

# Guide

May 25, 2020

## 0.1 Write a Flask App

Practice writing a Flask App!

### 0.1.1 Exercise 1

- Write a Flask App that outputs 'Hello World!' when you run the application.
- Run the application using `FLASK_APP=app flask run`, in debug mode.
- Try using the method outlined above to run the application using `$ python3 app.py` instead, using the `if __name__ == '__main__':` method.

### 0.1.2 Exercise 2

- Create a Postgres database for your application
- Import Flask-SQLAlchemy into your Flask app
- Connect from SQLAlchemy to your Postgres database on your local machine.
- This local machine's user is called `student`, with no password.
- Use the default database port to connect
- Create an instance of the SQLAlchemy class, that takes your flask app, and set it equal to a variable called `db`:

```
db = SQLAlchemy(app)
```

### 0.1.3 Exercise 3

- Create a SQLAlchemy model, `Person`, with custom table name `persons`, that includes ID and name attributes, in the server script `app.py`.
- Have SQLAlchemy create the `persons` table if it doesn't exist already, whenever the server is run.
- Run the server.
- In the terminal, check that SQLAlchemy ORM successfully created the table by connecting to the database using `psql`.

### 0.1.4 Exercise 4

- Run the script, so that the `persons` table exists.
- Run the application, with debug mode on.

- Create a person record in the persons table, by connecting to psql and using `INSERT INTO`.
- Change the index route from saying "Hello World!" to saying "Hello" to the name of a person in the persons table.
- Preview the app in the browser, and see it output "Hello" next to the name of the person record in the database.

In [ ]: