
Homework Problems

10-1. Learning Python: Open a blank file in your text editor and write a few lines summarizing what you've learned about Python so far. Start each line with the phrase *In Python you can...* . Save the file as *learning_python.txt* in the same directory as this notebook. Write a program that reads the file and prints what you wrote three times. Print the contents once by reading in the entire file, once by looping over the file object, and once by storing the lines in a list and then working with them outside the `with` block.

```
In [1]: filename = 'learning_python.txt'

with open(filename, 'r') as f:
    text = f.read()
    print(text)

with open(filename, 'r') as f:
    for line in f:
        print(line.strip())

with open(filename, 'r') as f:
    lines = f.readlines()

for line in lines:
    print(line.strip())
```

```
In Python you can create dictionaries.
In Python you can generate random numbers.
In Python you can make the computer do math for you.
In Python you can create dictionaries.
In Python you can generate random numbers.
In Python you can make the computer do math for you.
In Python you can create dictionaries.
In Python you can generate random numbers.
In Python you can make the computer do math for you.
```

10-2. Learning C: You can use the `replace()` method to replace any word in a string with a different word. Here's a quick example showing how to replace 'dog' with 'cat' in a sentence:

```
message = "I really like dogs."
message.replace('dog','cat')
'I really like cats.'
```

Read in each line from the file you just created, *learning_python.txt*, and replace the word *Python* with the name of another language, such as *C*. Print each modified line to the screen.

```
In [2]: with open(filename, 'r') as f:
        lines = f.readlines()

        for line in lines:
            print(line.replace('Python', 'C').strip())
```

In C you can create dictionaries.
In C you can generate random numbers.
In C you can make the computer do math for you.

10-3. Guest: Write a program that prompts the user for their name. When they respond, write their name to a file called *guest.txt*

```
In [3]: name = input("What is your name? \n")

        with open('guest.txt', 'w') as f:
            f.write(name)
```

What is your name?
Andrew

10-4. Guest Book: Write a `while` loop that prompts users for their name. When they enter their name, print a greeting to the screen and add a line recording their visit in a file called *guest_book.txt*. Make sure each entry appears on a new line in the file.

```
In [4]: while True:
        name = input("What is your name (type 'q' to quit)? \n")

        if name == 'q':
            break

        print(f"Hello {name.title()}!")

        with open('guest_book.txt', 'a') as f:
            f.write(f"{name} \n") # make sure each entry is on a new line!
```

What is your name (type 'q' to quit)?
Andrew
Hello Andrew!
What is your name (type 'q' to quit)?
q

10-5. Programming Poll: Write a `while` loop that asks people why they like programming. Each time someone enters a reason, add their reason to a file that stores all the responses.

```
In [5]: while True:
        why = input("Why do you like programming (answer 'q' to quit)? \n")

        if why == 'q':
            break

        with open('why.txt', 'a') as f:
            f.write(f"{why} \n")
```

```
Why do you like programming (answer 'q' to quit)?
Because it's cool
Why do you like programming (answer 'q' to quit)?
Because I want to
Why do you like programming (answer 'q' to quit)?
q
```

10-6. Addition: One common problem when prompting for numerical input occurs when people provide text instead of numbers. When you try to convert the input to an `int`, you'll get a `ValueError`. Write a program that prompts for two numbers. Add them together and print the result. Catch the `ValueError` if either input value is not a number, and print a friendly error message. Test your program by entering two numbers and then by entering some text instead of a number.

```
In [6]: num1 = input("Please enter a number: \n")
        num2 = input("Please enter another number: \n")

        try:
            my_sum = int(num1) + int(num2)
        except ValueError:
            print("Sorry, you need to enter a number.")
        else:
            print(f"The sum is: {my_sum}")
```

```
Please enter a number:
1
Please enter another number:
two
Sorry, you need to enter a number.
```

10-7. Addition Calculator: Wrap your code from the exercise above in a `while` loop so the user can continue entering numbers even if they make a mistake and enter text instead of a number.

```
In [7]: while True:
        num1 = input("Please enter a number: \n")
        if num1 == 'q':
            break
        num2 = input("Please enter another number: \n")
        if num2 == 'q':
            break

        try:
            my_sum = int(num1) + int(num2)
        except ValueError:
            print("Sorry, you need to enter a number. \n")
        else:
            print(f"The sum is: {my_sum} \n")
```

```
Please enter a number:
1
Please enter another number:
two
Sorry, you need to enter a number.
```

```
Please enter a number:
1
Please enter another number:
2
The sum is: 3
```

```
Please enter a number:
q
```

