1. ***A program for Hangman in python***

The code satisfies the requirements

* + 1. One word generated randomly
    2. Player will be presented with a number of blank spaces representing the missing letters the player needs to find.
    3. If the player’s chosen letter exists in the answer, then all places in the answer where that letter appear will be revealed.
    4. Every time the player guesses a letter wrong, the player’s life will be deducted.
    5. The player must find the missing word before the player’s life becomes zero.
    6. Player lives is represented using number instead of hangman picture

1. import random
2. from wordslist import words
3. def get\_word():
4. word =random.choice(words)
5. return word.upper()
6. def hangman():
7. word = get\_word()
8. alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
9. guessed\_letter=[]
10. lives =6
11. guessed = False
12. print(len(word),'letters in the word')
13. print(len(word)\*'\_')
14. while guessed == False and lives > 0:
15. print(lives,'lives left')
16. guess = input('guess a letter:').upper()
17. if len (guess) == 1:
18. if guess not in alphabet:
19. print('Invalid character')
20. lives -=1
21. elif guess in guessed\_letter:
22. print('you have already guessed the letter')
23. elif guess not in word:
24. print('Guessed letter not present in the word')
25. guessed\_letter.append(guess)
26. lives -=1
27. elif guess in word:
28. print('your guess is present in the word')
29. guessed\_letter.append(guess)
31. status = ''
32. if guessed == False:
33. for letter in word:
34. if letter in guessed\_letter:
35. status += letter
36. if status == word:
37. lives = 0
38. print('you win the game')
39. elif letter not in guessed\_letter:
40. status += '\_'
41. print(status)
42. print('You run out of guesses')
43. hangman()

**Output**

![A screenshot of a cell phone

Description automatically generated]()

1. ***Refactoring***

Code smells

**Duplicated code:** The function gets\_word is used only once, so it can be directly used in the main function

1. import random
2. from wordslist import words
3. def hangman():
4. word =random.choice(words).upper()
5. alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
6. guessed\_letter=[]
7. lives =6
8. guessed = False
9. print(len(word),'letters in the word')
10. print(len(word)\*'\_')
11. while guessed == False and lives > 0:
12. print(lives,'lives left')
13. guess = input('guess a letter:').upper()
14. if len (guess) == 1:
15. if guess not in alphabet:
16. print('Invalid character')
17. lives -=1
18. elif guess in guessed\_letter:
19. print('you have already guessed the letter')
20. elif guess not in word:
21. print('Guessed letter not present in the word')
22. guessed\_letter.append(guess)
23. lives -=1
24. elif guess in word:
25. print('your guess is present in the word')
26. guessed\_letter.append(guess)
28. status = ''
29. if guessed == False:
30. for letter in word:
31. if letter in guessed\_letter:
32. status += letter
33. if status == word:
34. lives = 0
35. print('you win the game')
36. elif letter not in guessed\_letter:
37. status += '\_'
38. print(status)
39. print('You run out of guesses')
40. hangman()

***3. Create a Git directory for your assignment***

Link: <https://github.com/s331717/Jilu_hangman>

Step 1: To create a new respiratory fill the columns owner and respiratory name. If you are selecting public domain you must ensure to select any license. Then click create respiratory. Then you will get the link for the project.

![A screenshot of a social media post

Description automatically generated]()

Step 2: Choose your files related to the project

A screenshot of a cell phone

Description automatically generated

Step 3: uploaded files can be seen in the link .

![A screenshot of a cell phone

Description automatically generated]()

1. ***How TDD has been implemented to create your program***

Test driven development is an approach program development in which the testing and coding are interleaved. Once the coding for a function is finished, then it will be tested and meet the requirements. After that coding for next requirement will do and after that its testing should finish before identifying next functionality

**First requirement: check the random selection of word from wordlist**

![A screenshot of a cell phone

Description automatically generated]()

**Second Requirement: To print the length of the word and ensure the blank spaces in the place of letters**

![A screenshot of a cell phone

Description automatically generated]()

**Third requirement: Main loop for getting input and check the condition such as whether the input letter is present in the word or the input is an invalid character, or it is an already enter letter**

![A screenshot of text

Description automatically generated]()

![A screenshot of a cell phone

Description automatically generated]()

**Fourth requirement: Combine the entire requirements and do the user testing**

import random

from wordslist import words

def get\_word():

    word =random.choice(words)

    return word.upper()

def hangman():

    word = get\_word()

    alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

    guessed\_letter=[]

    lives =6

    guessed = False

    print(len(word),'letters in the word')

    print(len(word)\*'\_')

    while guessed == False and lives > 0:

        print(lives,'lives left')

        guess = input('guess a letter:').upper()

        if len (guess) == 1:

            if guess not in alphabet:

                print('Invalid character')

                lives -=1

            elif guess in guessed\_letter:

                print('you have already guessed the letter')

            elif guess not in word:

                print('Guessed letter not present in the word')

                guessed\_letter.append(guess)

                lives -=1

            elif guess in word:

                print('your guess is present in the word')

                guessed\_letter.append(guess)

        status = ''

        if guessed == False:

            for letter in word:

                if letter in guessed\_letter:

                    status += letter

                    if status == word:

                        lives = 0

                        print('you win the game')

                elif letter not in guessed\_letter:

                    status += '\_'

            print(status)

    print('You run out of guesses')

hangman()

![A screenshot of a cell phone

Description automatically generated]()