

## Assignment 4

### *Python Web Applications*

**Purposes:** Extend our explorations beyond Lab 3, and begin to build interactive graphical web applications in Python.

**Due dates and partnerships:** Problem 2, parts a, b, and c are due Friday, February 20, before 5:00 PM. Problem 1 is to be done as "individual" work (not group work), and each student must do it independently and turn in a separate document. It is due Monday, February 23 before 5:00 PM. Problem 2 (all parts) is a partnership problem, and partnerships are optional. You may either do Problem 2 individually or with a partner. Parts a, b, and c of Problem 2 are due on Friday, February 20 before 5:00 PM (as already mentioned), and the rest of Problem 2 is due on Wednesday, February 25 before 5:00 PM.

Problem 1. (Individual work, turned in separately.)

Port your "Guess my number" game (from Assignment 3) to the web.

a. Use Brython as your Python language processor, as in Lab. 3.

b. Instead of using the `raw_input` function, as in Assignment 3, obtain user input from an HTML Input field and a Button. The prompt for the user should be HTML text that is part of a Paragraph object on the page, which in turn, is within a DIV object in your page.

c. As the game proceeds, display the history of the game (questions asked, answers given) as new Paragraph items within the DIV. (How to add to the DOM was shown in Lab. 3.)

d. Add styling to the above "history" text, such that questions are shown in blue text, negative answers are shown in red, positive answers are shown in green, and warnings about improper questions, etc., are shown in orange. This styling can be accomplished by setting "style" properties of the paragraphs in the DOM.

Problem 2. (Optionally done in partnerships of 2.)

Design and implement a Clock WebApp.

For this problem, you should examine a variety of different clock programs online, sketch your own unique clock design, and then start figuring out how to implement it using Brython. One good example of an implementation is this:

<http://www.brython.info/gallery/clock.html>

The code for this shows you how to deal with the showing of time as well as the use of Canvas graphics in Brython.

Your design could be for a digital clock, a bar-graph clock, a simulated water clock, etc. Originality is good!

a. Draw a picture of your clock design, using paper and pencil. Both partners (if you are in a partnership) should sign the drawing. Please scan it and turn it in via Catalyst CollectIt. Otherwise, hand in hardcopy in class on Friday, Feb. 20.

b. Make a list of its time-keeping and other features. For example, it might show the day of the month (as in the Brython clock example). It could also show the day of the week (e.g., "Monday"), the name of the month, etc. You might have an alarm function (settable?) or a Big-Ben bell-ringing feature, etc.

c. Explain how you might go about implementing your clock program: what code would you try to create first?

d. Implement and debug some version of your clock.

e. Put your clock onto the students web server and get it working there.

f. Turn in the URL to your clock's html page with your A4 turn-in.

Turn in your work by each of the corresponding deadlines:

Problem 2, parts a, b, and c: 5:00 PM, Friday, Feb. 20. (only the partner having the last name that comes first in the alphabet need take care of the turn-in.)

Problem 1 (individual work). Monday, Feb. 23, by 5:00 PM.

Problem 2, parts d, e, and f. Wednesday, Feb. 25, by 5:00 PM. (one turn-in per partnership. This should be turned in by the same partner who will have turned in parts a, b, and c.)

Submit the documents electronically that contains your answers to the Assignment 4 DropBoxes at Catalyst CollectIt. <https://catalyst.uw.edu/collectit/dropbox/tanimoto/34051>