10-8. Cats and Dogs: Make two files, *cats.txt* and *dogs.txt*. Store at least three names of cats in the first file and three names of dogs in the second file. Write a program that tries to read these files and print the contents of the file to the screen. Wrap your code in a `try-except` block to catch the `FileNotFoundError` error, and print a friendly message if a file is missing. Move one of the files to a different location on your system, and make sure the code in the `except` block executes properly.

```
In [8]: try:
        with open('cats.txt', 'r') as f:
            cats = f.read()
            print(cats)
        with open('dogs.txt', 'r') as f:
            dogs = f.read()
            print(dogs)
    except FileNotFoundError:
        print("Sorry, a file is missing.")
```

```
Charlie
Miku
George
Sorry, a file is missing.
```

10-9. Silent Cats and Dogs: Modify the `except` block from the exercise above to fail silently if either file is missing.

```
In [9]: try:
        with open('cats.txt', 'r') as f:
            cats = f.read()
            print(cats)
        with open('dogs.txt', 'r') as f:
            dogs = f.read()
            print(dogs)
    except FileNotFoundError:
        None
```

Charlie
Miku
George

10-10. Common Words: Visit Project Gutenberg (<https://gutenberg.org/>) and download a text file of a work you like (you can copy the raw text from the browser into a textfile). You can use the `count()` method to find out how many times a word or phrase appears in a string. For example, the following code counts the number of times 'row' appears in a string:

```
>>> line = "Row, row, row your boat"
>>> line.count('row')
2
>>> line.lower().count('row')
3
```

Notice that converting the string to lowercase using `lower()` catches all appearances of the word you're looking for, regardless of how it's formatted.

Write a program that reads the file you found at Project Gutenberg and determines how many times the word 'the' appears in the text. This will be an approximation because it will also count words such as 'then' and 'there'. Try counting 'the ' with a space in the string, and see how much lower your count is.

```
In [10]: with open('alice.txt', 'r', encoding='utf-8') as f:
        text = f.read()
        print(text.lower().count('the '))
```

1695

10-11. Favorite Number: Write a program that prompts the for the user's favorite number. Use `json.dump()` to store this number in a file. Write a separate program that reads in this value and prints the message, "I know your favorite number! It's ____."

```
In [11]: # prompt for a favorite number
import json

num = input("Please type in your favorite number: \n")

with open('favorite_number.txt', 'w') as f:
    json.dump(num, f)
```

Please type in your favorite number:
21

```
In [12]: # display the stored favorite number
with open('favorite_number.txt', 'r') as f:
    fav_num = json.load(f)
    print(f"I know you favorite number! It's {fav_num}.")
```

I know you favorite number! It's 21.

10-12. Favorite Number Remembered: Combine the two programs for the exercise above into a single program. If the number is already stored, report the favorite number to the user. If not, prompt for the users' favorite number and store it in a file. Run the program twice to see that it works.

```
In [13]: import json

try:
    with open('favorite_number.txt', 'r') as f:
        fav_num = json.load(f)
        print(f"I know you favorite number! It's {fav_num}.")
except FileNotFoundError:
    num = input("Please type in your favorite number: \n")

    with open('favorite_number.txt', 'w') as f:
        json.dump(num, f)
```

I know you favorite number! It's 21.

10-13. Verify User: The final listing for *remember_me.py* assumes either that the user has already entered their username or that the program is running for the first time. We should modify it in case the current user is not the person who last used the program.

Before printing a welcome back message in `greet_user()`, ask the user if this is the correct username. If it's not, call `get_new_username()` to get the correct username.

```
In [14]: from pathlib import Path
import json

def get_stored_username(path):
    """Get stored username if available."""
    if path.exists():
        contents = path.read_text()
        username = json.loads(contents)
        return username
    else:
        return None

def get_new_username(path):
    """Prompt for a new username."""
    username = input("What is your name? ")
    contents = json.dumps(username)
    path.write_text(contents)
    return username

def greet_user():
    """Greet the user by name."""
    path = Path('username.json')
    username = get_stored_username(path)

    if not username:
        username = get_new_username(path)
        print(f"We'll remember you when you come back, {username}!")
        return None

    correct = input(f"Is this the correct username (yes or no)? {username} \n")
    if correct == 'yes':
        print(f>Welcome back, {username}!")
    else:
        username = get_new_username(path)
        print(f"We'll remember you when you come back, {username}!")

greet_user()
```

Is this the correct username (yes or no)? Andrew Bernas
yes
Welcome back, Andrew Bernas!

In []: