

File IO

Refers to reading and writing files

IO - Input/Output

Motivation

- Real applications need to process lots of data

- Data needs to persist beyond the lifetime of the program

Example files: video, spreadsheets, blogs, images, and more!

Files

Two primary types

Text files - sequences of characters

You can read text files in an editor such as Atom!

Binary files - general data (including floats, strings, and more) that can represent anything

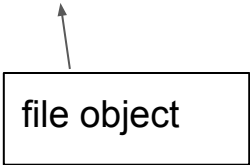
To read a binary file, your program must understand the format

Example: your browser can run video files (ogv) but Atom can't!

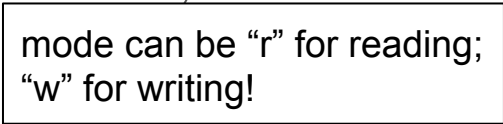
Reading files in code

Step 1: Open the file

```
infile = open(<name>, <mode>)
```



file object



mode can be "r" for reading;
"w" for writing!

Step 2: Read the contents

Usually done line by line!

```
for line in infile.readlines():  
    print(line)
```

Step 3: Close the file

```
infile.close()
```

Example: Shopping List - version 1.0

```
$ python3 shopping.py
```

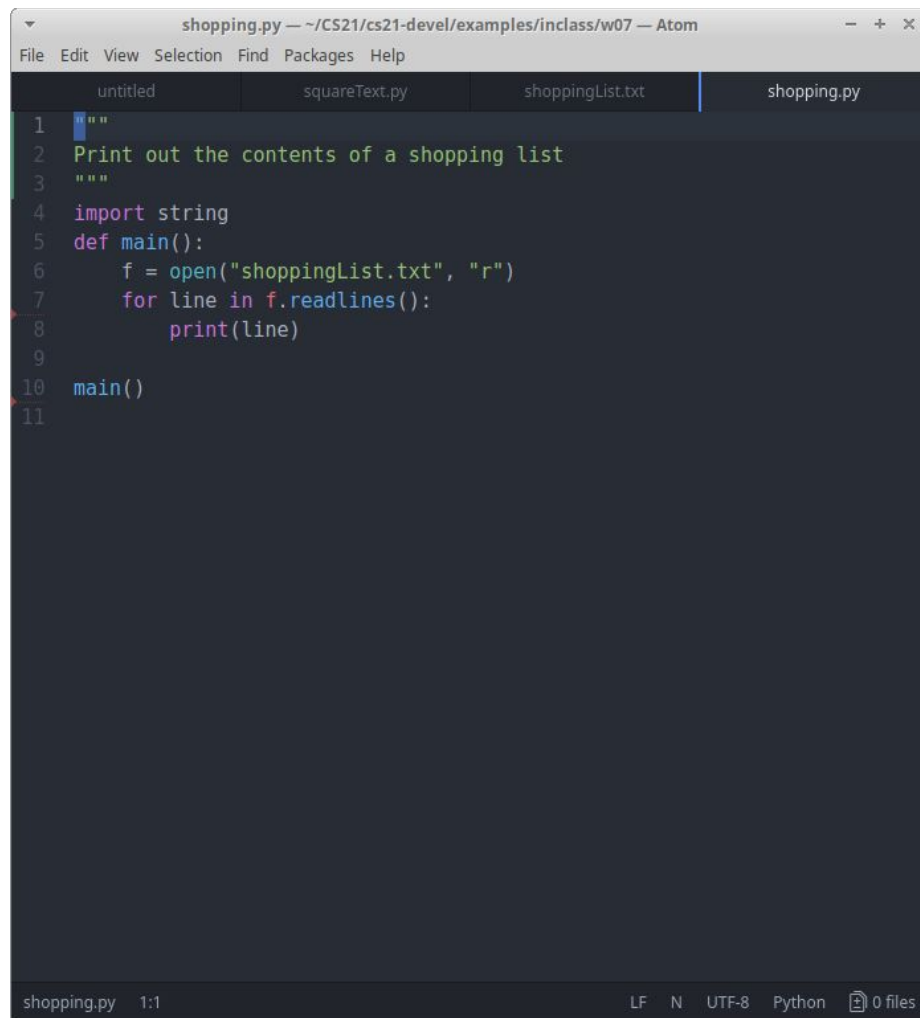
apples

bananas

flour

soap

cereal



```
shopping.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help

untitled squareText.py shoppingList.txt shopping.py

1 """
2 Print out the contents of a shopping list
3 """
4 import string
5 def main():
6     f = open("shoppingList.txt", "r")
7     for line in f.readlines():
8         print(line)
9
10 main()
11

shopping.py 1:1 LF N UTF-8 Python 0 files
```

Whitespace

When we read and write files, we need to be aware of whitespace characters!

