Flask Cheatsheet

Importing Flask

```python
from flask import Flask
```

Most Common Imports

Frequently used imports for building Flask applications:

```python
from flask import Flask, render_template, redirect, url_for, request, flash, session, jsonify
```

Basic Boilerplate Code

A minimal Flask app structure:

```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

if __name__ == "__main__":
    app.run(debug=True)
```

> **Tip:** Always wrap `app.run()` inside `if __name__ == "__main__":` for best practice.

Creating Routes

Create endpoints in your Flask app:

```
@app.route("/")  # Home page
@app.route("/about")  # About page
```

Setting Allowed Methods

Specify allowed HTTP methods:

```
@app.route("/submit", methods=['GET', 'POST'])
def submit():
    if request.method == "POST":
        return "Form submitted!"
    return "Send a POST request!"
```

Auto Reload During Development

Enable auto-reloading and debugging:

```
app.run(debug=True)
```

Changing Host & Port

Change the default host/port:

```
app.run(host='0.0.0.0', port=5001)
```

Flask Templates

Render HTML files from `templates/` folder:

```
@app.route("/")
def home():
    return render_template("index.html", name="Harry")
```

Pass variables to template:

```
<!-- index.html -->
<h1>Hello, {{ name }}!</h1>
```

Redirect & URL Building

```
return redirect(url_for("home"))  # Redirects to home route
```

Request Data

Access form and query parameters:

```
request.method
request.form["username"]   # POST form data
request.args.get("page")   # GET query param
```

Flash Messages

Show temporary messages to users:

```
app.secret_key = "your-secret-key"
flash("Data saved successfully!", "success")
```

Use in template:

```
{% with messages = get_flashed_messages(with_categories=true) %}
  {% for category, message in messages %}
    <div class="{{ category }}">{{ message }}</div>
  {% endfor %}
{% endwith %}
```

Sessions (Store Data Between Requests)

```
session["user"] = "Harry"
print(session.get("user"))
session.pop("user", None)
```

Returning JSON

```python
@app.route("/api")
def api():
    return jsonify({"status": "ok", "data": [1, 2, 3]})
```

SQLAlchemy Integration

Import & Setup

```python
from flask_sqlalchemy import SQLAlchemy

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
```

Creating a Model

```python
class TableName(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    column_1 = db.Column(db.String(80), nullable=False)
    column_2 = db.Column(db.String(12), nullable=False)

    def __repr__(self):
        return f"<TableName {self.column_1}>"
```

Creating Database

```python
with app.app_context():
    db.create_all()
```

CRUD Operations

Create / Add Data:

```
entry = TableName(column_1="Hello", column_2="World")
db.session.add(entry)
db.session.commit()
```

Read Data:

```
data = TableName.query.all()        # Get all rows
data = TableName.query.first()      # Get first row
data = TableName.query.filter_by(column_1="Hello").all()  # Filter
```

Update Data:

```
entry = TableName.query.first()
entry.column_1 = "Updated Value"
db.session.commit()
```

Delete Data:

```
db.session.delete(entry)
db.session.commit()
```

Error Handling

```
@app.errorhandler(404)
def page_not_found(e):
    return render_template("404.html"), 404
```

Flask Docs

- Flask Official Documentation
- Flask SQLAlchemy Documentation