

ENGR 102 – Programming Practice

Mini Project 4

Fall 2018

Tags: Tkinter, GUI Widgets, Layout, Databases, Crawling, Searching, Filtering, PIL

Pokemon, which is short for "pocket monster," has become a phenomenal success in world among children. There are cartoons and even a movie based on the Pokemon characters. In that imaginary world, Pokemon are the creatures of all shapes and sizes who live in the wild or alongside humans. There are currently more than 700 creatures that inhabit the Pokemon universe. The Pokedex is an electronic device designed to catalogue and provide information regarding the various species of Pokemon featured in the Pokemon video game, anime and manga series. In this mini project, you are going to develop a Pokedex, which you will collect data by crawling, and transfer simple information about Pokemon to the user.

How should it look like?

Your program will have a graphical user interface (GUI) which will function and look like as explained below.

Details about how it should work are also provided.

1. Initially, your GUI will look like as shown in Figure 1. There are 4 labels, 3 buttons, one progress bar, one combobox, one entry and one listbox.

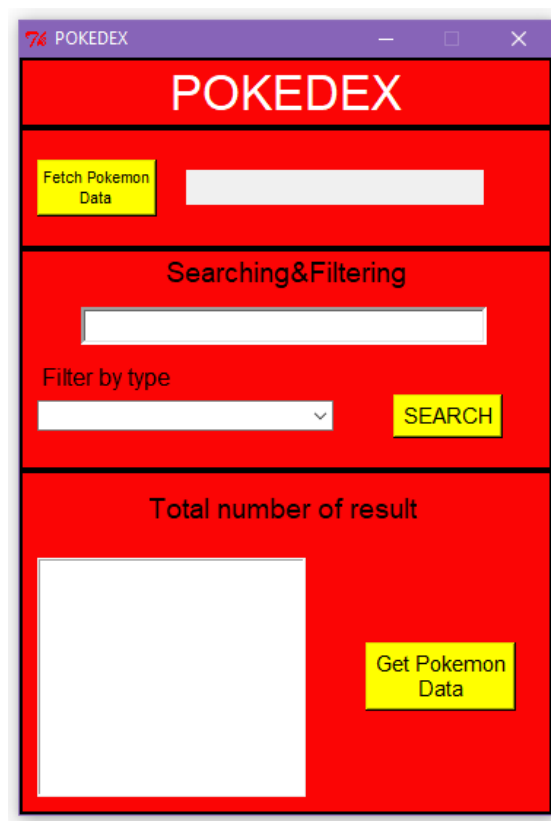


Figure 1. Main GUI window

- After the user presses **Fetch Pokemon Data** button, your program should collect data from internet by crawling. The user then should be able to do some Searching & Filtering on that collected data and get results by pressing **SEARCH** button. After that, the user can press **Get Pokemon Data** button, if he/she wants to see Pokemon Data. In that case, you should show that data to the user with some labels. The GUI window will expand to the right and will look like as shown in Figure 2 when showing Pokemon Data. There are multiple labels and one canvas (for Pokemon image) in the extended part of the GUI.