“\n”, “\r” newlines

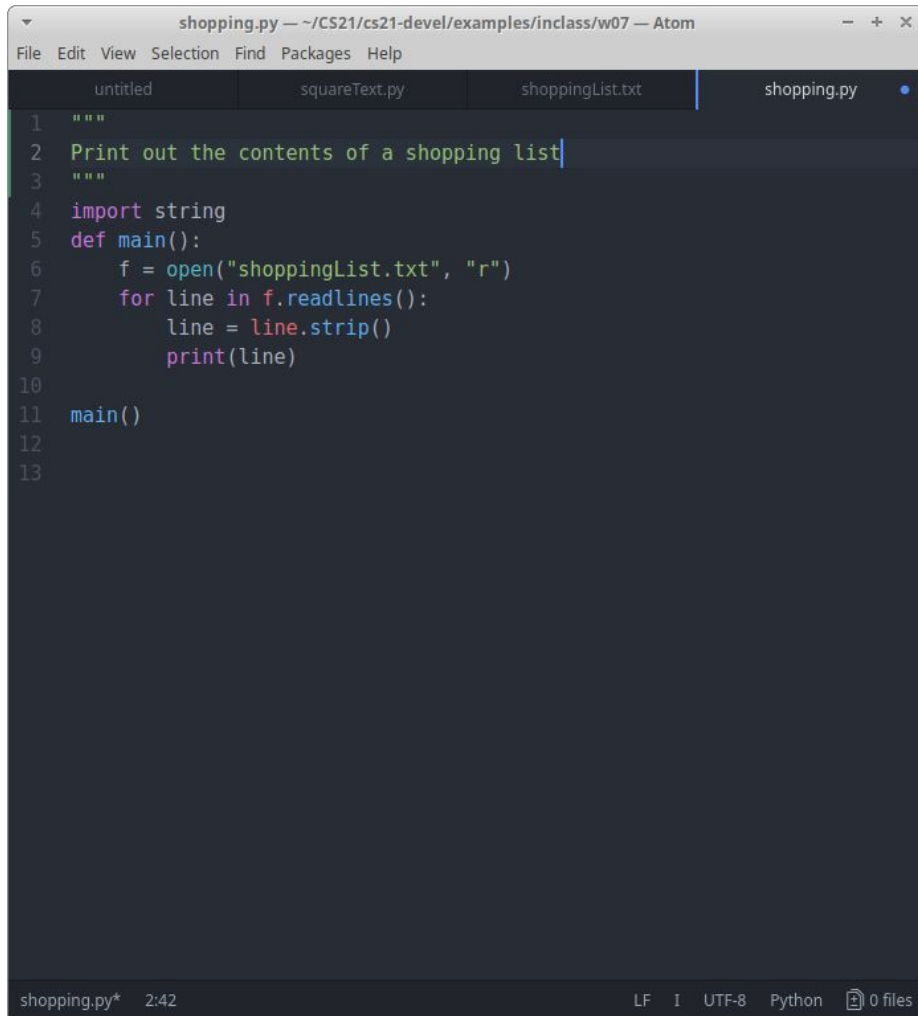
“\t” tab

Use the string module function, `strip()`, to remove whitespace characters!

```
for line in infile.readlines():  
    cleanLine = line.strip()  
    print(cleanLine)
```

