Purpose of each file in the repository:

- .gitignore -> Was already in the repository.
- LICENSE -> A license made by git that was already in the repository.
- README.pdf -> The current README file.
- List_of_foods.txt -> A list of foods followed by how much it is able to increase the hunger stat on the pet.
- main.py ->
- Random_behavior.txt -> A list of actions the pet could take followed by numbers separated by commas representing how the action affects the different attributes of the pet.
- Try_text.py -> Allows us to test how GitHub works in terms of changing text and how we can add, commit, push, and pull from and into GitHub.

How to run the program from the command line:

To run this program, it depends on the device the individual currency has. If they are on Mac, they must use a command such as:

```
python3 main.py [name]
```

where the name can be any string the user may want to import. For example, if I would like a pet named Harry, the command will now be `python3 main.py Harry`. For those who are on Windows, the user would have to do the same but without the 3 next to the word "python" such as:

```
python main.py [name]
```

How to use your program and/or interpret the output of the program:

To start our program, please follow the instructions on how to run the program above. From there, a menu would appear with a list of potential actions you can take to help execute the code and take care of the pet. In doing this, a behavior that the Pet wants to do is randomized. The menu will be promoted for the user to specify the number that correlates with that behavior and input that number into the terminal. When starting the program, the format will be as follows: "Welcome to the virtual pet menu", a list of 1-9 of the possible options, a random action and finally an input statement. As the user keeps playing, the output will change into the print statement provided by the action, the list of possible options, a random action and again, the input statement. Depending on the user's choice, it would either prompt another input based on the instructions or indicate that the current action is being taken place. To look at the pet's stats, the user must input the option "8". Then, a bar plot is outputted to graph the emotions attributes of the pet. It updates the stats of the attributes based on the behavior that is carried out. When the user determines that they do not want to take care of the pet, the user must input "cancel" when the menu prompts another response. By doing so, it allows the code to stop with the user's command.

Attribution:

Method/function	**Primary author**	**Techniques demonstrated**
pet_emotion	Yasmeen Sbih	Visualizing data with pyplot or seaborn
__str__	Anna Lian	Magic methods other than __init__()
read_food_list	Andrea Almeda	Regular expressions
wellbeing_pet	Logan Le	List Comprehensions
hug_pet	Rachel Wang	Conditional expressions
clean()	Yasmeen Sbih	Use of a key function
parse_arguments	Anna Lian	The ArgumentParser class from the argparse module
eat()	Andrea Almeda	Sequence unpacking
pet_pet()	Logan Le	F-string containing expressions
random_behavior	Rachel Wang	With statements

These methods and functions all have techniques demonstrated within the code but there are still many methods and functions that are not included in the chart that we have done such as the menu() function and the rest of the action functions.