Cheatsheets / **Introduction to Flask**

Introduction to Flask

Python Flask Framework

Flask is a popular Python framework for developing web applications. It comes with minimal built-in functionalities and requirements, making it simple to get started and flexible to use. The developer has the liberty to customize most aspects of the app, including database integration, accounts and authentication, and more.

Creating Flask App Object

The Python `flask` module contains all the classes and functions needed for building a Flask app. The `Flask` class can be imported to create the main application object. It takes the name of the app as an argument.

```python
# Import Flask class
from flask import Flask


# Create Flask object
app = Flask(__name__)
```

Running Flask App

A Flask app can be run by exporting the `FLASK_APP` environment variable and running `flask run` in the terminal.

```
$ export FLASK_APP=app.py
$ flask run
```

Creating a Route

Routes in a Flask app can be created by defining a view function and associating a URL with it using the `route()` decorator. Routes specify how the Flask app handles requests it receives, such as what to display on the webpage at a certain URL.

```python
@app.route('/')
def hello_world():
    return 'Hello, World!'

# Output: The text `Hello, World!` is
displayed at the URL path '/'
```

Returning HTML From Route

In a Flask app, HTML can be returned from a view function to be rendered on a webpage. The HTML can be returned as a string.

```
@app.route('/')
def hello_world():
    return '<h1>Hello, World!</h1>'

# Output: The text `Hello, World!` is displayed as a level 1 heading at the URL path '/'
```

Variable Rules

Variable rules allow a Flask app to respond to dynamic URLs. Variable sections of a URL can be indicated by angular brackets and an optional converter: `<converter:variable_name>`. These variable parts will then be passed to the view function as arguments.

```
# Responds to `/page/1`, `/page/20`, etc.
@app.route('/page/<int:pg_num>')
def content(pg_num):
    return f'<h1>Displaying results for page {pg_num}</h1>'
```