

PYTHON TEST 2: Hangman

Requirements

Write a small app/script that simulates the hangman game. Follow the next steps:

1. Create a word list (spanish, english... the language is not important).
2. Prompt the user for letter guesses until the word is correctly guessed or the player has exceeded the maximum number of guesses.
3. Show an error if user introduces numbers or other chars that are not letters.
4. Show an error if user introduces words instead letters.
5. Your program should print a list of letters that have been guessed as well as display the correctly guessed letters in the word.

Sample run:

```
Welcome to hangman. You get seven chances to guess the myster

- - - - -

Pick a letter --> e

Guessed letters: E

  O

- - - - -

Pick a letter --> a

Guessed letters: E A

  O
  |

- - - - -
```

```

Pick a letter --> e
Sorry, you already guessed 'E'
Pick a letter --> i

Guessed letters: E A I

  O
 \|

- - - - -

Pick a letter --> o

Guessed letters: E A I O

  O
 \|

_ O _ _ _

Pick a letter --> u

Guessed letters: E A I O U

  O
 \|/

_ O _ _ _

Pick a letter --> y

Guessed letters: E A I O U Y

  O
 \|/

_ O _ _ _ Y

Pick a letter --> 4
'4' is not a valid letter
Pick a letter --> xyz
'XYZ' has more than one letter.

```

Pick a letter --> l

Guessed letters: E A I O U Y L

o
\\|/

_ o _ _ l y

Pick a letter --> s

Guessed letters: E A I O U Y L S

o
\\|/
|

_ o _ _ l y

Pick a letter --> r

Guessed letters: E A I O U Y L S R

o
\\|/
|
/

_ o _ _ l y

Pick a letter --> k

Guessed letters: E A I O U Y L S R K

o
\\|/
|
/ \

_ o _ _ l y

So sorry. You struck out.

The mystery word was 'COMPLY.'

Notes

- You can use the programming language or library that you want.
- Please upload the code in a public repo such as GitHub.

Bonus Points

- Use Python.
- Keep a tracker of used letters and show a message if user try to use the same.
- A “graphical” representation of the hanging man.