Example: Shopping List - version 2.0

Stripping the line removes the extra newline character!



```
shopping.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help

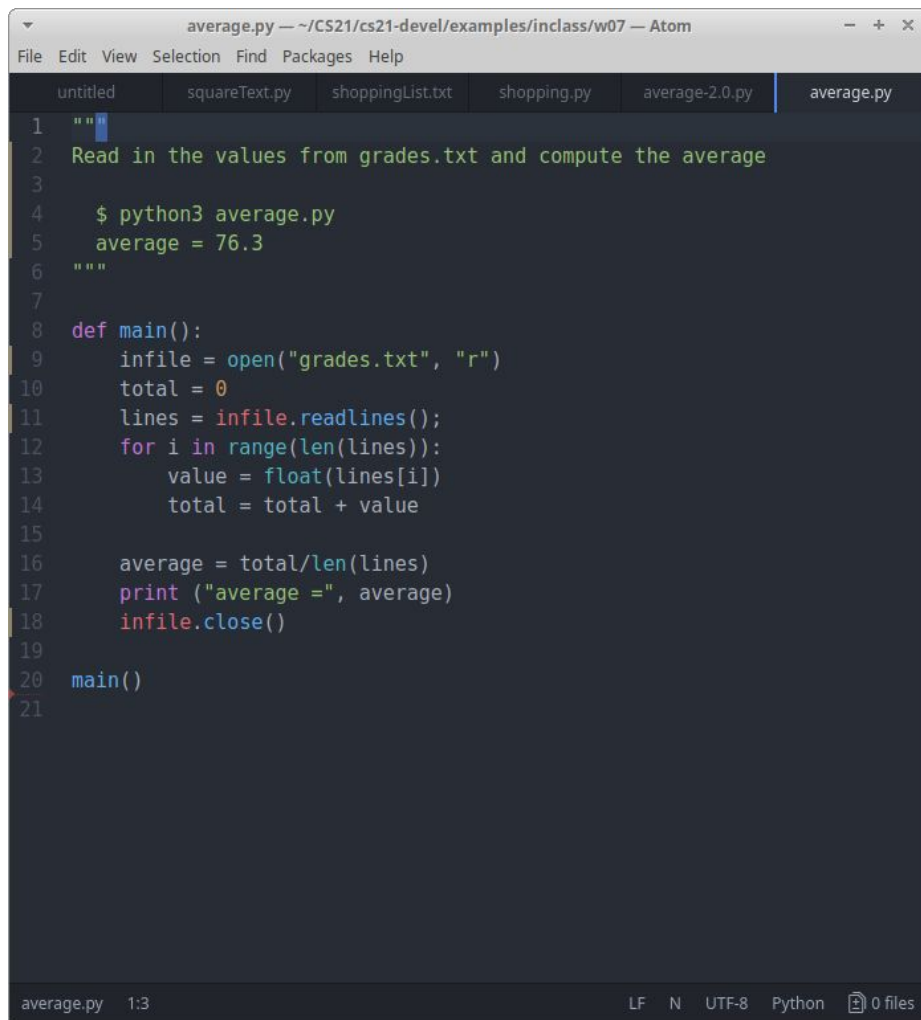
untitled squareText.py shoppingList.txt shopping.py

1 """
2 Print out the contents of a shopping list
3 """
4 import string
5 def main():
6     f = open("shoppingList.txt", "r")
7     for line in f.readlines():
8         line = line.strip()
9         print(line)
10
11 main()
12
13

shopping.py* 2:42 LF I UTF-8 Python 0 files
```

Example: Average grade - read and compute

Tip: Copy and paste the values into a spreadsheet program to make sure your calculations are correct!



```
average.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help
untitled squareText.py shoppingList.txt shopping.py average-2.0.py average.py

1 """
2 Read in the values from grades.txt and compute the average
3
4 $ python3 average.py
5 average = 76.3
6 """
7
8 def main():
9     infile = open("grades.txt", "r")
10    total = 0
11    lines = infile.readlines();
12    for i in range(len(lines)):
13        value = float(lines[i])
14        total = total + value
15
16    average = total/len(lines)
17    print ("average =", average)
18    infile.close()
19
20 main()
21
```

average.py 1:3 LF N UTF-8 Python 0 files

Example: Average grade - alternate solution

How does this differ from the previous? (Two changes)

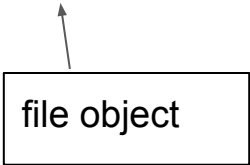
```
average.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help
untitled squareText.py shoppingList.txt shopping.py average-2.0.py average.py
1 """
2 Read in the values from grades.txt and compute the average
3
4 $ python3 average.py
5 average = 76.3
6 """
7
8 def main():
9     infile = open("grades.txt", "r")
10    total = 0
11    count = 0
12    for line in infile.readlines():
13        value = float(line)
14        total = total + value
15        count = count + 1
16
17    average = total/count
18    print("average =", average)
19    infile.close()
20
21    main()
22
```

average.py 18:5 LF N UTF-8 Python 0 files

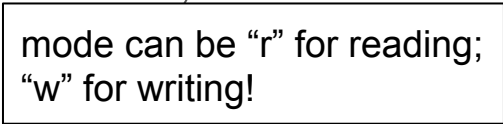
Writing files in code

Step 1: Open the file

```
outfile = open(<name>, <mode>)
```



file object



mode can be "r" for reading;
"w" for writing!

