

Lab 03 - Lists and tuples

Script Languages

Learning goals

1. Use lists and tuples to store and retrieve information.
2. Use `range()` to generate lists of numbers
3. Perform operations on strings: searching, slicing, replacing.

Exercises

1. Data preparation

1. Use the same approach as in the previous lab: read content of the web server log file from standard input. The file can be downloaded from eportal or you may find some online.
2. Place your main code into function `run()`
3. Add the code that starts function `run()` only if your Python file is run directly:

```
python my_app.py
```

4. Place your functions in separate, logically divided files and import them into main file.

2. Using lists and tuples

1. Create function `read_log` that reads all lines of the input at once and returns list containing entries from the log. Inside this function:
 1. Split each line (entry) into separate attributes.
 2. Convert strings to the proper data type e.g. resource name - string, HTML result code - int, and so on. Assume that each line of input log contains always correct number of items or is empty (contains only whitespaces or nothing at all).
 3. Store each entry (non-empty line) as a tuple in the list.
 4. Log as a DEBUG message number of lines read. Remember to add meaningful content in the message.
 5. Log as a DEBUG message number of entries in the list.

2. Create function `get_successful_reads` that:
 1. Takes as a parameter list of all entries in the log.
 2. Logs as an INFO message the number of entries in the resulting list.
 3. Returns a new list containing only resources successfully read (HTTP result codes 2xx).
3. Create function `get_failed_reads` that:
 1. Takes as a parameter list of all entries in the log.
 2. Creates two separate lists containing entries with HTTP codes 4xx and 5xx respectively.
 3. Takes optional argument to decide whether return merged list or two separate lists
 4. Merges both lists into one.
 5. Logs as an INFO message the number of entries with 4xx and (separately) with 5xx result codes,
 6. Returns appropriate result
4. Create function `get_entries_by_code` that:
 1. Takes as a parameter list of all entries in the log.
 2. Takes required argument with HTTP status code (e.g. `get_entries_by_code(200)`)
 3. Validate the given status code.
 4. Returns a list of entries with given status code.
 5. Sort the returned list by chosen attribute. What is the difference between using `sorted()` and `sort()`?
 6. Logs as an INFO message the number of entries.
5. Create function `get_entries_by_extension` that returns a list containing entries of successfully retrieved pages with the extension given as a parameter. For example, to retrieve all html entries `get_entries_by_extension("html")` it

should return entries with `/news/index.html`, but not `index.php` or `/feed/`). To retrieve successful reads use functions implements in earlier tasks.

6. Create function `print_entries` that prints a list containing entries given as its parameter.

3. Extended Version

1. Read PEP8 style guide (<https://www.python.org/dev/peps/pep-0008/>) and use it in your code.
2. Use list comprehensions.
3. Use a cookiecutter of your choice to structure your application.