This is a Pokemon “Do All” program that can be run from the command line. It provides several functionalities such as searching for Pokemon by name, type, or stats, comparing Pokemon, visualizing Pokemon stats, adding and removing Pokemon.

To run this program from the command line, you need to have Python installed on your machine.

Follow the steps below:

1. Save the program code to a file with the name pokemon.py.
2. Open a command prompt or terminal window in the directory where the pokedex.py file is located.
3. Run the program by typing the command python pokedex.py pokedex.json, where pokedex.json is the file name containing the Pokemon data in JSON format. The program will create a Pokedex object using the specified file.

Once the program is running, you will be prompted with a menu of options to choose from. You can type the number of the option you want to select and press Enter to execute it. The available options are:

1. Search for a Pokemon by name.
2. Search for Pokemon by type.
3. Search for a Pokemon by stats.
4. Compare two Pokemon.
5. Visualize a Pokemon's stats.
6. Add a Pokemon to the Pokedex.
7. Remove a Pokemon from the Pokedex.
8. Quit the program.

Depending on the option you select, the program may prompt you to enter additional information such as the name of the Pokemon you want to search for or the stats you want to search by.

The output of the program will depend on the option you select. For example, if you search for a Pokemon by name, the program will display information about the Pokemon, such as its type, HP, attack, defense, special attack, special defense, and speed. If you compare two Pokemon, the program will display a comparison of their stats. If you visualize a Pokemon's stats, the program will display a bar chart of its stats. Add Pokemon does not have an expected output but instead appends a row based on an input string ("", tester,テスター,测试员,testeur,Grass,Poison,21,21,21,21,21,21). Remove Pokemon is the same but removes rows based on a string(tester) of the English name of the Pokemon.

Files:

- README.pdf
  - Our readme
- codetest.py
  - A place to test our code so we don't need to mess up the main code
- pokedex.csv
  - This is used for our visualization
- pokedex.json
  - What the code uses to run. Code uses this file to pull information about different Pokemon
  - Source: https://github.com/fanzeyi/pokemon.json
  - Author: https://github.com/fanzeyi
  - We used this code as a way to pull information on different Pokemon in order to be used in our code
- pokemon.py
  - This is the main code that is being used

Pokemon Class:

Method/Function	**Primary Author**	**Techniques Demonstrated**
init	Samson Mulugeta	

Pokedex Class:

Method/Function	**Primary Author**	**Techniques Demonstrated**
init	Samson Mulugeta	with statements
search_by_name	Samson Mulugeta	conditional expressions
search_by_type	Samson Mulugeta	comprehensions or generator expressions
search_by_stats	Kyle Duong	Optional parameters and/or keyword arguments
compare_pokemon	Samson Mulugeta	F-strings containing expressions
pokemon_visualize	Peter Zheng	visualizing data with pyplot or seaborn specifically plt.subplots
add_pokemon	Peter Zheng	Sequence unpacking
remove_pokemon	Peter Zheng	Read_csv/to_csv (method not

		used to demonstrate technique but as pair with add_pokemon)
get_all_types	Kyle Duong	set operations on sets or frozensets
print_all_types	Kyle Duong	use of a key function (which can be a lambda expression) with one of the following commands: list.sort(), sorted(), min(), or max()
get_pokemon_name	Samson Mulugeta	
main	Samson Mulugeta	
parse_args	Kyle Duong	the ArgumentParser class from the argparse module