Hangman is a game of words. The game gives generates a random word and we have to guess what the word is. To do this we need a large list of words and luckily we found one of the internet that contains 2466 different words in a list.

The next step is getting the computer to randomly pick a word from the list however on inspection of the list, there are some words that are compound words and that would be hard to get so we define a function that helps us get valid words (words without “-“ in them).

import random

from words import words

def get\_valid\_word(words):

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

    while "-" in word or " " in word:

        word = random.choice(words)

    return word.upper()

<code>from words import words</code> line of code imports a words variable from a words.py file. This is what the words.py file looks like. It contains a list variable that contains the 2466 words we are going to use for the game. <br />

<img src=”/images/hangman\_words.JPG”><br />

In our <code>get\_valid\_word</code> function, what we are simply doing is getting a random word using <code>random.choice()</code> and then we check if “-“ is in it. If it is, we pick another random word then at the end the function returns our word in capital letters i.e. <code>word.upper()</code>. <br />

The next step is to define our hangman function. What we want to do is we want to call the <code>get\_valid\_word</code> function and assign it to a variable called word so that we can make use of the word in the hangman function. After getting the word we want the user to be able to input a letter and then we check if the letter is in the word. To be able to do this, we are going to make use of <code>set()</code> data type. After we get our valid word in our hangman function, we would use <code>set(word)</code> to split the word into individual items in the set and we assign it to a variable(“word\_letters”). Then we also create an empty set of “used\_letters” so we can keep track of the letters the player has used. We are also going to ensure that the player only inputs valid characters (A-Z) so we would import string into our code just at the top. <code>string.ascii\_uppercase</code> returns A-Z so we would assign that to our alphabet variable.

def hangman():

    word = get\_valid\_word(words)

    word\_letters = set(word) #letters in the word

    alphabet = set(string.ascii\_uppercase)

    used\_letters = set() #what the user guessed.

    #getting user input

    while len(word\_letters) > 0:

        user\_letter = input("Guess a letter: ").upper()

        if user\_letter in alphabet - used\_letters: #alphabet - used\_letters would give you a set of the letters in the alphabet not yet used.

            used\_letters.add(user\_letter)

            if user\_letter in word\_letters:

                word\_letters.remove(user\_letter)

            else:

                print(f"Your letter {user\_letter} is not in the word.")

        elif user\_letter in used\_letters:

            print("You have already used this letter before. Please try again.")

        else:

            print("Invalid character. Please try again.")

    else:

        print(f"You guessed the word {word} !!!")

In order for the player to keep guessing until they get the word, we would make use of a while loop. While the length of our <code>word\_letters</code> is greater than zero(0), the player can input a letter and then we check if the letter is in the alphabet but not in <code>used\_letters</code> (so that we only input valid characters we have not already used). If that statement is true, we add the <code>user\_letter</code> to our <code>used\_letters</code> set and then we check if the <code>user\_letter</code> is in the <code>word\_letters</code> set. If that statement equates to true meaning the letter the user typed is in the word, we remove that word from the <code>word\_letters</code> set so the length of our <code>word\_letters</code> set reduces which would affect our while loop. If the <code>user\_letter</code> is not in the <code>word\_letters</code> set, the else statement runs which would print out “Your letter is not in the word”. <br />

Of course, if our <code>user\_input</code> is in the <code>used\_letters</code> we send the player a message that they have already used the letter before and then the last else statement there is for if the <code>user\_letter</code> is not in the <code>alphabet-used\_letters</code> set meaning it is not a valid character so we send them a message that says it is not a valid character. <br />

That is the basis of the hangman game and then we call the function at the end to get it to run. This is what the entire code looks like.

import random

from words import words

import string

def get\_valid\_word(words):

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

    while "-" in word or " " in word:

        word = random.choice(words)

    return word.upper()

def hangman():

    word = get\_valid\_word(words)

    word\_letters = set(word) #letters in the word

    alphabet = set(string.ascii\_uppercase)

    used\_letters = set() #what the user guessed.

    #getting user input

    while len(word\_letters) > 0:

        user\_letter = input("Guess a letter: ").upper()

        if user\_letter in alphabet - used\_letters: #alphabet - used\_letters would give you a set of the letters in the alphabet not yet used.

            used\_letters.add(user\_letter)

            if user\_letter in word\_letters:

                word\_letters.remove(user\_letter)

            else:

                print(f"Your letter {user\_letter} is not in the word.")

        elif user\_letter in used\_letters:

            print("You have already used this letter before. Please try again.")

        else:

            print("Invalid character. Please try again.")

    print(f"You guessed the word {word} !!!")

hangman()

An example of the implementation can be seen below with the result in the bottom right console. <br />

<img src=”/images/hangman\_basic.JPG”><br />

<br />

Cheers 🥂