Hangman

Using Python

IDLE Shell 3.13.0

Edit Shell Debug Options Window Help

```
Python 3.13.0 (tags/v3.13.0:60403a5, Oct  7 2024, 09:38:07) [MSC v.1941 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

========== RESTART: C:\Users\lulur\Desktop\Python\hangman.py ==========
You have 7 lives left and you have used these letters:
Current word:  - - - -
Guess a letter: e

You have 7 lives left and you have used these letters:  E
Current word:  - - - E
Guess a letter: h

Your letter, H is not in the word.
You have 6 lives left and you have used these letters:  H E
Current word:  - - - E
Guess a letter: d

Your letter, D is not in the word.
You have 5 lives left and you have used these letters:  H E D
Current word:  - - - E
Guess a letter: t

Your letter, T is not in the word.
You have 4 lives left and you have used these letters:  T H E D
Current word:  - - - E
Guess a letter: h

You have already used that letter. Guess another letter.
You have 4 lives left and you have used these letters:  T H E D
Current word:  - - - E
Guess a letter: a

You have 4 lives left and you have used these letters:  H T E A D
Current word:  - A - E
Guess a letter: v

Your letter, V is not in the word.
You have 3 lives left and you have used these letters:  H T E A V D
Current word:  - A - E
Guess a letter:
```

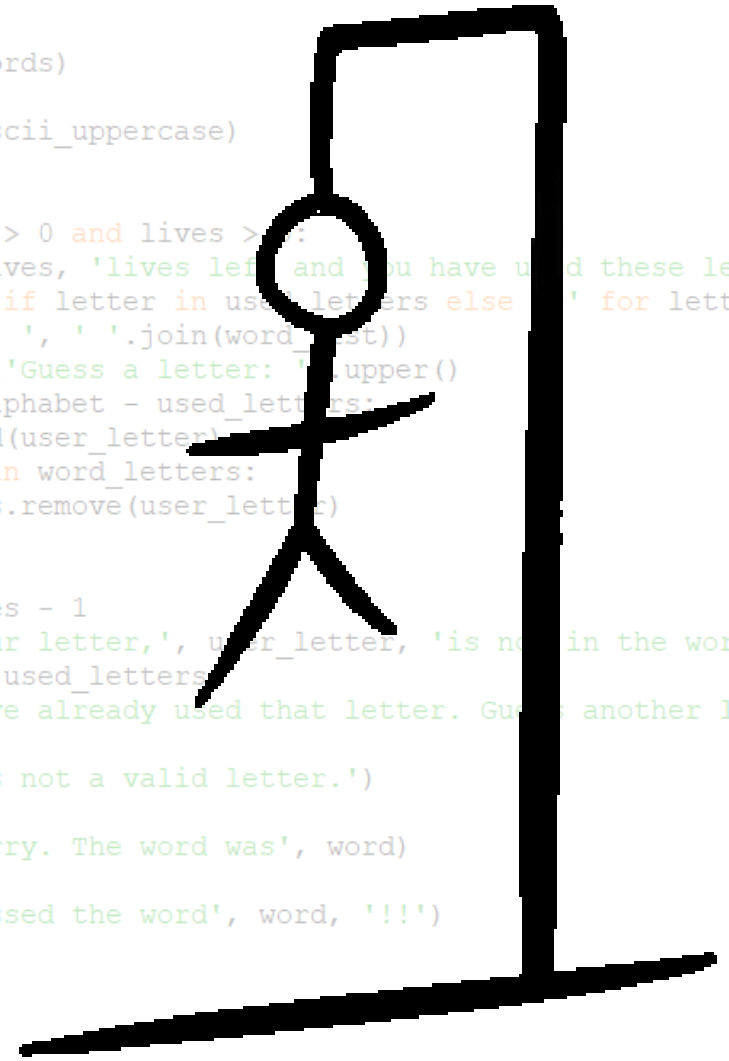
Ln: 40 Col: 16

hangman.py - C:\Users\lulur\Desktop\Python\hangman.py (3.13.0)