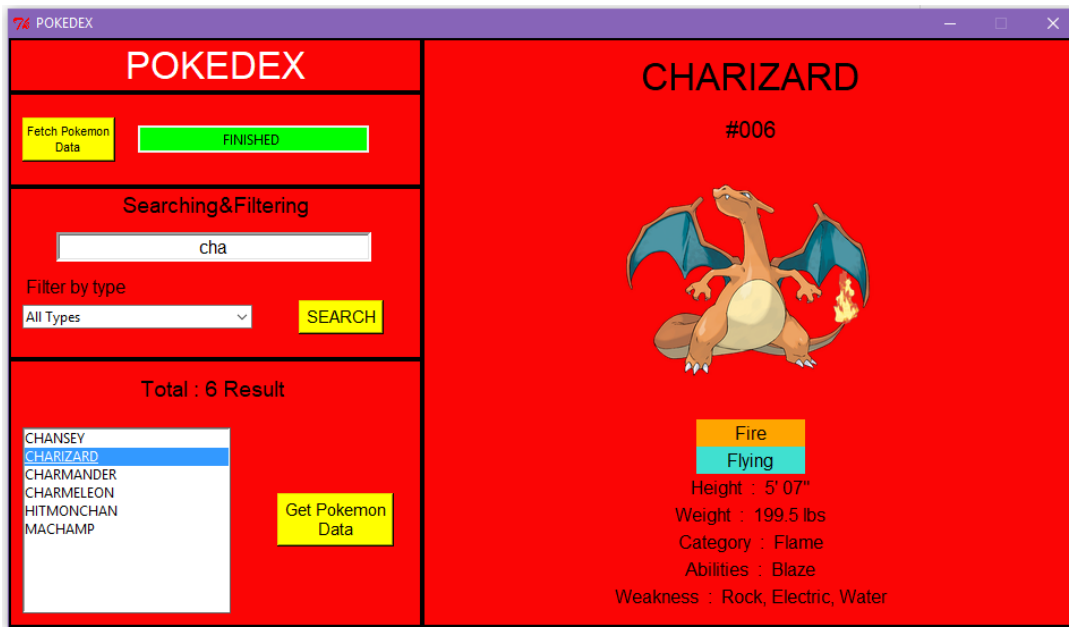


Figure 2. GUI window when displaying Pokemon data.

How should it work?

- First, the user presses **Fetch Pokemon Data** button. When this button is pressed, you should do some crawling from “<https://www.pokemon.com/us/pokedex/>”+”PokemonName” website. The URL will be changed according to the Pokemon name, which means that you should do crawling from multiple websites, but the format of each site is identical. In other words, you can collect data for 151 Pokemon with one crawler function. This function takes the name of the Pokemon as a parameter, and connects to the appropriate link to that name and crawl the necessary data with BeautifulSoup. The Pokemon list is shared with you in a txt file for you to access the website of all the Pokemons. For example, you should go to the following website to collect data about Charizard: “<https://www.pokemon.com/us/pokedex/charizard>”.
- In total, there are 151 Pokemon. You must collect specific data for each Pokemon when you take data from the website. You can store these data in a nested dictionary like below. At the end, there will be 151 keys in that nested dictionary which stores each Pokemon’s data.

{PokemonName: {“attribute_name”: attribute}}

3. Although it is logical to store the data on the dictionary for temporary use, this dictionary will be deleted after the program is closed. It will be compelling and time consuming to enter 151 websites repeatedly on every restart. In order to solve this problem, once you have crawled the data and created a nested dictionary, you must save this dictionary as a database file through the **pickle** module.
4. When the user presses **Fetch Pokemon Data** button, the program first should look at the database file. If the database file exists, a nested dictionary from this database file must be created. If the database file is not available, a nested dictionary should be created by crawling data from 151 websites with BeautifulSoup and saved as database file for future use.
5. It will took time to crawl data from 151 websites. You must create a progress bar for this data crawling process so that the user can see progress stage. You can create the logic of the progress bar as follows. There are 151 sites in total, and if you are crawling data from the 30th site, you have made 30/151, approximately 20% progress. When all crawling process is finished, you must display a text in the progress bar to indicate that the crawling process is over. (see Figure 3 and Figure 4)

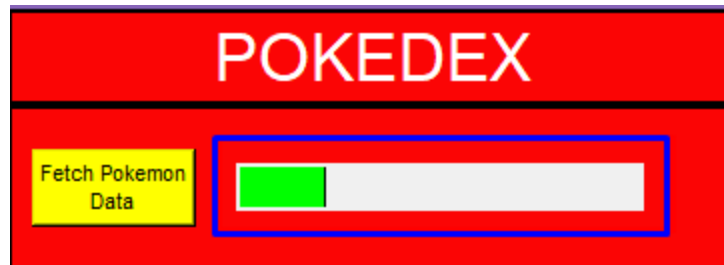
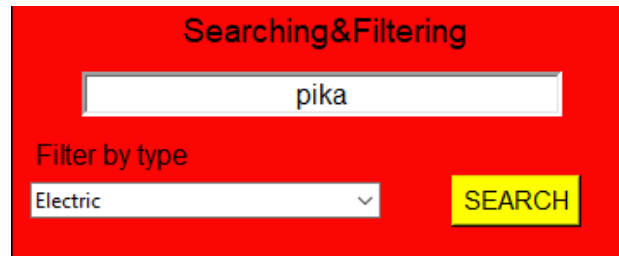


Figure 3. Progress Bar on action.



Figure 4. Progress Bar when crawling process finished.

