The `flask` Package

Reference:

- Website
- Docs
- Source
- Quick-start Guide
- Tutorial
- Jinja Templates

The `flask` package provides a micro-framework for making applications with web interfaces (a.k.a "web applications").

Run a `flask` application "in development" using a web server on your local machine, and/or "in production" using a remote web server hosted by a provider like Heroku. If you run it in development, you should be able to use it by visiting `localhost:5000` in a browser, whereas if you run it in production, you should be able to use it by visiting the production server's URL.

Installation

First install `flask`, if necessary:

```
pip install flask
```

Usage

Follow the Official Flask Tutorial or the "Web Application" Exercise for an introduction to making web applications with Flask.

Also reference these example applications by the professor and previous students:

App Name	Functionality Description
Rock-Paper-Scissors (Flask)	Prompts the user to select an option from a dropdown menu, then processes that selection and displays the results on another page.
Starter Web App	A basic navigable web application with examples of capturing user inputs through a web form.
Starter Web App w/ Google Sheets datastore	A basic navigable web application with examples of reading and writing data to and from a Google Sheets datastore.

App Name	Functionality Description
DineCision by @jessicalee	Prompts the user to input their zipcode via a web form so it can use the Yelp API to provide a dining recommendation.
GMR's Algo Machine by @gmr50	Prompts the user to login to their Spotify account so it can create new Spotify playlist recommendations on their behalf.
Time Tracker by @kyokang1	Evaluates your work-life balance status and warn you if you work too hard. Interfaces with Google Sheets.
Products API	A JSON API with no front-end interface, uses CSV datastore.
Salad System	Includes a front-end interface and uses an SQL database as a datastore.