Python Script Tutorial – Hangman Game
Python Workshop, University of Cincinnati Libraries
Amal Chaturvedi, Sean Crowe, James Van Mil

Create an interactive script based on the popular grade-school game, Hangman.

Constraints

- Accept multiple words, select one word randomly for game
- Accept only single-character guesses
- Allow only 5 incorrect guesses, player not penalized for correct guesses
- All input words should be lower-case

1. Add a comment header for the script and import random module.

```python
#hangman.py
import random
```

2. Create an empty list to store submitted guesses.

```python
words = []
```

3. Prompt user for secret words using raw_input built-in method. Repeat the prompt 3 times using a for loop with range method.

```python
for x in range(0, 3):
    words.append(raw_input('Enter a word for our game: '))
```

4. Randomly select a word from list of words using the random module.

```python
word = random.choice(words)
```

5. Create empty string variable to store guessed letters.

```python
guessed_letters = ''
```

6. Set count variable to track remaining guesses (set to 5).

```python
remaining_guesses_count = 5
```

7. Initialize while loop and continue as long as count is greater than zero. Use raw_input to get guess from user. Use lower() to make all submitted letters lowercase.

```python
while remaining_guesses_count > 0:
    guess = raw_input('

You have ' + str(remaining_guesses_count) + ' guesses left, guess a letter: ').lower()
```
Within while loop:

8. For each pass in loop, add single guess to string of collective guesses.
   ```python
guessed_letters += guess
```

9. Track remaining hidden letters with integer variable. For every pass, set to zero. When the guess is evaluated further down, the number of un-guessed letters will be counted and set to this variable.
   ```python
unguessed_letters_count = 0
```

10. Add for loop to check guesses against secret word and output word with mixed characters: correctly guess characters are revealed, un-guessed characters are replaced with asterisk.
   ```python
   for letter in word:
       if letter in guessed_letters: # reveal the letter if guessed
           print letter,
       else:
           # print asterisk for letters that haven't been guessed
           print '*',
   ```

11. Under else statement, add 1 to unguessed_letters variable for every letter that remains un-guessed. This results in 1 being added to the variable for every un-guessed letter.
   ```python
   for letter in word:
       if letter in guesses: # reveal the letter if guessed
           print letter,
       else:
           # print asterisk for letters that haven't been guessed
           print '*',
           unguessed_letters_count += 1
   ```

12. Check if all letters have been guessed, if so, display winning message and break.
    ```python
    if unguessed_letters_count == 0:
        print ''\n\n\t\a\aYou won!!!!!!\n\n' 
        break
    ```

13. Check if guessed letter is in word, if not, reduce remaining guesses by 1.
    ```python
    if guess not in word:
        remaining_guesses_count -= 1
    ```

14. Check remaining guesses, if zero, display losing message and break.
    ```python
    if remaining_guesses_count == 0:
        print ''\n\n\tNice try, chump\n\n'
Script with comments

#hangman.py

import random

words = []

#prompt user for secret word, assign to variable
for x in range(0,3):
    words.append(raw_input('Enter a word for our game: '))

#randomly select a word from the list
word = random.choice(words)

guessed_letters = ''
remaining_guesses_count = 5

#initialize loop
while remaining_guesses_count > 0:  #continue loop while count is greater than zero
    guess = raw_input('You have ' + str(remaining_guesses_count) + ' guesses left, guess a letter: ')  #keyboard input for guess, assign single guess to string variable, display remaining guess count
    guessed_letters += guess

    unguessed_letters_count = 0
    for letter in word:
        if letter in guessed_letters:  #reveal the letter if guessed
            print letter,
        else:
            print '*',  #print asterisk for letters that haven't been guessed
            unguessed_letters_count += 1

    if unguessed_letters_count == 0:
        print 'You won!!!!!
        break

    if guess not in word:
        remaining_guesses_count -= 1

    if remaining_guesses_count == 0:
        print 'Nice try, chump
        break