Step 2: Write the contents

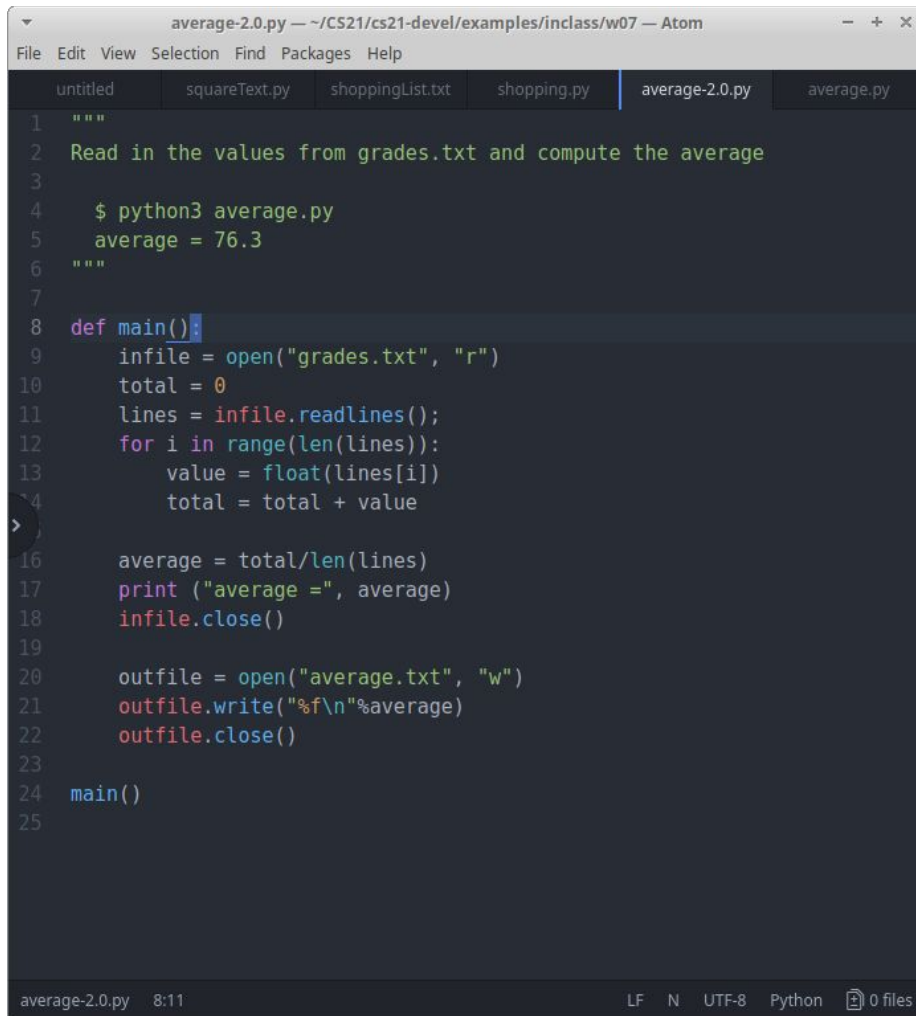
Need to explicitly add newlines!

```
outfile.write(value+"\n")
```

Step 3: Close the file (Important!)

```
outfile.close()
```

Example: Average grade - save to file



```
average-2.0.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help

untitled squareText.py shoppingList.txt shopping.py average-2.0.py average.py

1 """
2 Read in the values from grades.txt and compute the average
3
4 $ python3 average.py
5 average = 76.3
6 """
7
8 def main():
9     infile = open("grades.txt", "r")
10    total = 0
11    lines = infile.readlines();
12    for i in range(len(lines)):
13        value = float(lines[i])
14        total = total + value
15
16    average = total/len(lines)
17    print ("average =", average)
18    infile.close()
19
20    outfile = open("average.txt", "w")
21    outfile.write("%f\n"%average)
22    outfile.close()
23
24    main()
25
```

average-2.0.py 8:11 LF N UTF-8 Python 0 files

Parsing

Reading and converting text input into data types which we can use in our program

