Reading from Files
(it's magic!)

Three Steps to File Reading

- Any program that needs to read from a file will follow the same basic 3-step process.
- Example: Suppose we have a file with integers in it, and we'd like to have a program compute the sum of all the integers in the file.

Step 1: Open the file

Step 1: Open the file

- Uses the open( ) function.
- Always done the same way no matter how the file is organized.

```
file = open("filename.txt", "r")
```

open( ) returns a "file object," which is a data type like int, float, or string.

Replace this string with the real name of your file (don't forget the quotes!)

The "r" means open the file for reading.

Step 2: Loop over the file

Step 2: Loop over the file

```
file = open("filename.txt", "r")
for line in file:
```

`line` can be any string variable you want. This variable will store each line of the file as it is read.

Step 2: Loop over the file

```
file = open("filename.txt", "r")
while [there are more lines in the
       file that we haven't read]:
   line = [read the next line
           from the file]
```

- The "magic" for loop operates similarly to the pseudocode above.
- You only have access to one line of the file at a time.

Step 2: Loop over the file

- Inside the loop, do whatever you need to do with the string variable --- this will depend on the program you're trying to write.

```
file = open("filename.txt", "r")
for line in file:
  print(line) # Print the line
```

Step 2: Loop over the file

- Usually a good idea to "strip" the newline character from the line before processing:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  print(line)
```

Step 3: Close the file

Step 3: Close the file

- After you are done reading from the file, you should close the file:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  print(line)
file.close()
```

Complete file-reading loop

- Use this as a template for file reading:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  print(line) # Use for debugging
  [process the line]
file.close()
```

Reminders

- Each line of the file is always read as a **string**.
- If you want to process each line as an int or a float, you must do the conversion manually.

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  num = int(line)
  print("this is a number:", num)
```

`line` is a string variable (read by the magic for loop)

num is an int variable (converted through the **`int`** function)

- When initially writing code to read from a file, always print the lines from the file as you read them.
  - Incredibly helpful for debugging.
  - Remove the print part when you're convinced the program works otherwise.

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  # Keep this print stmt for debugging.
  print(line)
```

- The "magic" for loop stops automatically at the end of the file. It will never read past the end of the file.
- However, blank lines at the end of the file (often put there by accident) can cause weird errors.

You try

- In Python, create a new file with at least six two-digit integers (one integer per line).
- Save this file as `numbers.txt`.
- Write the basic file-reading loop to read in all the numbers and print them out as they're being read.
- Add code to add up all the numbers and print their total at the end.
- Add code to print their average at the end.
- Add code to print the differences between consecutive numbers as they're being read.
- Add code to print a "moving average" of the most recent three numbers in the file. (This means print the average of the numbers on lines 1, 2, and 3; then 2, 3, and 4; then 3, 4, and 5, etc)