Banners

psp-06-03

1 Overview

A publisher has asked you to develop software to create clever text banners using ascii art. For example the word 'hello' could be rendered:

```
#       #   #
### ### #   #   ###
# # ##  #   #   # #
# # ### ##  ##  ###
```

2 Learning Outcomes

By the end of this project students should be able to:

- read and write programs that define simple functions;
- read and write programs that effectively use lists and dictionaries;
- read and write programs that use comparison operators;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

- Read Problem Space Chapter 6: Collections.
- Read the full lab instructions and plan your strategy for representing each letter. Draw your letter designs on paper. You only have to do one case (upper or lower). Be prepared to show your designs to the lab aide at the beginning of class.

4 Lab Instructions

Do this part in lab:

You should implement a function called "print_banner" that takes a string input for what to print and a second input to signify if the banner is horizontal or vertical. This lab will be easy if you use lists and dictionaries effectively and tedious if you don't! The font size and design is up to you (although four is the minimum height).

When you have completed the lab run pep8 against your code until all formatting errors have been corrected and your code is PEP 8 compliant. See the Getting Started lab if you need instructions on running the program, or the pep8 documentation found here.

5 Lab Report

For each pair of students one lab report (the same one) will be turned in by each student on BBLearn.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in **your** own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process..

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code
- a screen shot of your application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

Colophon

This project was developed by Dr. James Dean Palmer of Northern Arizona University.