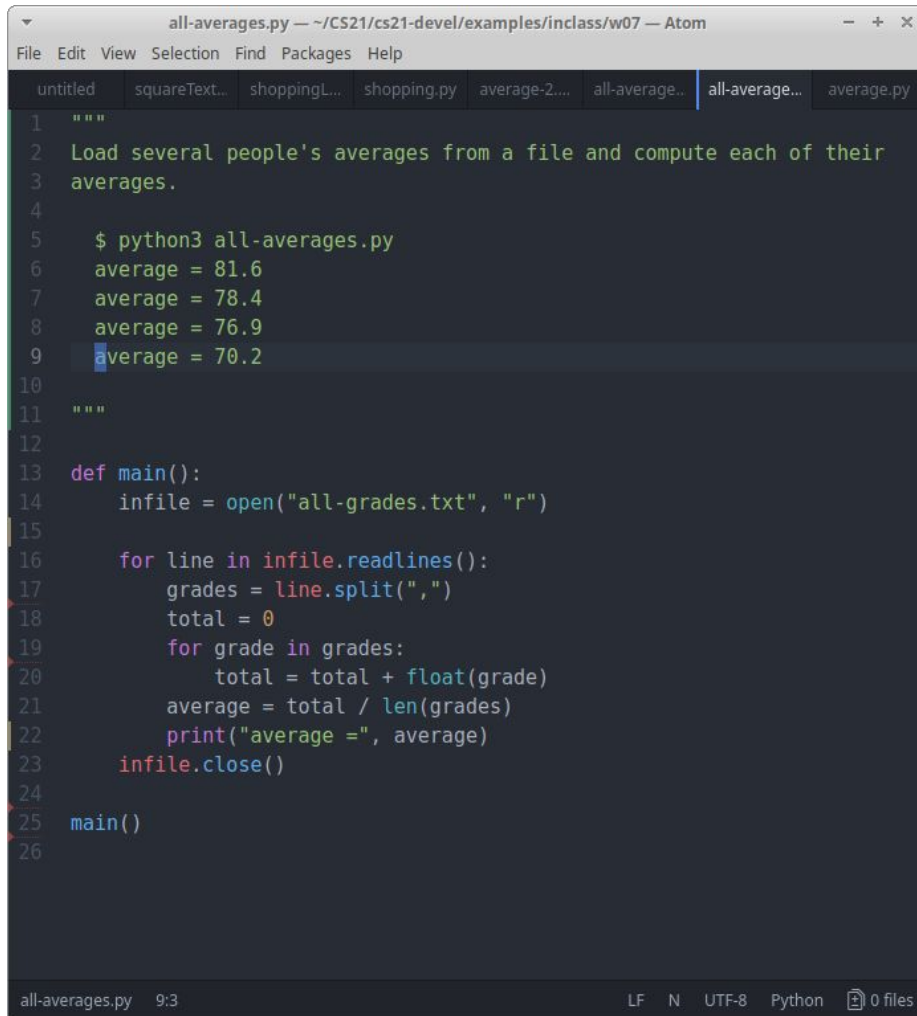
Example: Parse the string “10,90,34” so we can compute the total

```
valueStrings = "10,90,34"

# Use split to convert the above string to a list of strings
valueStringList = valueStrings.split(",")

# Convert each of the items in our list to an integer and add it to a total
total = 0
for i in range(len(valueStringList)):
    value = int(valueStringList[i])
    total = total + value
```