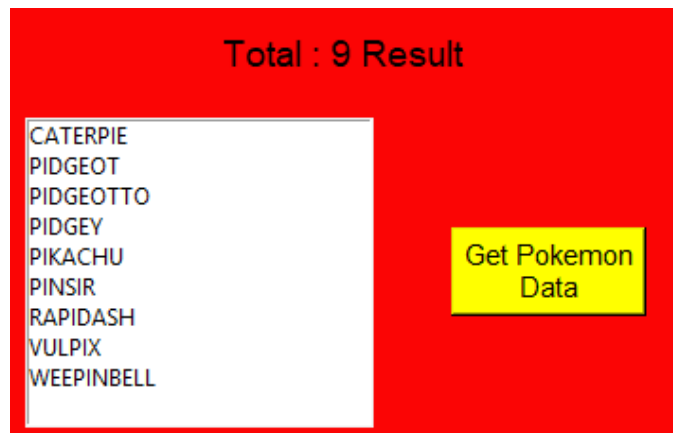
6. Once all the Pokemon data is collected, the user should be able to perform some basic searching and filtering on these data. (see Figure 5)



The image shows a red rectangular widget titled "Searching&Filtering". Inside, there is a white text input field containing the text "pika". Below the input field, the text "Filter by type" is displayed. Underneath this text is a dropdown menu with "Electric" selected and a downward arrow. To the right of the dropdown is a yellow button with the text "SEARCH" in black.

Figure 5. Searching&Filtering widgets

7. Search will proceed through the characters. For example, when the user searches the "cha", all the Pokemons that contain the characters "cha" must be listed. If no query is entered in the search entry, the user should only be able to filter without any search. (see Figure 2)
8. Filtering process will be done on Pokemon types. There are 17 types in total. You can obtain these types when you are crawling data and store them in a list. You must use Combobox for filtering. At the top of the Combobox, there must be the "all types" element (see Figure 2). When this element is selected, only a search can be conducted without any filtering. Another thing you need to pay attention is that some Pokemon can have more than one type. For example, Charizard is both Flying and Fire Pokemon. You should be able to get the Charizard in the results, when you filter for Fire Pokemon or Flying Pokemon.
9. If the user wants to list all the Pokemons, he/she should not enter any query and select all types via combobox.
10. When the filtering and searching process is completed, the user presses the **SEARCH** button. You should show the correct search results on the listbox. You should also show how many results are found through a label (see Figure 6).



The image shows a red rectangular widget titled "Total : 9 Result". Inside, there is a white listbox containing the following text: CATERPIE, PIDGEOT, PIDGEOTTO, PIDGEY, PIKACHU, PINSIR, RAPIDASH, VULPIX, and WEEPINBELL. To the right of the listbox is a yellow button with the text "Get Pokemon Data" in black.

Figure 6. Search Results widgets

11. When the user wants to access the data belonging to a specific Pokemon, he/she should choose the Pokemon from the listbox (Figure 2) and click **Get Pokemon Data** button. The GUI should then be expanded to show up the selected Pokemon data. (Figure 7)



Figure 7. Pokemon Data widgets

12. For showing the picture of Pokemon, you should use a canvas. First, download the image of the Pokemon to the computer by using the URL of that image, which you should have it in your nested dictionary. You can add the image that you have downloaded to the canvas with the create_image method of canvas. (see Figure 8 and Figure 9)
13. You must associate colors and types when showing the types of a Pokemon. That is, each type must have its own color. You can create a {type: color} dictionary for this. When choosing colors please pay attention to readability in GUI. (see Figure 9 and Figure 10)



Figure 8. Image and type widgets of Pokemon with two types.



Figure 9. Image and type widgets of Pokemon with one type.

14. You should show the following data for a Pokemon:
Name, id, image, types, height, weight, category, abilities and weakness (see Figure 2).
Please see Figure 10 to search for these information on the website.

The screenshot shows the Pokemon website's Pokedex entry for Charizard. The URL is <https://www.pokemon.com/us/pokedex/charizard>. The navigation bar includes Home, Pokedex, Watch Pokémon TV, Play Minigames, Trading Card Game, Video Games, and Play! Pokémon Events. The Pokedex entry for Charizard is displayed, with the name "Charizard" and ID "#006" highlighted. A dropdown menu shows "Charizard". The Charizard image is shown, with a description: "Charizard flies around the sky in search of powerful opponents. It breathes fire of such great heat that it melts anything. However, it never turns its fiery breath on any opponent weaker than itself." The Versions section shows two versions: a red one and a blue one. The Height is "5' 07\"", Weight is "199.5 lbs", and Gender is "♂ ♀". The Category is "Flame" and the Ability is "Blaze". The Type section shows "Fire" and "Flying". The Weaknesses section shows "Rock", "Electric", and "Water". The Stats section shows a bar chart for HP, Attack, Defense, Special Attack, Special Defense, and Speed.