File Edit Format Run Options Window Help

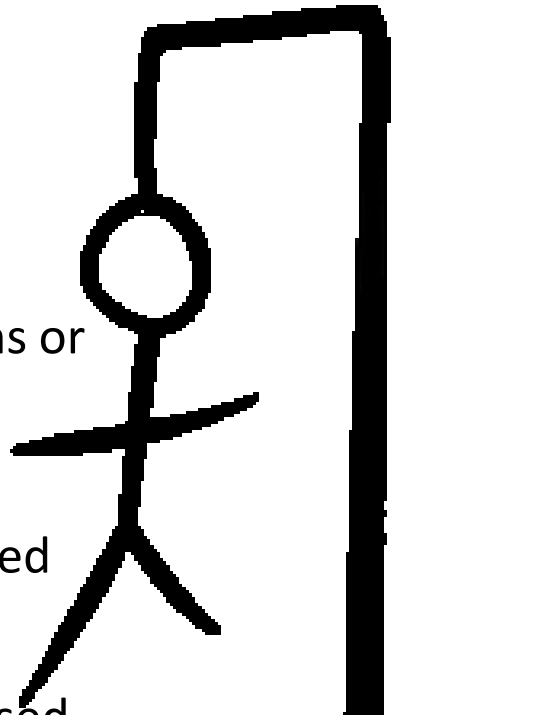
```
import random
from words import words
import string
def get_valid_word(words):
    word = random.choice(words)
    while '-' in word or ' ' in word:
        word = random.choice(words)
    return word.upper()
def hangman():
    word = get_valid_word(words)
    word_letters = set(word)
    alphabet = set(string.ascii_uppercase)
    used_letters = set()
    lives = 7
    while len(word_letters) > 0 and lives >
        print('You have', lives, 'lives lef and you have u d these le
        word_list = [letter if letter in us letters else ' ' for lett
        print('Current word: ', ' '.join(word st))
        user_letter = input('Guess a letter: ').upper()
        if user_letter in alphabet - used_let rs:
            used_letters.add(user_letter)
            if user_letter in word_letters:
                word_letters.remove(user_lett )
                print('')
            else:
                lives = lives - 1
                print('\nYour letter,', u r_letter, 'is n in the wor
        elif user_letter in used_letters:
            print('\nYou have already used that letter. Gu another l
        else:
            print('\nThat is not a valid letter.')
    if lives == 0:
        print('You died, sorry. The word was', word)
    else:
        print('YAY! You guessed the word', word, '!!')

if __name__ == '__main__':
    hangman()
```



Hangman in Python

- This was a quick project in Python to recreate the classic hangman game using Python IDLE.
- It uses a separate words file that contains over 5,000 English words.
- A script was created to:
  - Create a random word generator which excluded words that included hyphens or spaces
  - Dashes such as _ _ _ were added to show the length of the word
  - Letters correctly guessed removed the dashes and were added to the word
  - Letters incorrectly guessed were added to a used letter list which was displayed for the player.
  - Lives were also added to limit the number of times a person could guess.
  - Repeated letter guesses were added to explain this letter had already been used, please give another letter, but it did not impact with loss of life.
  - Print commands to give feedback to the player.



hangman.py - C:\Users\lulur\Desktop\Python\hangman.py (3.13.0)

