**Overview**

For the program the chat functions are broken out to the prompts to elicit the source and destination languages from the user, and the translation function that does the work of translating. This allows for adjustments of the prompts, ability to accept other languages and could do preprocessing work before sending the parameters to the function that does the work without having extra processing or statements. Keeping each function focused on only what it needs to do makes the code easier to read, keeps the functions limited to what they need to do, and keeps any variables they need scoped only to the function and not need to be tied to any global variables.

With the prompting function a listing of different languages could be provided, or the list of available languages could be presented, or a different prompt used to display them, so that the user is not tied to any hardcoded language. The translation function could be tied to English only as the destination language, or as source, providing a specific translation function and the user would only need to provide the alternate language needed. Being more open, as in the coding done, the translation function is more abstract and supports any variable needed.

With the addition of the PyDoc in the functions they are well documented, and any Developer could use the functions and know what is required for the parameters.

For testing the following were used:

|  |  |  |  |
| --- | --- | --- | --- |
| **Source Phase** | **Source Language** | **Dest Language** | **Result** |
| Bonjour | Fr | En | Good morning |
| Bonjour | Fr | Es | Buenos Dias |
| I like cake | En | Fr | J’aime le gateau |
| I like cake | En | Es | Me gusta el pastel |
| J’aime le gateau | Fr | Jp | 私はケーキが好きです |

Did not notice any issues with any of the values, and being able to translate words or phrases among many languages went well. Even using one, Japanese, that I did not list as a language in the prompts, but knew the code for.