**Цель:** С помощью сторонних библиотек Python создать переводчик.

**Ход Работы:** Создаём питоновский файл. Импортируем некоторые инструменты из Flask.

from flask import Flask, render\_template, request

Далее импортируем библиотеку переводчика

from deep\_translator import GoogleTranslator

Выбираем библиотеку, которая использует Google переводчик, так как это один из самых популярных переводчиков. Выбираем именно эту библиотеку, так как googletrans (другая популярная библиотеку по гугл-переводчику) недоступна (проблемы с отправлением слов на гугл-переводчик в момент выполнения работы).

Далее создаём приложение на Flask.

app = Flask(\_\_name\_\_)

Задаём декоратор для пути «/» с методами Post и Get для отправления запросов на HTML-файл и получения их обратно.

|  |
| --- |
| @app.route('/', methods=['post', 'get']) |
|  | def trans(): |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | return render\_template("trans.html") |

И в конце файла запускаем приложение.

|  |
| --- |
| if \_\_name\_\_ == '\_\_main\_\_': |
|  | app.run() |

Теперь создаём HTML файл и заполняем его.

|  |
| --- |
| <!DOCTYPE HTML> |
|  | <html> |
|  | <head xmlns="http://www.w3.org/1999/html"> |
|  | <meta charset="UTF-8"> |
|  | <title>Python-Translator</title> |
|  | </head> |
|  | <body> |
|  | <br> |
|  | <h1 align="center">Python-Translator by Iudin Artyom</h1> |
|  | <br> |
|  |  |
|  | <form action="" method="post"> |
|  | <p><select size="5" multiple name="lang"> |
|  | <option disabled>Chose language</option> |
|  | <option selected value="en">English</option> |
|  | <option value="ru">Russian</option> |
|  | <option value="es">Spanish</option> |
|  | <option value="de">German</option> |
|  | </select></p> |
|  | <p style="padding:20px"> |
|  | <label style="padding:20px">Enter your message</label> |
|  | <input name="text" type="text"> |
|  | </p> |
|  | <input type="submit"> |
|  | <br> |
|  | <h3>{{attention}}</h3> |
|  | <h3>{{message}}</h3> |
|  | </br> |
|  | </form> |
|  | </body> |
|  | </html> |

Используем переменные attention для предупреждения об ошибках (например, неправильно введённый текст). Выбор языков делаем через select с выбранным языком по-умолчанию, чтобы избежать ошибки невыбранного языка и дать пользователю выбор имеющихся языков.

Возвращаемся в python-файл. Модифицируем декоратор.

|  |
| --- |
| def trans(): |
|  | message = "" |
|  | attention = '' |
|  | if request.method == "POST": |
|  | lang = request.form.get('lang') |
|  | text = request.form.get('text') |
|  | if text == '': |
|  | attention = 'Incorrect text, try again' |
|  | else: |
|  | message = GoogleTranslator(source='auto', target=lang).translate(text) |
|  | return render\_template('trans.html', attention=attention, message=message) |
|  | return render\_template("trans.html") |

Если нам передают текст и язык через метод Post, то мы заполняем message сообщением, переведённое с помощью библиотеки. Source – определяет изначальный язык сообщения (задаётся автоматически); target – язык на который будет перевод, задаём из выбранных пользователем; у метода translate в скобках указываем сообщение, которое хотим перевести. Передаём в HTML-Файл на рендер сообщения.

**Вывод:** В ходе проделанной работы мы создали библиотеку с помощью Python-Библиотек Flask и deep\_translator