiz grade

- Write a program to compute a students' overall quiz percentage.
  - Read three scores for each quiz from the user
  - Print out the largest of those scores (final points for the quiz)
  - Print out the average score for the first three quizzes
    - Assume all quizzes are worth 10 points

```
Enter your quiz scores (three for each quiz) and your
overall quiz percentage will be printed.

Quiz 1 score 1? 9
Quiz 1 score 2? 10
Quiz 1 score 3? 8
Your final score for quiz 1 is 10

...

Your overall quiz percentage is: 100%
```

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

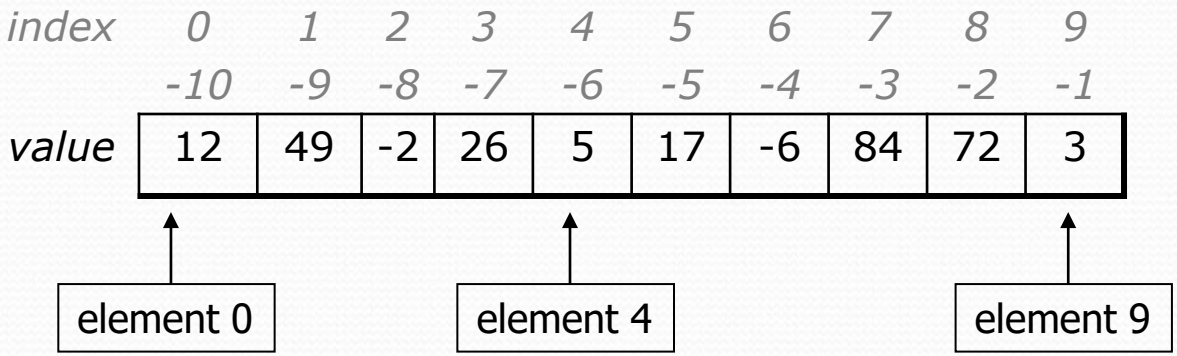
- We need each input value twice:
  - to compute the average (a cumulative sum)
  - to count how many were above average

- 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 declare many variables in one step.

Lists

- **list**: object that stores many values.
  - **element**: One value in a list.
  - **index**: A 0-based integer to access an element from a list.

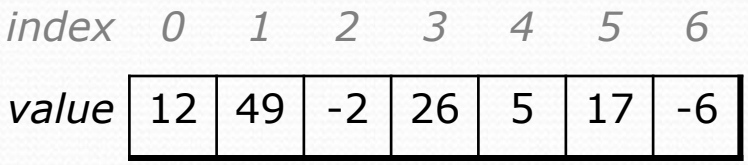


List initialization

name = [**value**, **value**, … **value**]

- Example:

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

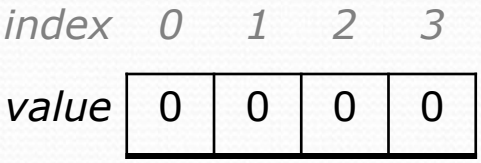


- Useful when you know what the list's elements will be

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

- Example:

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



Accessing elements

```
name[index]                    # access
name[index] = value            # modify
```

- Example:

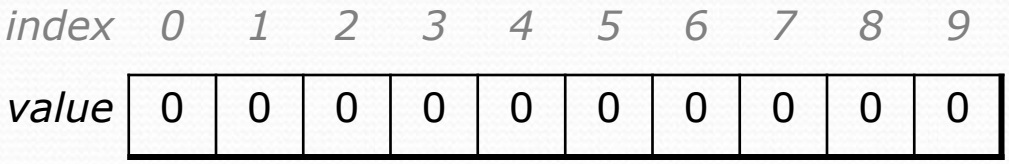
```
numbers = [0] * 2
numbers[0] = 27
numbers[1] = -6

print(numbers[0])
if (numbers[1] < 0):
    print("Element 1 is negative.")
```

index	*0*	*1*
value	**27**	**-6**

Out-of-bounds

- Legal indexes to use []: between **– list's length** and the **list's length - 1**.
  - 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
```



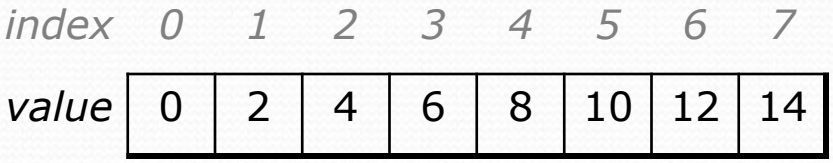
Lists and `for` loops

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

```
for i in range(0, 8):
    print(str(numbers[i]) + " ", end='')
print()  # output: 0 4 11 0 44 0 0 2
```

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

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



len()

- Use `len()` to find 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 expressions refer to:
  - The last element of any list?
  - The middle element?

Lists and `for` loops

- You can also loop directly over lists:

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

  - Each element in `list` is stored in the `number` variable, a different one on each iteration
    - The first time through `number` is `1`
    - The second time through `number` is `3`
    - The third time through `number` is `6`

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 += 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")
```

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 declaration

name = []

- Example:

```
numbers = []
```

Creates an empty list

List functions

Function	Description
`append(x)`	Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.
`extend(L)`	Extend the list by appending all the items in the given list. Equivalent to `a[len(a):] = L`
`insert(i, x)`	Inserts an item at a given position. i is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list.
`remove(x)`	Removes the first item from the list whose value is *x*. Errs if there is no such item.
`pop(i)`	Removes the item at the given position in the list, and returns it. `a.pop()` removes and returns the last item in the list.
`clear()`	Remove all items from the list.
`index(x)`	Returns the index in the list of the first item whose value is *x*. Errs if there is no such item.
`count(x)`	Returns the number of times *x* appears in the list.
`sort()`	Sort the items of the list
`reverse()`	Reverses the elements of the list
`copy()`	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 += done
        temps.append(done)
        done = input(("Day " + str(day + 1) + "'s high temp: "))
        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 += 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 += 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**: An examination of each element 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
```