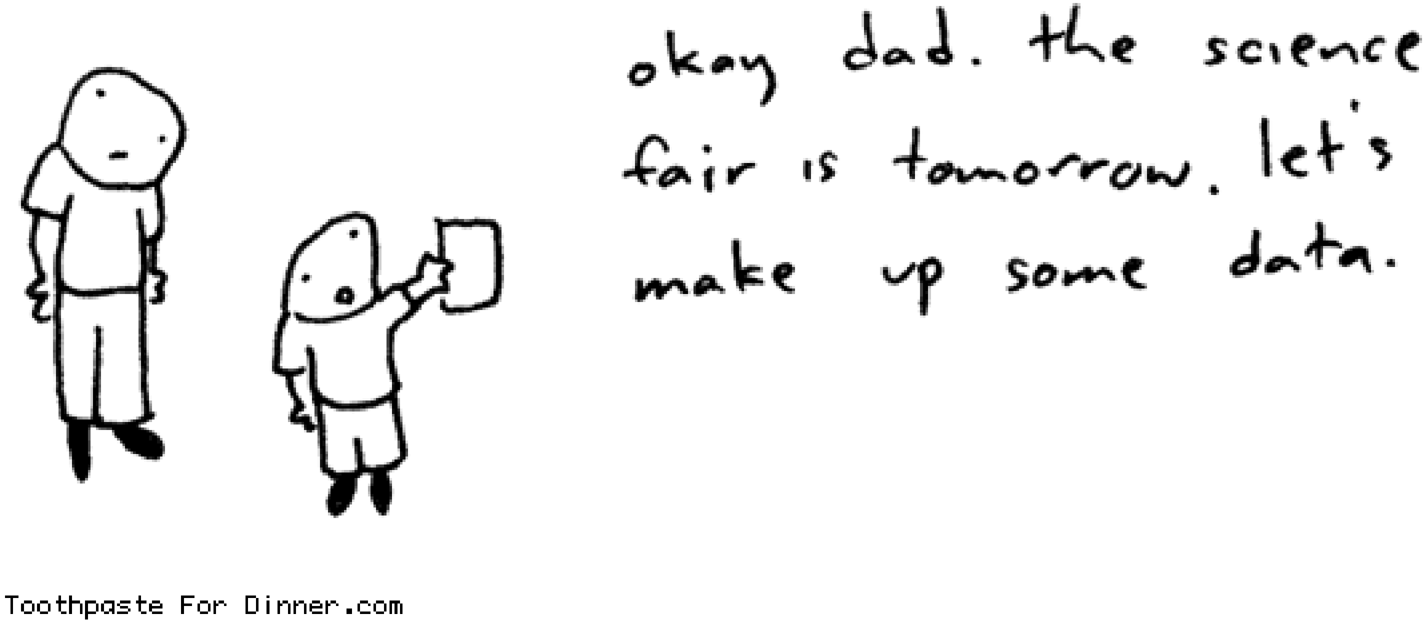
CSc 110, Autumn 2016

Lecture 16: File Input


okay dad. the science fair is tomorrow. let's make up some data.
Toothpaste For Dinner.com

Input/output (I/O)

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

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

```
>>> f = open("hours.txt")
>>> f.read()
'123 Brett 12.5 8.1 7.6 3.2\n
456 Sarina 4.0 11.6 6.5 2.7 12\n
789 Nick 8.0 8.0 8.0 8.0 7.5\n'
```

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 reading line by line

- **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 paths

- **name**.`readline()` – next line from file as a string
  - Returns an empty string if there are no more lines in the file

- **name**.`readlines()` – file's contents as a list of lines

```
>>> f = open("hours.txt")
>>> f.readline()
'123 Susan 12.5 8.1 7.6 3.2\n'

>>> f = open("hours.txt")
>>> f.readlines()
['123 Susan 12.5 8.1 7.6 3.2\n',
'456 Brad 4.0 11.6 6.5 2.7 12\n',
'789 Jenn 8.0 8.0 8.0 8.0 7.5\n']
```

Looping through a file

- A file object can be the target of a `for` ... `in` loop
- A template for reading files in Python:

```
for line in open("filename"):
        statements
```

```
>>> for line in open("hours.txt"):
...     print(line.strip())   # strip() removes \n

123 Susan 12.5 8.1 7.6 3.2
456 Brad 4.0 11.6 6.5 2.7 12
789 Jenn 8.0 8.0 8.0 8.0 7.5
```

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 = lines[i]
        print(str(prev) + " to " + str(next) + ", change = " +
              str(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
```

Gas prices solution

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

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

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