On Thursday, February 16th at 6:00AM EST, UTC-5, we will be conducting a brief database maintenance. The event should last about 5 minutes.

MITx: 6.00.1x Introduction to Computer Science and Programming U..

Help

Bookmarks

- Welcome to the edX Platform
- Entrance Survey
- Download Python and Get Motivated!
- Week 1: Python Basics
- Week 2: Simple Programs
- **Week 3: Structured Types**
- Week 4: Good Programming Practices

7. Testing and Debugging
Finger Exercises

8. Exceptions and Assertions
Finger Exercises

Problem Set 4

Week 3: Structured Types > Problem Set 3 > Problem 4

Problem 4

Bookmark this page

Problem 4 - The Game

15.0 points possible (graded)

Now you will implement the function `hangman`, which takes one parameter - the `secretWord` the user is to guess. This starts up an interactive game of Hangman between the user and the computer. Be sure you take advantage of the three helper functions, `isWordGuessed`, `getGuessedWord`, and `getAvailableLetters`, that you've defined in the previous part.

Hints:

- You should start by noticing where we're using the provided functions (at the top of `ps3_hangman.py`) to load the words and pick a random one. Note that the functions `loadWords` and `chooseWord` should only be used on your local machine, not in the tutor. When you enter in your solution in the tutor, you only need to give your `hangman` function.

- Consider using `lower()` to convert user input to lower case. For example:

```
guess = 'A'
guessInLowerCase = guess.lower()
```

- Consider writing additional helper functions if you need them!

- There are four important pieces of information you may wish to store:

  1. `secretWord`: The word to guess.
  2. `lettersGuessed`: The letters that have been guessed so far.

Problem Set due Feb 23, 2017 15:30 PST

- Midterm Exam
- Week 5: Object Oriented Programming
- Sandbox

3. `mistakesMade` : The number of incorrect guesses made so far.
4. `availableLetters` : The letters that may still be guessed. Every time a player guesses a letter, the guessed letter must be removed from `availableLetters` (and if they guess a letter that is not in `availableLetters` , you should print a message telling them they've already guessed that - so try again!).

Sample Output

The output of a winning game should look like this...

```
Loading word list from file...
55900 words loaded.
Welcome to the game, Hangman!
I am thinking of a word that is 4 letters long.
-------------
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Good guess: _ a_ _
------------
You have 8 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! You've already guessed that letter: _ a_ _
------------
You have 8 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: s
Oops! That letter is not in my word: _ a_ _
------------
You have 7 guesses left.
Available letters: bcdefghijklmnopqrtuvwxyz
Please guess a letter: t
Good guess: ta_ t
------------
You have 7 guesses left.
Available letters: bcdefghijklmnopqruvwxyz
Please guess a letter: r
Oops! That letter is not in my word: ta_ t
------------
You have 6 guesses left.
Available letters: bcdefghijklmnopquvwxyz
Please guess a letter: m
Oops! That letter is not in my word: ta_ t
------------
You have 5 guesses left.
Available letters: bcdefghijklnopquvwxyz
Please guess a letter: c
Good guess: tact
------------
Congratulations, you won!
```

And the output of a losing game should look like this...

```
Loading word list from file...
55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 4 letters long.
-----------
You have 8 guesses left.
Available Letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! That letter is not in my word: _ _ _ _
-----------
You have 7 guesses left.
Available Letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: b
Oops! That letter is not in my word: _ _ _ _
-----------
You have 6 guesses left.
Available Letters: cdefghijklmnopqrstuvwxyz
Please guess a letter: c
Oops! That letter is not in my word: _ _ _ _
-----------
You have 5 guesses left.
Available Letters: defghijklmnopqrstuvwxyz
Please guess a letter: d
Oops! That letter is not in my word: _ _ _ _
-----------
You have 4 guesses left.
Available Letters: efghijklmnopqrstuvwxyz
Please guess a letter: e
Good guess: e_ _ e
-----------
You have 4 guesses left.
Available Letters: fghijklmnopqrstuvwxyz
Please guess a letter: f
Oops! That letter is not in my word: e_ _ e
-----------
You have 3 guesses left.
Available Letters: ghijklmnopqrstuvwxyz
Please guess a letter: g
Oops! That letter is not in my word: e_ _ e
-----------
You have 2 guesses left.
Available Letters: hijklmnopqrstuvwxyz
Please guess a letter: h
Oops! That letter is not in my word: e_ _ e
-----------
You have 1 guesses left.
Available Letters: ijklmnopqrstuvwxyz
Please guess a letter: i
```

```
Oops! That letter is not in my word: e_ _ e
-----------
Sorry, you ran out of guesses. The word was else.
```

Note that if you choose to use the helper functions `isWordGuessed`, `getGuessedWord`, or `getAvailableLetters`, you do not need to paste your definitions in the box. We have supplied our implementations of these functions for your use in this part of the problem. If you use additional helper functions, you will need to paste those definitions here.

Your function should include calls to `input` to get the user's guess.

Why does my Output Have `None` at Various Places?

```
    * At the start of the game, let the user know how many
      letters the secretWord contains.

    * Ask the user to supply one guess (i.e. letter) per round

    * The user should receive feedback immediately after each
      about whether their guess appears in the computers word.

    * After each round, you should also display to the user th
      partially guessed word so far, as well as letters that t
      user has not yet guessed.

    Follows the other limitations detailed in the problem writ
    '''
    # FILL IN YOUR CODE HERE...

```

Press ESC then TAB or click outside of the code editor to exit

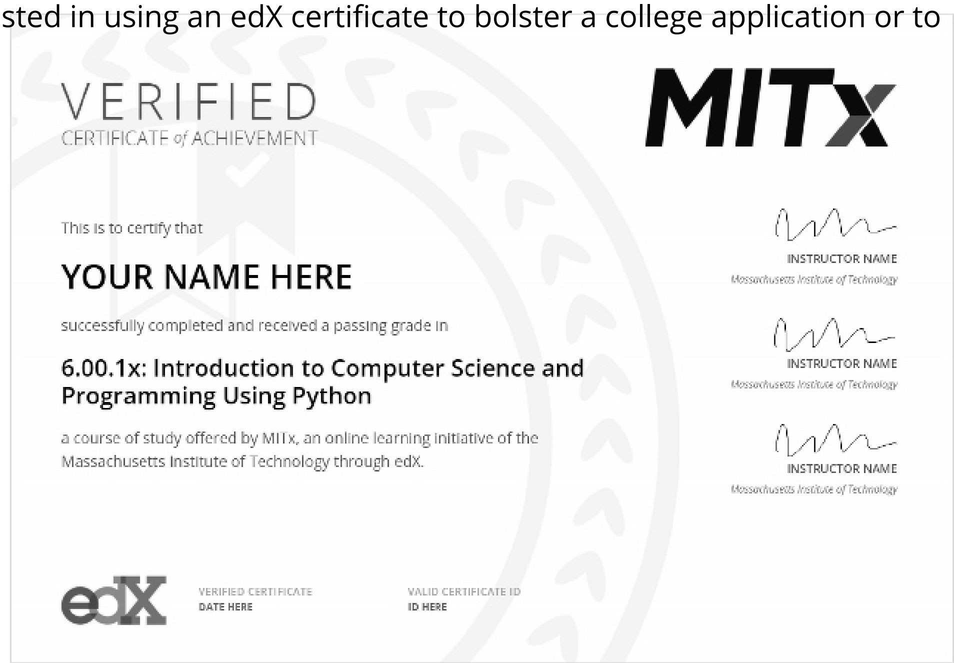
Unanswered

Note: In "See Full Output", the parameter to the hangman function is a string.

Submit You have used 0 of 30 attempts

VERIFIED CERTIFICATES

Interested in using an edX certificate to bolster a college application or to advance your career? With an edX verified certificate, you can demonstrate to colleges, employers, and colleagues that you successfully completed a challenging edX course, while helping to support the edX non-profit mission. You can register to earn a Verified Certificate directly here.



VERIFIED
CERTIFICATE of ACHIEVEMENT

MITx

This is to certify that

YOUR NAME HERE

successfully completed and received a passing grade in

6.00.1x: Introduction to Computer Science and Programming Using Python

a course of study offered by MITx, an online learning initiative of the Massachusetts Institute of Technology through edX.

INSTRUCTOR NAME
Massachusetts Institute of Technology

INSTRUCTOR NAME
Massachusetts Institute of Technology

INSTRUCTOR NAME
Massachusetts Institute of Technology

edX

VERIFIED CERTIFICATE
DATE HERE

VALID CERTIFICATE ID
ID HERE

Problem 4: Hangman Game

Topic: Problem Set 3 / Problem 4

Show Discussion