Create a Hangman Game

Click here (https://en.wikipedia.org/wiki/Hangman_(game)) to learn how to play, or to simply get a refresher for how the game works.

Create an object-orient program that works exactly as the Wikipedia page suggests.

Given, are all of the classes to build out the game. While extremely helpful, please note that this may be an ineffective method of building your program, because it hinders how one you would naturally build a program in your own way if they were left to figure it out for themselves.

How to build the program

- When you start the game, you will need to select a random word from a list of at least 10 words (You have full control over which words you want to use for you program). This will be your secret word. Your secret word will be represented in the program as a group of underscores. For as long as the word is, you should also have that many underscores.
- Once the word is selected, your game will commence. Perform a Google search to figure out how to select a random word from a list using Python.
  Hint There's package you can import into your application that does this for you.
- The end user will have a total of 8 chances to guess the correct letter from the secret word. If the end user makes 8 incorrect guesses, the game will end.
- As you guess the correct letters, the letters you have guess will then take place of the underscores that letter represents.
  For Example: If your secret word is 'watermelon' and so far you have guessed the letters 'a' and 'e', the word you're trying to guess will appear as follows: *a* e e _.
  Keep in mind that if you guess a letter that appears more than once in your secret word, make sure that the letter is populated anywhere that letter would be.

In [ ]:
```
# Object-Oriented Programming Method

import random
from IPython.display import clear_output

#Creating the Object for Hangman

class Hangman():
    def __init__(self):
        self.chances = 8
        self.word = list()
        self.hidden_word = str()
        self.correct_guesses = int()
        self.word_list = ('mango','pear','peach','plum')

    def start(self):
        print("Welcome to Python Hangman!")
        print('-'*60)
        self.hidden_word = random.choice(self.word_list)
        self.word = ['_'] * len(self.hidden_word)
        self.correct_guesses = ''.join(self.word).count('_')
        self.instructions() # This is optional based on preference

    def instructions(self):
        print(''.join([i + ' ' for i in self.word]))
        print('You have {} remaining. Guess Carefully!'.format(self.chance
s))

    def ask(self):
        letter = input("Guess Letter. ")
        if letter.lower() == 'done':
            return False
        if letter.lower() not in self.hidden_word:
            print("{} is incorrect. Please Try again!".format(letter))
            self.chances -=1
        else:
            print("{} has been found".format(letter))
            for index, l in enumerate(self.hidden_word): #find the index an
d value of the chosen word
                if l is letter: #if the letter you are searching for is fou
nd at the index of the chosen word
                    self.word[index] = letter # make the index of the under
scored word equal to the letter we chose.
            self.correct_guesses = ''.join(self.word).count('_')
        return True



# Here lies our Main function - This will run our program and create/instan
tiate our class

def main():
    game = Hangman()

    game.start()

    while True:
        #Base Case
        while True:
            if game.ask() == False:
                break
            if game.chances == 0: #if you lose
```

```
Welcome to Python Hangman!
-------------------------------------------------------------

_ _ _ _ _
You have 8 remaining. Guess Carefully!
Guess Letter. p
p has been found
Guess Letter. e
e has been found
Guess Letter. a
a has been found
Guess Letter. c
c has been found
Guess Letter. h
h has been found
You won!
```

In [ ]: