

Laboratory: CGI Scripting

Summary: This lab reviews some mechanics related to the use of Scheme for CGI Scripts in CSC151.

Useful Files:

- `greeting.ss`
- `greeting.html`

Note: You may find that the formatting of this lab is a little bit off. I haven't had time to fix that formatting.

Contents:

- -
 -
 -
 - Exercise 7: Hidden Values

Exercises

Exercise 0: Preparation

- Make sure you've done the reading on CGI.
- Make copies of `greeting.ss` and `greeting.html` in your `public_html` directory.
- Create (using gnotepad or DrScheme) the file `greeting.cgi` in your `public_html` directory that contains the following two lines

```
#!/bin/bash
/home/rebelsky/bin/schemeweb greeting.ss
```

The lines must be flush against the left margin!

- Share all three files.
- Make `greeting.cgi` executable by typing the following in your terminal window:

```
% chmod a+x greeting.cgi
```

Don't type the percent sign.

Exercise 1: Testing within DrScheme

- a. Load `greeting.ss` from within DrScheme.
- b. Execute `greeting.ss`.
- c. Type `(page)` and see what happens.
- d. Find the code that defines `user` and change the string "Sam" to your name.
- e. Type `(page)` and see what happens.
- f. Type `(greeting-page "Fred")` and see what happens.

Exercise 2: Testing From the Web

Things may go seriously wrong in this step. Feel free to ask lots of questions.

- a. In a Web browser, open `<http://www.cs.grinnell.edu/~yourid/greeting.html>`.
- b. Enter **XXX** and click the button.
- c. What happens? Can you tell why?

Exercise 3: Responding to Input

Things may go seriously wrong in this step. Feel free to ask lots of questions.

- a. Within DrScheme, change the line in `greeting.ss` that reads `(define testing #t)` to `(define testing #f)`.
- b. What happens when you try to execute the program and run `(page)` from within DrScheme? (Yes, you may get an error.)
- c. What happens when you repeat the steps of the previous exercise?

Exercise 4: Inserting Errors

- a. Reset the value of `testing` to `#t`.
- b. Change the body of `page` to read `(greeting-page person)`.
- c. What do you expect to happen when you execute your modified code and then evaluate `(page)` within DrScheme?
- d. Verify your answer experimentally.

- e. What do you expect to happen when you try the Web interface to your code (that is, when you enter values on the HTML page and then click the button)?
- f. Verify your answer experimentally.
- g. Reset the value of testing to #f.
- h. What do you expect to happen when you try the Web interface to your code (that is, when you enter values on the HTML page and then click the button)?
- i. Verify your answer experimentally.
- j. Fix the error you introduced in step b. above.

Exercise 5: Getting More Input

- a. Add a text field to the Web page that prompts for a color.
- b. See what happens when someone fills in the color (presumably nothing).
- c. Update `greeting.ss` to store the choice in a variable. (Hint: Look at what I did for `user`.)
- d. Update the `greeting-page` procedure to take the selected color as a second parameter. You must include some comment about the color in the output from `greeting-page`.
- e. Verify within DrScheme that the new `greeting-page` still works.
- f. Update `page` so that it passes both `user` and the color on to `greeting-page`.
- g. Test from within your Web browser.

Exercise 6: Radio Buttons

As you may have observed in your experiences on the Web, there are many different kinds of input device on forms. One popular input device is the so-called *radio button* in which someone selects between a few options by clicking to the left of an option.

You can create those buttons with

```
<input type="radio" name="XXX" value="Val1">Text  
<input type="radio" name="XXX" value="Val2">Text  
<input type="radio" name="XXX" value="Val3">Text
```

Each radio button in a group must have the same name.

For example, here are a collection of buttons that let someone indicate what their status is at Grinnell. Note that the value specified does not need to match the text.

```
<input type="radio" name="status" value="5">President
<input type="radio" name="status" value="4">Administrator
<input type="radio" name="status" value="3">Student
<input type="radio" name="status" value="2">Staff
<input type="radio" name="status" value="1">Faculty
```

Here's how those buttons appear in a browser:

◇ President ◇ Administrator ◇ Student ◇ Staff ◇ Faculty

- a. Replace the color field in `greeting.html` with some radio buttons.
- b. Verify that the CGI script still works with those buttons.

Exercise 7: Hidden Values

You can also encode "hidden" values in your forms. These are values that you want to pass on to the CGI script but that you don't want the user to change.

- a. Replace the line that reads

```
<input type="text" name="user" value="">
```

with one that reads

```
<input type="hidden" name="user" value="John Doe">
```

- b. What effect do you expect this change to have?
- c. Verify your results experimentally.