15-112, Spring 2015

Assignment 4

Submission

You must submit one ZIP file on Autolab. This ZIP file must be named *userid*-hw4.zip and must include all of the Python files that you have written for this assignment. Please write your name and userid as a comment at the beginning of each Python file.

Recommendations

- **Make sure to test your program**.
- Make sure that your program is executable. If you are unable to complete portions of the assignment, comment out the part of the code that does not work properly, and explain what you did, what worked, and what did not. It is your responsibility to explain as carefully as you can why you think you were unable to get the code working, what you think is wrong, and how you might go about fixing it. The quality of such an explanation will be important to us in deciding whether to give you partial credit.

Tip 1: the instruction `find`

The function `find(text, word)` takes a `text`, a `word` (both as `string`) and returns the index (the position) where `word` appears first in `text`. It returns `-1` if the word is not found.

To get familiar with this instruction, try this snippet of code:

```
def find(text,word):
    return text.find(word)

print find("I think this article is well written!","article")
print find("I do not understand what you are talking about","blue")
```

Tip 2: double quotes in text

In python, a string value is a text delimited by double quotes. However, how can we have a text that contains double quotes without having this quotes interpreted as delimiters by python?

For instance, let us imagine that we want to store `Hello "Thierry", how are you?` in a variable. The following code is incorrect

```
blahblah = "Hello "Thierry", how are you?"

print blahblah
```

because Python thinks that you are giving the string value `"Hello "` followed by incorrect code. To avoid this problem, we should let Python know that the quotes between `"Thierry"` are **not** delimiters but normal text. To do that, we will use what we called *an escape character* which is `/` in Python. This is a correct code:

```
blahblah = "Hello \"Thierry\", how are you?"

print blahblah
```

Goal

The news website *Q-news* wants to allow people posting comments on their news articles. However, they do want to have:

- comments with bad language.
- comments with SMS language
- comments that are too long

They are asking for your help to write a program that will automatically format the comments and display it on the webpage.

1 Someone said [10 points]

To post a comment, people should fill an online form with their name and their comment. After submitting the form, the webpage shows a box with the person's name and comment.

Write a function `someoneSaid(comment, name)` that takes a `comment`, a `name` (both as `string`) and returns the `comment` between double quotes preceded by the `name` of the person who said this comment.

- `comment` is the comment.
- `name` is name of the person posting the comment.

For instance,

```
someoneSaid("This assignment is easy", "Thierry")
```

should returns

```
"Thierry said "This assignment is easy""
```


2 Maximum [10 points]

Q-news wants to limit the size of the comments. If the comment exceed a certain limit, it should be truncated.

Write a function `maxLength(comment, max)` that takes a `comment` (as `string`), a `max` (both as `number`) and returns the `comment` truncated if it exceeds the maximum length `max`.

- `comment` is the comment.
- `max` is maximum length of the comment.

For instance,

```
maxLength("I think that this assignment is awesome!", 30)
```

should returns

```
"I think that this assignment i"
```

3 The Word Filter [20 points]

Q-news does not want bad language to appear in the comments. They want to filter well known bad words automatically.

3.1 `filter`

Write a function `filter(comment, word)` that takes a `comment`, a `word` (both as `string`) and returns the `comment` with all occurrences of `word` deleted.

- `comment` is the comment.
- `word` is the word to be removed.

For instance, assuming that word `duck` is bad language,

```
filter("What the duck is that duck?", "duck")
```

should returns

```
"What the is that?"
```


4 The Word Cleaner [20 points]

Q-news does not want SML language in the comments. They want to automatically replace well known SML abbreviations with the correct word.

4.1 `cleaner`

Write a function `cleaner(comment, abbreviation, word)` that takes a `comment`, an `abbreviation` and a `word` (all as `string`)) and returns the `comment` with all occurrences of `abbreviation` replaced by the corresponding `word`.

- `comment` is the comment.
- `abbreviation` is the text to be replaced.
- `word` is the word that replaces all occurrences of the abbreviation.

For instance,

```
cleaner("What do u think?  Do u agree?", "u", "you")
```

should returns

```
"What do you think?  Do you agree?"
```

5 All Together [20 points]

5.1 `postComment`

Write a function `postComment(comment,name)` that takes a `comment`, a `name` (both as `string`)) and returns the `comment` with well known bad word deleted and well known abbreviations replaced.

- `comment` is the comment.
- `name` is the name of the person.

Instruction 1: The bad words to be deleted are:

- "duck"
- "shot"

Instruction 2: The abbreviations to be replaced are:

- `"u"` replaced by `"you"`
- `"r"` replaced by `"are"`
- `"4"` replaced by `"for"`
- `"1drfl"` replaced by `"wonderful"`
- `"LMAO"` replaced by `"Funny!"`

Instruction 3: The length of the comment should not exceed 150 characters after having deleted the bad words and replaced the abbreviations.

Instruction 4: The comment should be put between double quotes and preceded by the the name of the person who said this comment as done in exercise 1.

6 Multiple Filters and Cleaners [20 points]

Write a function `betterComment(comment,name,badWords,abbreviations)` that takes a `comment`, a `name`, a `badWordFilter`, an `abbreviationFilter` (all as `string`)) and returns the `comment` with all bad words specified in `badWords` deleted and all abbreviations specified in `abbreviations` replaced.

- `comment` is the comment.
- `name` is the name of the person.
- `badWordFilter` is the collection of bad words to be deleted. These words are separated by spaces.
- `abbreviationFilter` is the collection of abbreviations and their replacements. The abbreviation and the replacement are given together separated by a "/" and the pairs of abbreviation/replacement are separated by spaces.

For instance, one correct call to this function is:

```
betterComment("What the duck?  This is so stupid!  Do u think ur opinion is better than
mine?", "stupid duck shot", "John", "ur/your u/you r/are 4/for LAMAO/funny!")
```