Example: Compute averages for multiple people



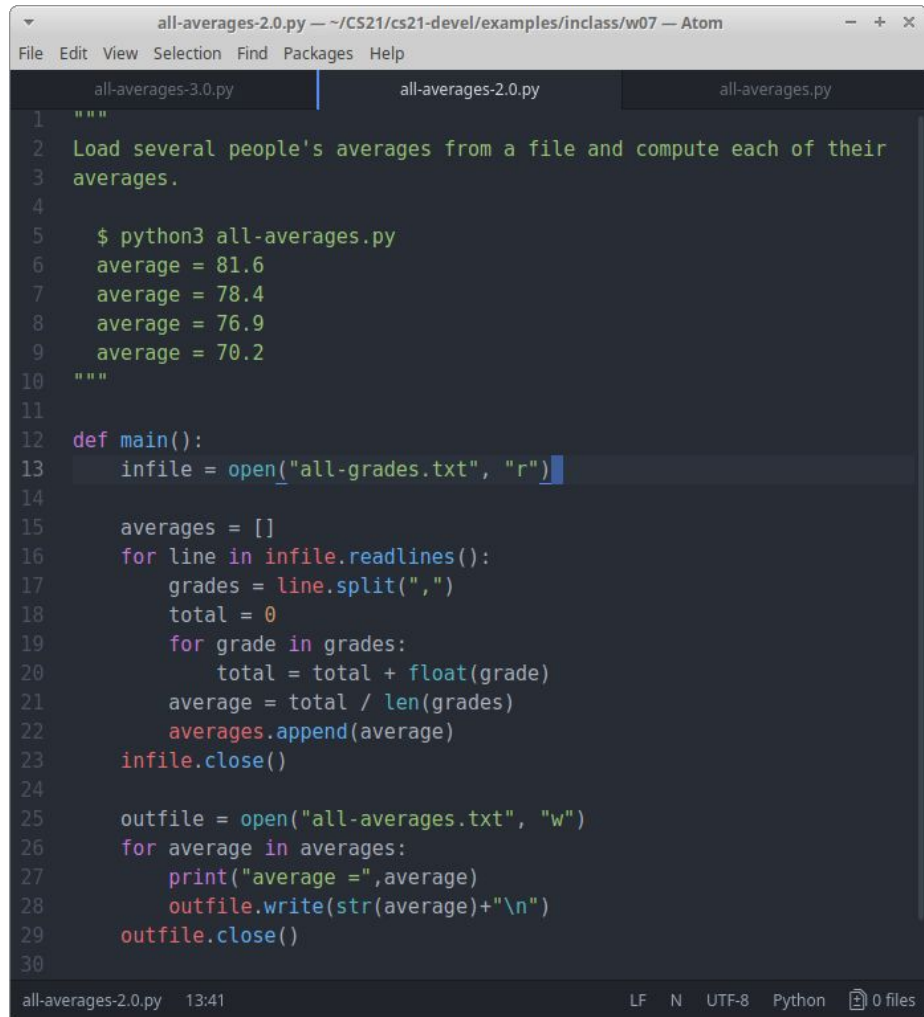
```
all-averages.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help
untitled squareText... shoppingL... shopping.py average-2... all-average... all-average... average.py

1 """
2 Load several people's averages from a file and compute each of their
3 averages.
4
5 $ python3 all-averages.py
6 average = 81.6
7 average = 78.4
8 average = 76.9
9 average = 70.2
10
11 """
12
13 def main():
14     infile = open("all-grades.txt", "r")
15
16     for line in infile.readlines():
17         grades = line.split(",")
18         total = 0
19         for grade in grades:
20             total = total + float(grade)
21         average = total / len(grades)
22         print("average =", average)
23     infile.close()
24
25 main()
26

all-averages.py 9:3 LF N UTF-8 Python 0 files
```

Example: Compute averages for multiple people - and save to file!

