Python REST API with Flask - Cheat Sheet

1. Setup Flask

```
from flask import Flask, request, jsonify
app = Flask(__name__)
```

2. HTTP Methods

```
@app.route('/api/resource', methods=['GET', 'POST', 'PUT', 'DELETE', 'PATCH'])
```

3. GET

```
@app.route('/api/items', methods=['GET'])
def get_items():
    return jsonify({"data": [1, 2, 3]})
```

4. POST

```
@app.route('/api/item', methods=['POST'])
def create_item():
    data = request.get_json()
    return jsonify({"received": data}), 201
```

5. PUT

```
@app.route('/api/item/<int:item_id>', methods=['PUT'])
def update_item(item_id):
    data = request.get_json()
    return jsonify({"id": item_id, "updated": data})
```

6. PATCH

```
@app.route('/api/item/<int:item_id>', methods=['PATCH'])
def patch_item(item_id):
    data = request.get_json()
    return jsonify({"id": item_id, "patched": data})
```

7. DELETE

```
@app.route('/api/item/<int:item_id>', methods=['DELETE'])
def delete_item(item_id):
    return jsonify({"id": item_id, "status": "deleted"})
```

8. Common Functions

```
request.get_json()            # JSON body as dict
request.args.get("param")     # Query param
request.headers.get("Name")   # Header value
request.method                # HTTP method
jsonify(obj)                  # JSON response
```

9. Curl Examples

```
----------------
GET:
curl http://localhost:5000/api/items

POST:
curl -X POST -H "Content-Type: application/json" -d '{"name":"Siva"}'
http://localhost:5000/api/item

PUT:
curl -X PUT -H "Content-Type: application/json" -d '{"name":"Updated"}'
http://localhost:5000/api/item/1

PATCH:
curl -X PATCH -H "Content-Type: application/json" -d '{"email":"a@b.com"}'
http://localhost:5000/api/item/1

DELETE:
curl -X DELETE http://localhost:5000/api/item/1

10. Run Flask App
------------------
if __name__ == '__main__':
    app.run(debug=True)
```