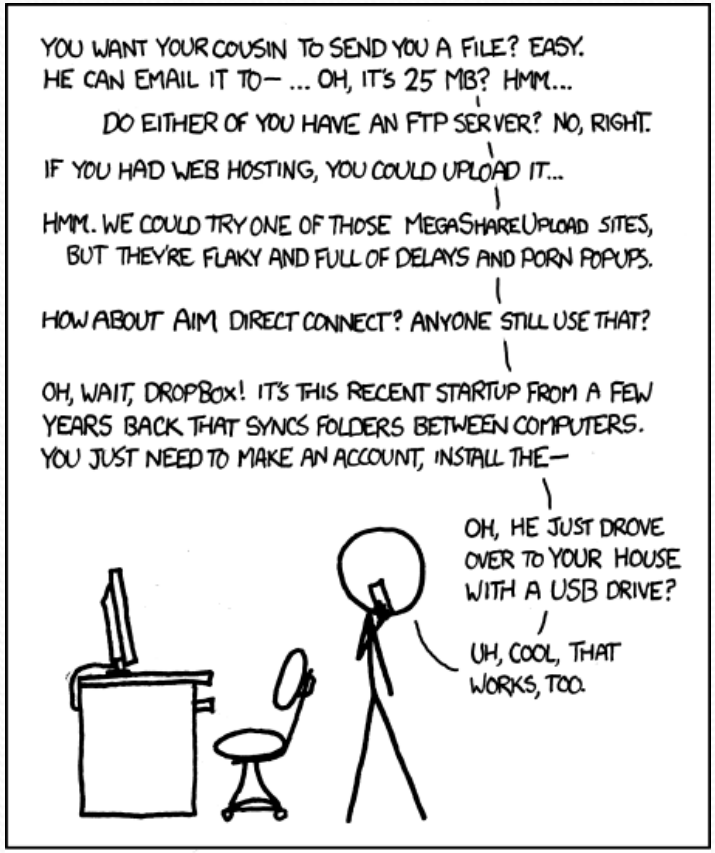
CS 115, Autumn 2021

Lecture 29: lists; files



I LIKE HOW WE'VE HAD THE INTERNET FOR DECADES, YET "SENDING FILES" IS SOMETHING EARLY ADOPTERS ARE STILL FIGURING OUT HOW TO DO.

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

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.
```

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
```

index	0	1	2	3	4	5	6	7
value	0	2	4	6	8	10	12	14

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
```

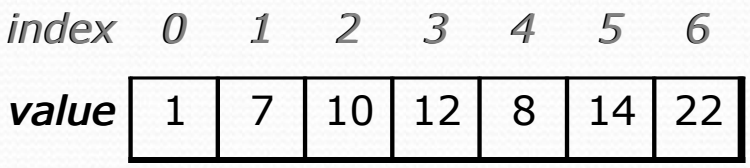
Weather answer 3

```
# 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
```



File Input/output (I/O)

- **name** = `open`("**filename**")
  - opens the given file for reading, and returns a file object

- **name**`.readlines()` - file's entire contents as a string

```
>>> f = open("weather.txt")
>>> f.readlines()
['42', '34', '35', '46', '45', '43', '43', '49']
```

File paths

- **absolute path**: specifies a drive or a top `"/"` folder

  `C:/Documents/smith/hw6/input/data.csv`

  - Windows can also use backslashes to separate folders.

- **relative path**: does not specify any top-level folder

  `names.dat`
  `input/kinglear.txt`

  - Assumed to be relative to the *current directory*:

    ```
    file = open("data/readme.txt")
    ```

    If our program is in `H:/hw6`,
    `open` will look for `H:/hw6/data/readme.txt`

File input question

- We have a file `weather.txt`:

```
16.2
23.5
19.1
7.4
22.8
18.5
-1.8
14.9
```

- Write a program that prints the change in temperature between each pair of neighboring days.

```
16.2 to 23.5, change = 7.3
23.5 to 19.1, change = -4.4
19.1 to 7.4, change = -11.7
7.4 to 22.8, change = 15.4
22.8 to 18.5, change = -4.3
18.5 to -1.8, change = -20.3
-1.8 to 14.9, change = 16.7
```

File input answer

```
# Displays changes in temperature from data in an input file.

def main():
    input = open("weather.txt")
    lines = input.readlines()
    prev = float(lines[0])          # fencepost

    for i in range(1, len(lines)):
        next = float(lines[i])
        print(prev, "to", next,  ", change =", (next - prev))
        prev = next
```

Gas prices question

- Write a program that reads a file `gasprices.txt`
  - Format: *Belgium $/gal US $/gal date …*

```
8.20 3.81 3/21/11 8.08 3.84 3/28/11 ...
```

- The program should print the average gas price over all data in the file for both countries:

```
Belgium average: 8.3
USA average: 3.9
```

Multiple tokens on one line

You can use `read` to read the whole file into a string and the `split` function to break a file apart

- **str**`.split()` – splits a string on blank space
- **str**`.split(`**other_str**`)` – splits a string on occurrences of the other string

```
>>> f = open("hours.txt")
>>> text = f.read()
'1 2\n45 6\n'

>>> f = text.split()
['1', '2', '45', '6']
```

Looping through a file

- The result of `split` can be used in a `for` ... `in` loop
- A template for reading files in Python:

```
file = open("filename")
text = file.read()
text = text.split()
for line in text:
        statements
```

Gas prices solution

```
def main():
    file = open("gasprices.txt")
    belgium = 0
    usa = 0
    count = 0
    lines = file.read().split()

    for i in range(0, len(lines), 3):
      belgium += float(lines[i])
      usa += float(lines[i + 1])

    print("Belgium average:", (belgium / count), "$/gal")
    print("USA average:", (usa / count), "$/gal")
```