Solution 1: Save the results in a list and then print



```
all-averages-2.0.py -- ~/CS21/cs21-devel/examples/inclass/w07 -- Atom
File Edit View Selection Find Packages Help

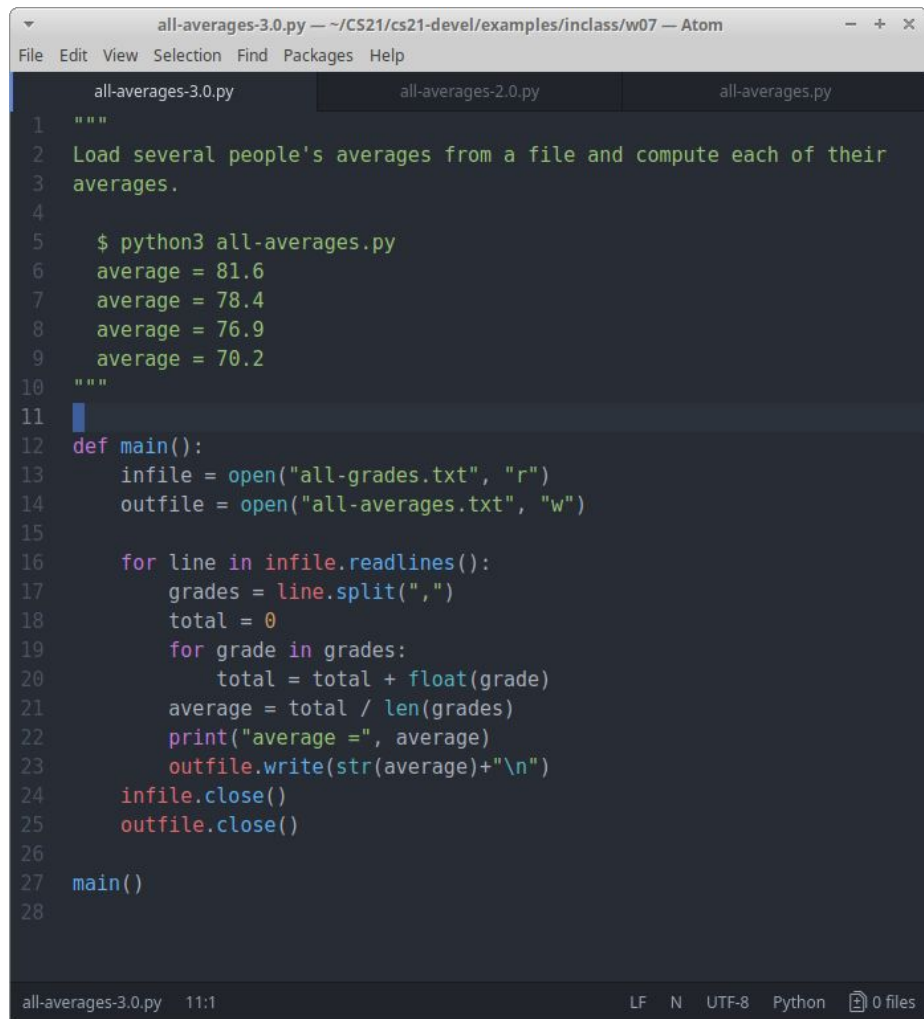
all-averages-3.0.py | all-averages-2.0.py | all-averages.py

1 """
2 Load several people's averages from a file and compute each of their
3 averages.
4
5 $ python3 all-averages.py
6 average = 81.6
7 average = 78.4
8 average = 76.9
9 average = 70.2
10 """
11
12 def main():
13     infile = open("all-grades.txt", "r")
14
15     averages = []
16     for line in infile.readlines():
17         grades = line.split(",")
18         total = 0
19         for grade in grades:
20             total = total + float(grade)
21         average = total / len(grades)
22         averages.append(average)
23     infile.close()
24
25     outfile = open("all-averages.txt", "w")
26     for average in averages:
27         print("average =", average)
28         outfile.write(str(average)+"\n")
29     outfile.close()
30
```

all-averages-2.0.py 13:41 LF N UTF-8 Python 0 files

Example: Compute averages for multiple people - and save to file!

Solution 2: Write the results right away



```
all-averages-3.0.py — ~/CS21/cs21-devel/examples/inclass/w07 — Atom
File Edit View Selection Find Packages Help

all-averages-3.0.py  all-averages-2.0.py  all-averages.py

1  """
2  Load several people's averages from a file and compute each of their
3  averages.
4
5  $ python3 all-averages.py
6  average = 81.6
7  average = 78.4
8  average = 76.9
9  average = 70.2
10 """
11
12 def main():
13     infile = open("all-grades.txt", "r")
14     outfile = open("all-averages.txt", "w")
15
16     for line in infile.readlines():
17         grades = line.split(",")
18         total = 0
19         for grade in grades:
20             total = total + float(grade)
21         average = total / len(grades)
22         print("average =", average)
23         outfile.write(str(average)+"\n")
24     infile.close()
25     outfile.close()
26
27 main()
28

all-averages-3.0.py  11:1  LF  N  UTF-8  Python  0 files
```

Splitting strings

```
listVariable = stringVariable.split(<delimiter>)
```

NOTE: Splitting strings removes the delimiter character and separates the stringVariable into sub strings!

```
>>> import string
>>> valueString = "40,50"
>>> valueList = valueString.split(",")
>>> valueList
['40', '50']
>>> phrase = "around-the-world"
>>> phrase.split("-")
['around', 'the', 'world']
>>> word = "banana"
>>> word.split("a")
['b', 'n', 'n', '']
```