Annotations in the image point to the following data:

- name: Charizard
- id: #006
- image: Charizard
- height: 5' 07"
- weight: 199.5 lbs
- category: Flame
- abilities: Blaze
- types: Fire, Flying
- weakness: Rock, Electric, Water

Figure 10. Pokemon Data in website.

Implementation Notes / Hints:

- Those who do not follow the **naming conventions** will get 10% off of their project grade.
- You won't need **mysearchengine.py** in any step. Therefore, **do not use** mysearchengine.py for this project.
- **You need to use at least 2 classes in this project.**
- **In order for you to understand better how the GUI should work, please check the GIF image or Video that will be provided with the project documents.**
- For the **Progress Bar** please look at Week 6 Practice Session.
- For **Pickle**, please see the following documentation:
<https://docs.python.org/2.7/library/pickle.html>
- For **Canvas.create_image** method, please see the following reference:
http://effbot.org/tkinterbook/canvas.htm#Tkinter.Canvas.create_image-method
- To display the image of a Pokemon, you may initially just extract the picture URL from the page source by using BeautifulSoup at the initial parsing phase, and record it in a nested-dictionary as mentioned above. If a user selects a Pokemon from the listbox, then you can get the URL of the image of a Pokemon from the dictionary, download and save the picture at that time. To download picture from URL, you may check the following link:
<https://stackoverflow.com/questions/40688391/save-image-with-urllib-urlretrieve>
- The size of the Pokemon's image can be large. If you want to resize you can use resize method from PIL, you may check the following link:
<https://pillow.readthedocs.io/en/3.1.x/reference/Image.html#PIL.Image.Image.resize>
- You can use the color chart when choosing colors for Pokemon types, you may check the following link:
http://www.science.smith.edu/dftwiki/index.php/Color_Charts_for_TKinter

Warnings:

- You **CANNOT** use place for geometry, only grid and pack are allowed.
- Do not talk to your classmates on project topics when you are implementing your projects. Do not show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. **Any similarity in your source codes will have serious consequences for both parties.**
- Carefully read the project document, and pay special attention to sentences that involve “should”, “should not”, “do not”, and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so may result in **plagiarism** investigation. Last but not the least, you need to understand code pieces that you may get some other resources. This is one of the goals of the mini projects.

Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is important to understand all the details in your submitted code.

How and when do I submit my project?

- Projects may be done individually or as a small group of two students (doing it individually is **strongly** recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for file naming details).
- Submit your own code in a single Python file. Name it with your and your partner’s first and last names. As an example, if your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do not use any Turkish characters in file name). If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
 - Those who do not follow the above naming conventions will **get** 10% **off** of their project grade.
- Unlike other MP projects, you are given three weeks instead of two weeks to work on this project. Submit it online on LMS by **17:00 on 26th December, 2018, Wednesday.**

Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions after which are up-to 24 hours late.
- -50%: Submissions which are up-to 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria?

GUI Design (25)	Crawling Pokemon Data and saving it on database for future use (25)	Searching and Filtering properly (15)	Showing Pokemon Data properly (15)	Displaying Pokemon Image properly (10)	Displaying Pokemon types with a specific color for each type (5)	Appropriate Use of Error Handling and Exceptions (5)
--------------------	--	--	---------------------------------------	---	---	---

Your code should be efficient, easy to follow and track. Therefore, from your overall grade, we will deduct points by the specified percentage for the following items:

- Inappropriate/Cryptic variable names and method names (10%)
- Classes and objects are not used properly (30%)
- Insufficient commenting (10%).

Have further questions?:

If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules.**

IMPORTANT NOTES :

Note 1: Plagiarism:

- Zero tolerance
- Cases will be referred to the Ethics Committee
- Both parties (provider and receiver) are responsible
- Process:
 - Automated computerized checks for pre-filtering
 - Human review for confirmation
 - Referral to the Ethics Committee if true positive