

Python – Programming Assignment 4 – Advanced

Professor:

Dr. Joanna Bieri
joanna_bieri@redlands.edu

Directions: You will build a word guessing game! (Hangman)

Specifications:

- - Your code should have a dictionary of secret words defined. Use the secret word as the key and then provide hints for the word so that when you call `my_words['word']` it returns a list of clues. The longer the better
 - Give a prompt that asks the user if they want to play the game and explain the rules.
 - If they say yes then
 - Have your code randomly choose a word from the secret word dictionary keys and randomly print one of the clues.
 - Create a guessed letter list that matches the length of the secret word and allows you to fill in correct letters
- ```
In: my_string = 'hello'
letter_list = list(my_string)
print(letter_list)

Out: ['h', 'e', 'l', 'l', 'o']

In: correct_letters = []
for ch in letter_list:
 correct_letters.append('--')

print(correct_letters)

Out: ['--', '--', '--', '--', '--']
```
- Now using the while loop allow the user to guess letters. Tell them if they get the letter wrong. If they get the letter right then update the `correct_letters` list with the letters in the correct spots.
- ```
my word -> ['h', 'e', 'l', 'l', 'o']

If user guesses 'l'

['--', '--', 'l', 'l', '--']
```
- Keep track of the number of guesses and give a limit (you choose).
 - Keep track of the letters they have guessed.
 - Let the player quit if they want to.
- Print a message with the result of the game
 - Your code should be a .py file
 - Your code should run without errors
 - Your code should use good python style - cleaned up, commented, etc.
 - Your code should be submitted on time.

How can I know if my code is working? Try out the game with a list of only one possible secret word. Then see what your code does if you guess the same letter twice, or quit, or get a letter right, etc.
