In [4]:

```
word_list = ["aardvark", "baboon", "camel"]

#TODO-1 - Randomly choose a word from the word_list and assign it to a variable called chosen_word.
import random
chosen_word = random.choice(word_list)

#TODO-2 - Ask the user to guess a letter and assign their answer to a variable called guess. Make guess lowercase.
guess = input("Guess a letter: ").lower()

#TODO-3 - Check if the letter the user guessed (guess) is one of the letters in the chosen_word.
for letter in chosen_word:
    if letter == guess:
        print("Right")
    else:
        print("Wrong")
```

```
Guess a letter: o
Wrong
Wrong
Wrong
Right
Right
Wrong
```

In [6]:

```
import random
word_list = ["aardvark", "baboon", "camel"]
chosen_word = random.choice(word_list)

#Testing code
print(f'Pssst, the solution is {chosen_word}.')

#TODO-1: - Create an empty List called display.
#For each letter in the chosen_word, add a "_" to 'display'.
#So if the chosen_word was "apple", display should be ["_", "_", "_", "_", "_"] with 5 "_" representing each letter to guess.
display = []
word_length = len(chosen_word)
for _ in range(word_length):
    display += "_"

guess = input("Guess a letter: ").lower()

#TODO-2: - Loop through each position in the chosen_word;
#If the letter at that position matches 'guess' then reveal that letter in the display at that position.
#e.g. If the user guessed "p" and the chosen word was "apple", then display should be ["_", "p", "p", "_", "_"].
for position in range(word_length):
    letter = chosen_word[position]
    #print(f"Current position: {position}\n Current letter: {letter}\n Guessed letter: {guess}")
    if letter == guess:
        display[position] = letter

#TODO-3: - Print 'display' and you should see the guessed letter in the correct position and every other letter replace with "_".
#Hint - Don't worry about getting the user to guess the next letter. We'll tackle that in step 3.
print(display)
```

```
Pssst, the solution is camel.
Guess a letter: c
['c', '_', '_', '_', '_']
```

In [ ]:

In [ ]:

```
#Step 3

import random
word_list = ["aardvark", "baboon", "camel"]
chosen_word = random.choice(word_list)
word_length = len(chosen_word)

#Testing code
print(f'Pssst, the solution is {chosen_word}.')

#Create blanks
display = []
for _ in range(word_length):
    display += "_"

#TODO-1: - Use a while loop to let the user guess again. The loop should only stop once the user has guessed all the letters in the chosen_word and 'display' has no more blanks (
end_of_game = False

while not end_of_game:
    guess = input("Guess a letter: ").lower()

    #Check guessed letter
    for position in range(word_length):
        letter = chosen_word[position]
        #print(f"Current position: {position}\n Current letter: {letter}\n Guessed letter: {guess}")
        if letter == guess:
            display[position] = letter

    print(display)

    #Check if there are no more "_" left in 'display'. Then all letters have been guessed.
    if "_" not in display:
        end_of_game = True
        print("You win.")
```

In [ ]:

```
#Step 4

import random

stages = ['''
  +---+
  |   |
  O   |
 /|\  |
 / \  |
      |
=========
''', '''
  +---+
  |   |
  O   |
 /|\  |
 /    |
      |
=========
''', '''
  +---+
  |   |
  O   |
 /|\  |
      |
      |
=========
''', '''
  +---+
  |   |
  O   |
 /|   |
      |
      |
=========
''', '''
  +---+
  |   |
  O   |
  |   |
      |
      |
=========
''', '''
  +---+
  |   |
  O   |
      |
      |
      |
=========
''', '''
  +---+
  |   |
      |
      |
      |
      |
=========
''']

end_of_game = False
word_list = ["ardvark", "baboon", "camel"]
chosen_word = random.choice(word_list)
word_length = len(chosen_word)

#TODO-1: - Create a variable called 'lives' to keep track of the number of lives left.
#Set 'lives' to equal 6.
lives = 6

#Testing code
print(f'Pssst, the solution is {chosen_word}.')

#Create blanks
display = []
for _ in range(word_length):
    display += "_"

while not end_of_game:
    guess = input("Guess a letter: ").lower()

    #Check guessed letter
    for position in range(word_length):
        letter = chosen_word[position]
        # print(f"Current position: {position}\n Current letter: {letter}\n Guessed letter: {guess}")
        if letter == guess:
            display[position] = letter

    #TODO-2: - If guess is not a letter in the chosen_word,
    #Then reduce 'lives' by 1.
    #If lives goes down to 0 then the game should stop and it should print "You lose."
    if guess not in chosen_word:
        lives -= 1
        if lives == 0:
            end_of_game = True
            print("You lose.")

    #Join all the elements in the list and turn it into a String.
    print(f"{' '.join(display)}")

    #Check if user has got all letters.
    if "_" not in display:
        end_of_game = True
        print("You win.")

    #TODO-3: - print the ASCII art from 'stages' that corresponds to the current number of 'lives' the user has remaining.
    print(stages[lives])
```

In [ ]:

```
#Step 5

import random

#TODO-1: - Update the word list to use the 'word_list' from hangman_words.py
#delete this line: word_list = ["ardvark", "baboon", "camel"]
from hangman_words import word_list

chosen_word = random.choice(word_list)
word_length = len(chosen_word)

end_of_game = False
lives = 6

#TODO-3: - Import the logo from hangman_art.py and print it at the start of the game.
from hangman_art import logo
print(logo)

#Testing code
print(f'Pssst, the solution is {chosen_word}.')

#Create blanks
display = []
for _ in range(word_length):
    display += "_"

while not end_of_game:
    guess = input("Guess a letter: ").lower()

    #TODO-4: - If the user has entered a letter they've already guessed, print the letter and let them know.
    if guess in display:
        print(f"You've already guessed {guess}")

    #Check guessed letter
    for position in range(word_length):
        letter = chosen_word[position]
        #print(f"Current position: {position}\n Curreot letter: {letter}\n Guessed letter: {guess}")
        if letter == guess:
            display[position] = letter

    #Check if user is wrong.
    if guess not in chosen_word:
        #TODO-5: - If the letter is not in the chosen_word, print out the letter and let them know it's not in the word.
        print(f"You guessed {guess}, that's not in the word. You lose a life.")

        lives -= 1
        if lives == 0:
            end_of_game = True
            print("You lose.")

    #Join all the elements in the list and turn it into a String.
    print(f"{' '.join(display)}")

    #Check if user has got all letters.
    if "_" not in display:
        end_of_game = True
        print("You win.")

    #TODO-2: - Import the stages from hangman_art.py and make this error go away.
    from hangman_art import stages
    print(stages[lives])
```

In [ ]:

```
import random
from hangman_art import stages, logo
from hangman_words import word_list
from replit import clear

print(logo)
game_is_finished = False
lives = len(stages) - 1

chosen_word = random.choice(word_list)
word_length = len(chosen_word)

display = []
for _ in range(word_length):
    display += "_"

while not game_is_finished:
    guess = input("Guess a letter: ").lower()

    #Use the clear() function imported from replit to clear the output between guesses.
    clear()

    if guess in display:
        print(f"You've already guessed {guess}")

    for position in range(word_length):
        letter = chosen_word[position]
        if letter == guess:
            display[position] = letter
    print(f"{' '.join(display)}")

    if guess not in chosen_word:
        print(f"You guessed {guess}, that's not in the word. You lose a life.")
        lives -= 1
        if lives == 0:
            game_is_finished = True
            print("You lose.")

    if not "_" in display:
        game_is_finished = True
        print("You win.")

    print(stages[lives])
```

In [ ]: