In this assignment you will:

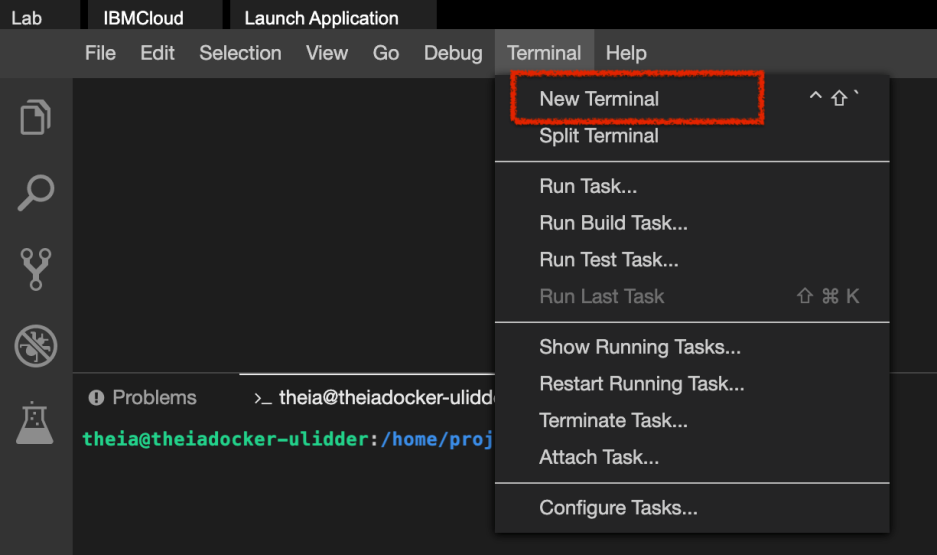
* Use the deep\_translator python package for the translation.
* Create a function that translates English to French.
* Create a function that translates French to English.
* Run coding standards check against the functions above.
* Write unit tests to test the above functions.
* Run unit tests and interpret the results.
* Package the above functions and tests as a standard python package.

***Important Notice****- Please keep in mind that sessions for this lab environment are not persisted. If you will not be completing the entire lab in one sitting, please save your code outisde of this environment, so you can resume your work without loss.*

*👉 Click here for instructions to save your code on github*

**Task1: Write a function that translates English text to French in translator.py**

1. Open a terminal window by using the menu in the editor: Terminal > New Terminal.



1. Go to the project home directory.
   1. 1
   2. cd /home/project

Copied!

1. Run the following command to Git clone the project directory from the clone URL you had copied in the prework lab.
   1. 1
   2. [ ! -d 'xzceb-flask\_eng\_fr' ] && git clone <paste\_your\_repo\_name>

Copied!

1. Change to the final\_project folder.
   1. 1
   2. cd /home/project/xzceb-flask\_eng\_fr/final\_project

Copied!

1. Create folder named machinetranslation and change to that directory.
   1. 1
   2. 2
   3. mkdir machinetranslation
   4. cd machinetranslation

Copied!

1. Run the following command to make sure that python3 is using version 3.8.
2. 1
3. sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.8 10

Copied!

1. Run the following command to check the python version.
2. 1
3. python3 --version

Copied!

*Note: It should be Python 3.8.0.*

1. Install the packages that you will be using in this code, namely deep\_translator and Flask.
   1. 1
   2. 2
   3. python3 -m pip install deep\_translator
   4. python3 -m pip install Flask

Copied!

1. In the explorer, go to the machinetranslation directory and create a new file called translator.py. Enter the following line of code.
   1. 1
   2. from deep\_translator import MyMemoryTranslator

Copied!

*📷 Take a screenshot of your import statement and save it as a .jpg or .png with the filename import\_translator. You will be prompted to upload the screenshot in the Peer Assignement that follows.*

1. Add function **englishToFrench** which takes in the englishText as a string argument, in translator.py. Use the instance of the MyMemory Translator you imported previously, to translate the text input in English to French and return the French text.
   1. 1
   2. 2
   3. 3
   4. def englishToFrench(englishText):
   5. #write the code here
   6. return frenchText

Copied!

📷 Take a screenshot of your functions and save it as a .jpg or .png with the filename e2f\_translator\_function. You will be prompted to upload the screenshot in the Peer Assignement that follows.

1. Add function **frenchToEnglish** which takes in the frenchText as a string argument, in translator.py. Use the instance of the MyMemory Translator you imported previously, to translate the text input in French to English and return the English text.
   1. 1
   2. 2
   3. 3
   4. def frenchToEnglish(frenchText):
   5. #write the code here
   6. return englishText

Copied!

*📷 Take a screenshot of your functions and save it as a .jpg or .png with the filename f2e\_translator\_function. You will be prompted to upload the screenshot in the Peer Assignement that follows.*

# ****Task 2:**** Write the unit tests for English to French translator and French to English translator function in tests.py

1. Create a new file called tests.py in the machinetranslation directory.
2. Write at least 2 tests in tests.py for each method
3. Test for the translation of the word ‘Hello’ and ‘Bonjour’.
4. Test for the translation of the word ‘Bonjour’ and ‘Hello’.
5. Take a screenshot of your unit tests and save it as a .jpg or .png with the filename translation\_unittests.

# ****Task 3:**** Check your code against python coding standards

1. At the terminal run the following command to install pylint.
   1. 1
   2. python3 -m pip install pylint

Copied!

1. Run pylint translator.py to check the coding standard compliance in your code. Refer to this [exercise](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0222EN-SkillsNetwork/labs/module%206/Lab%20-%20Static%20Code%20Analysis/Static_Code_Analysis.md.html) you did earlier, if needed.
2. Make sure your rating is at least 7.
3. 📷 Take a screenshot of the output of the pylint analysis report showing your score and save it as a .jpg or .png with the filename pylint\_score.

# ****Task 4:**** Run tests

1. At the terminal run the command
   1. 1
   2. python3 tests.py

Copied!

1. 📷 Take a screenshot of test results and save it as a .jpg or .png with the filename **unit\_test\_results**.

PreviousNext

# ****ask 5:**** Package the above functions and tests as a standard python package.

1. Create \_\_init\_\_.py file in the directory machinetranslation.
2. Create a folder called tests under the newly created folder
3. Copy the unit tests into the tests folder

*📷 Take a screenshot of the folder structure of the package ( From the menu go to View -> Explorer to set the explorer view) and save it as a .jpg or .png with the filename package\_folder\_structure.*

Previous