Practical Assignment 1: Scraping data with the Scrapy-Framework

Description: In this assignment, you will learn how to crawl data with Python's Framework Scrapy. Scrapy is a simple tool to make use of a website's HTML-structure and extract vital information. Here, you will extract whether a Pokemon is a legendary, a semi-legendary or a mythic (Label) and its unique Dex identifier. The data will be saved to an already existing Sqlite-Database for later use.

Assignment:

1. **Getting the project.** Use Git-Desktop to fetch the data from the global repository or type git pull in the command shell (only works if git is installed).

2. **Understanding the project**. You will get a new folder called extract_labels. Inside this folder you will find a simple scrapy project containing three main files:

    a. ./spiders/label_spider.py
    b. items.py
    c. pipelines.py

    You will have to make changes to the spider and the pipeline.

    First, look at the files and try to understand what is happening. Some code has already been written for you. Make use of the given structures and functions.

3. **Scraping the Dex identifier.** Inside the label_spider.py you will find three functions. Each function is meant to scrape a different type of Pokemon. Extract the *Dex* identifier from a Pokemon page (example https://www.serebii.net/pokemon/volcanion/). Use the response parameter and xpath. After you have successfully scraped the data, create a PokemonLabel item and store it inside the respective fields. Do not forget to store the label as well. We will need it later to classify the Pokemons. Lastly, yield the PokemonLabel item to forward it to a scrapy pipeline.

4. **Storing the data to a sqlite database**. Open the pipelines.py file. Inside, you will find a class LabelPipeline. A PokemonLabel item will be forwarded to this class and processed by the function process_item(…). The pipeline's task is to open a connection to the database, create a table and store each extracted item inside the database.

    a. First, create the table Label in the function create_table(self). The table should contain two columns. An integer for the *Dex* value and a varchar for the *label*. The

Dex entry will be the primary key and also a foreign key to Pokemon(Dex). Important: Do not change the name *Dex*.
After creating the table, execute the command with class attribute self.curr.

b. Next, store the item to the database with the function store_item(…). Remember that you can access an item like a python dictionary. Beware of the order in which you store the item values.

5. **Crawl the page**. Now, you should be able to crawl the webpage. Go to the parent folder extract_labels and open a command shell inside it. Inside the shell type:

```
scrapy crawl labelSpider
```

If everything works fine, you should see several links with a HTTP-Code 200.