File Edit Format Run Options Window Help

```
import random
from words import words
import string
def get_valid_word(words):
    word = random.choice(words)
    while '-' in word or ' ' in word:
        word = random.choice(words)
    return word.upper()
def hangman():
    word = get_valid_word(words)
    word_letters = set(word)
    alphabet = set(string.ascii_uppercase)
    used_letters = set()
    lives = 7
    while len(word_letters) > 0 and lives > 0:
        print('You have', lives, 'lives left and you have used these letters: ', ' '.join(used_letters))
        word_list = [letter if letter in used_letters else '-' for letter in word]
        print('Current word: ', ' '.join(word_list))
        user_letter = input('Guess a letter: ').upper()
        if user_letter in alphabet - used_letters:
            used_letters.add(user_letter)
            if user_letter in word_letters:
                word_letters.remove(user_letter)
                print('')
            else:
                lives = lives - 1
                print('\nYour letter,', user_letter, 'is not in the word.')
        elif user_letter in used_letters:
            print('\nYou have already used that letter. Guess another letter.')
        else:
            print('\nThat is not a valid letter.')
    if lives == 0:
        print('You died, sorry. The word was', word)
    else:
        print('YAY! You guessed the word', word, '!!')

if __name__ == '__main__':
    hangman()
```

```
"aback","abart","abandoned","abashed","aberrant","abhorrent","abiding"
"aboriginal","abortive","abounding","abrasive","abrupt","absent","abs
"abusive","acceptable","accessible","accidental","accurate","acid","a
"addicted","adhesive","adjoining","adorable","adventurous","afraid","a
"alike","alive","alleged","alluring","aloof","amazing","ambiguous","a
"annoying","anxious","apathetic","aquatic","aromatic","arrogant","ash
"available","average","awake","aware","awesome","awful","axiomatic","
"bent","berserk","best","better","bewildered","big","billowy","bite-s
"boiling","boorish","bored","boring","bouncy","boundless","brainy","b
"bumpy","burly","bustling","busy","cagey","calculating","callous","ca
"changeable","charming","cheap","cheerful","chemical","chief","childl
"cloudy","closed","clumsy","cluttered","coherent","cold","colorful","
"conscious","cooing","cool","cooperative","coordinated","courageous","
"cumbersome","curious","curly","curved","curvy","cut","cute","cute","
"dazzling","dead","deadpan","deafening","dear","debonair","decisive","
"demonic","delirious","dependent","depressed","deranged","descriptive
"direful","dirty","disagreeable","disastrous","discreet","disgusted","
"doubtful","drab","draconian","dramatic","dreary","drunk","dry","dull
"economic","educated","efficacious","efficient","eight","elastic","ela
"enchanting","encouraging","endurable","energetic","enormous","enterta
"evasive","even","excellent","excited","exciting","exclusive","exotic
"faithful","fallacious","false","familiar","famous","fanatical","fancy
"feigned","female","fertile","festive","few","fierce","filthy","fine",
"flowery","fluffy","fluttering","foamy","foolish","foregoing","forget
"friendly","frightened","frightening","full","fumbling","functional","
"garrulous","gaudy","general","gentle","giant","giddy","gifted","gigan
"gorgeous","graceful","grandiose","grateful","gratis","gray","greasy",
"grumpy","guarded","guiltless","gullible","gusty","guttural","habitua
"hard-to-find","harmonious","harsh","hateful","heady","healthy","heart
"highfalutin","high-pitched","hilarious","hissing","historical","holis
"humdrum","humorous","hungry","hurried","hurt","hushed","husky","hypno
"illustrious","imaginary","immense","imminent","impartial","imperfect
"industrious","incredible","inexpensive","infamous","innate","innocent
"irate","irritating","itchy","jaded","jagged","jazzy","jealous","jitte
"kind","kindhearted","kindly","knotty","knowing","knowledgeable","know
"laughable","lavish","lazy","lean","learned","left","legal","lethal","
"lonely","long","longing","long-term","loose","lopsided","loud","lout
"lyrical","macabre","macho","maddening","madly","magenta","magical","
"massive","married","marvelous","material","materialistic","mature","
"mighty","military","milky","mindless","miniature","minor","miscreant
```

Left hand side: Python script. Right hand side: A selection of the words used for the random word generator.

IDLE Shell 3.13.0

File Edit Shell Debug Options Window Help

```
Python 3.13.0 (tags/v3.13.0:60403a5, Oct  7 2024, 09:38:07) [MSC v.1941 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\lulur\Desktop\Python\hangman.py ==============
You have 7 lives left and you have used these letters:
Current word:  - - - -
Guess a letter: e

You have 7 lives left and you have used these letters:  E
Current word:  - - - E
Guess a letter: h

Your letter, H is not in the word.
You have 6 lives left and you have used these letters:  H E
Current word:  - - - E
Guess a letter: d

Your letter, D is not in the word.
You have 5 lives left and you have used these letters:  H E D
Current word:  - - - E
Guess a letter: t

Your letter, T is not in the word.
You have 4 lives left and you have used these letters:  T H E D
Current word:  - - - E
Guess a letter: h

You have already used that letter. Guess another letter.
You have 4 lives left and you have used these letters:  T H E D
Current word:  - - - E
Guess a letter: a

You have 4 lives left and you have used these letters:  H T E A D
Current word:  - A - E
Guess a letter: v

Your letter, V is not in the word.
You have 3 lives left and you have used these letters:  H T E A V D
Current word:  - A - E
Guess a letter:
```

Ln: 40 Col: 16