Deeper Insights Coding Test

Nourdine Bah

Contents

1 What the solution does

The solution is a python script that takes the path of a file as an argument. The input file is a contiguous list of lines, the final line specifies the search term. The script searches the source text for matches of the search term, and outputs all the matches after removing unwanted characters.

2 Details about the solution

2.1 Support of UTF-8 only

In order to remove the unwanted characters, the script filters them by using their UTF-8 code. This way, the script supports accents and the developer using the API can specify as many characters as she/he wants. However, this requires to know the encoding table. So, only UTF-8 support was implemented at this stage.

2.2 File opening

Before opening the file, the script will check:

- if the file exists on the system
- if the file size doesn't exceed a certain value (value can be changed by the user as an option)
- if the file has adequate right access
- if the file can be open
- if the file is not a directory
- if the file is not a link (we decide to exclude links)
- if the file is encoded in UTF-8

2.3 Checking validity of the file content

The search term must be a single word. So, it can't match empty lines or lines with only spaces. Therefore, the script removes such lines. The script also checks that the search term (last line) is a single word.

2.4 About the documentation

The documentation was generated with `sphinx` and contains details about the modules and classes used to write the script. It can be generated by following the instructions below. If generated, the documention will be found in `deepintest/docs/build/html`.

3 How to install and run the solution

We assume that you have `unzip` and `python3` already installed on your system. We also assume that `black`, `pylint` and `pytest` python packages are installed.

3.1 Extract the content of the `zip` archive

First, extract the content of the `zip` archive by running:

```
$ unzip MachineLearningSoftwareEngineer_nourdine_bah.zip
```

This should create a folder called `deepintest` in your current directory.

3.2 Create package archive and generate documentation

Change directory to go to the `deepintest` folder and build the package by running:

```
$ cd deepintest
$ make build
```

The archive should be now visible in the `dist` directory:

```
$ ls dist
deepintest-0.0.0-py3-none-any.whl  deepintest-0.0.0.tar.gz
```

If you have `sphinx` and `sphinx-rtd-theme` installed, you can generated the documentation by running:

```
$ make doc
```

The generated documentation should be in `deepintest/docs/build/html`. You can now leave the `deepintest` directory and go back to where you were:

```
$ cd ../
```

3.3   Create a `virtualenv` environment

If you don't have `virtualenv` already on your system, please install it by running:

```
$ pip install virtualenv
```

Then, create a virtual environment called `venv` and source it by running:

```
$ virtualenv venv
$ . venv/bin/activate
```

You are now in the virtual environment.

3.4   Install the solution package

Now, install the solution package by running:

```
(venv) $ pip install deepintest/dist/deepintest-0.0.0.tar.gz
```

The solution is now installed. You can display information about the script by running:

```
solution --help
```

3.5   Run the solution

To run the solution on a file you can run:

```
solution <PATH_OF_THE_FILE>
```

3.6   Leave the virtual environment

When you are done, you can leave the virtual environment by running:

```
deactivate
```