

Homework 5: Mad Libs[®], More or Less

Assigned: Friday, February 2, 2007

Due: Tuesday, February 6, 2007

No extensions!

Summary: In this assignment, you will use what we've learned about strings to build a simple game of Mad Libs[®]

Purposes: To give you experience with string operations. To help you consider the ways in which you can use multiple files. To have a bit of fun.

Expected Time: One hour.

Collaboration: Although you may work alone on this assignment, I would prefer that you work in groups of size two, three, or four. You need only submit the assignment once for each group. You may also discuss the assignment with anyone you wish, but please cite those outside your group with whom you talk.

Submitting: Email me your files as attachments, using a subject of *CSC151 Homework 5*. Add a sample run and info on group members and citations. More details below.

Warning: So that this exercise is a learning assignment for everyone, I may spend class time publicly critiquing your work. I will also post all answers in a public directory and look to combine work from different groups.

Background

You may be familiar with the game *Mad Libs*[®]. In this game of two or more players, one player serves as the questioner and the remaining players provide answers. The questioner works from a text in which certain words are missing and asks the players to submit words for those that are missing. The only detail that the questioner provides is the part of speech (or other description of the expected word). The questioner then reads the resulting text.

Consider the following text:

John likes to *_verb_* with Scheme
because he thinks that Scheme is *_adjective_*.
When things go well, John is *_adjective_*.
When things go badly, John yells "*_exclamation_!*"

In this case, the questioner would ask for a verb, two adjectives, and an exclamation. If the players provided the words “sleep”, “wicked”, “purple”, and “excelsior”, the questioner would then read

John likes to sleep with Scheme
because he thinks that Scheme is wicked.
When things go well, John is purple.
When things go badly, John yells "excelsior!"

Assignment

As you might guess, the form letter technique we studied in the lab on characters and strings is well suited to the game of Mad Libs[®]. We will explore it in this assignment.

Of course, we do not yet have the capability to interact with the user (that is, to query for words and to get responses). Hence, your Mad Lib[®] will be a bit simpler. Your primary assignment is to write a Scheme definition of the form from the lab on characters and strings that incorporates a variety of previously named values.

In particular, assume that someone has defined the following values as lists of strings strings that match the requirements of the name. (E.g., each member of `adjectives` should be a string, and not only should each member of `pronouns` be a string, but each value should be the pronoun for the corresponding value of `people`.)

- `people`, a list of two strings;
- `pronouns`, a list of two strings;
- `transitive-verbs`, a list of three strings;
- `intransitive-verbs`, a list of two strings;
- `nouns`, a list of three strings;
- `adjectives`, a list of five strings;
- `adverbs`, a list of two strings;
- `places`, a list of two strings; and
- `exclamations`, a list of two strings.

You should also assume that a list of three numbers, `numbers` is defined. (All three values in that list should be numbers, not strings.)

Part A: The Words File

Create a file, `madwords.ss` that defines all of the words given above. Here is a template to get you started.

```
(define people (list "---" "---"))
(define pronouns (list "---" "---"))
(define transitive-verbs (list "---" "---" "---"))
(define intransitive-verbs (list "---" "---"))
(define nouns (list "---" "---" "---"))
(define adjectives (list "---" "---" "---" "---" "---"))
(define adverbs (list "---" "---"))
(define places (list "---" "---"))
(define exclamations (list "---" "---"))
(define numbers (list 0 0 0))
```

Part B. The MadLibs Expression

Create a file, `madlibs.ss` that defines the value `madlibs` as a string built by appending together some fixed strings and most of the strings from the variables in part A. Here's a really short definition (much shorter than yours should be).

```
(define madlibs
  (string-append
    "I hear that " (car people) " " (cadr transitive-verbs)
    " "(caddr nouns) ". "
    "Whenever " (car people) " sees a " (caddr nouns) ", "
    (car pronouns) " says \" " (car exclamations) "!\""))
```

Part C. Testing Your Code

Once you are happy with your definitions (of the basic words and the madlibs), run the following.

```
(load "madwords.ss")
(load "madlibs.ss")
(display madlibs)
```

Important Evaluation Criteria

Particularly amusing, clever, or elegant solutions may earn extra credit.

Submitting Your Homework

We'll be sharing your word definitions and your Mad Libs[®], so I'd like both files. Please send me an email message with a subject of *CSC151 Homework 5* and include `madwords.ss` and `madlibs.ss` as attachments. In the body of the message, include (a) a list of your group members, (b) the text of the result of your Mad Libs[®] game, and (c) any appropriate citations.

Some Advice

If you don't feel creative enough to write your own Mad Libs[®] paragraph, consider taking a paragraph from a piece of public-domain literature, identifying some parts of speech in that paragraph, and replacing those parts of speech with the variables above.

Copyright © 2007 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.