Python Activity: LISTS
"Hangman Game"

Learning Objectives

Students will be able to:

- Demonstrate their applied knowledge of IF statements, Loops, and Lists
- Write code using the Python syntax for IF statements, Loops, and Lists
- Demonstrate debugging skills
- Interpret pre-existing code and make additions/edits

Hangman is a classic guessing game for two or more players. One player thinks of a word and the other(s) tries to guess it by suggesting letters, within a certain number of guesses.

How can we take that idea and write a functional computer program around it?

First, we'll need to create the word that will act as our "answer". Then, we'll need to ask the user to guess a letter. Then, we will need to check to see if that letter is included in the "answer" word.

These last two steps will need to be repeated (in a loop) until all of the letters are successfully guessed.

In Python, the following is code for a functioning Hangman game. Make sure to read the comments to understand what is happening in each part of the code.

```
# Hangman game!

# Assume the answer is "hangman"

A = ['h','a','n','g','m','a','n']
L = ['_','_','_','_','_','_','_']

play = True

while play == True:

        # Ask the user to guess a letter

        letter = str(input("Guess a letter: "))

        # Check to see if that letter is in the Answer

        i = 0
        for currentletter in A:
```

```
                # If the letter the user guessed is found in the answer,
                # set the underscore in the user's answer to that letter

                if letter == currentletter:

                        L[i] = letter

                i = i + 1

        # Display what the player has thus far (L) with a space
        # separating each letter

        print(' '.join(str(n) for n in L))

        # Test to see if the word has been successfully completed,
        # and if so, end the loop

        if A == L:

                play = False

print("GREAT JOB!")
```

1. Type the code above, save the file with a .py extension, and run the program in the Terminal. If you receive any error messages, determine what the problems are likely to be and debug your code.

2. Once you can run the program with no error messages, try it out. Does it work? How do you know if it is working correctly?

3. ASSIGNMENT: Write code to make this program react to INCORRECT letters by printing "BAD GUESS!".

4. ASSIGNMENT: In Hangman, the user only gets 6 incorrect guesses before they lose and the game is over. Write code to implement this.

5. ASSIGNMENT: Rather than hard-coding the answer as "hangman", randomly select a word from a LIST of words to use as the answer.

6. CHALLENGE ASSIGNMENT: Rather than hard-coding the answer as "hangman", randomly select a word from an EXTERNAL FILE to use as the answer.

7. CHALLENGE ASSIGNMENT: Use a Python library to access a dictionary on the internet, and use a randomly selected word from that as the answer.

8. CHALLENGE ASSIGNMENT: Create a visualization for the game using the Turtle library.