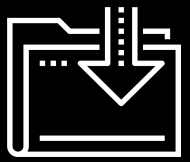
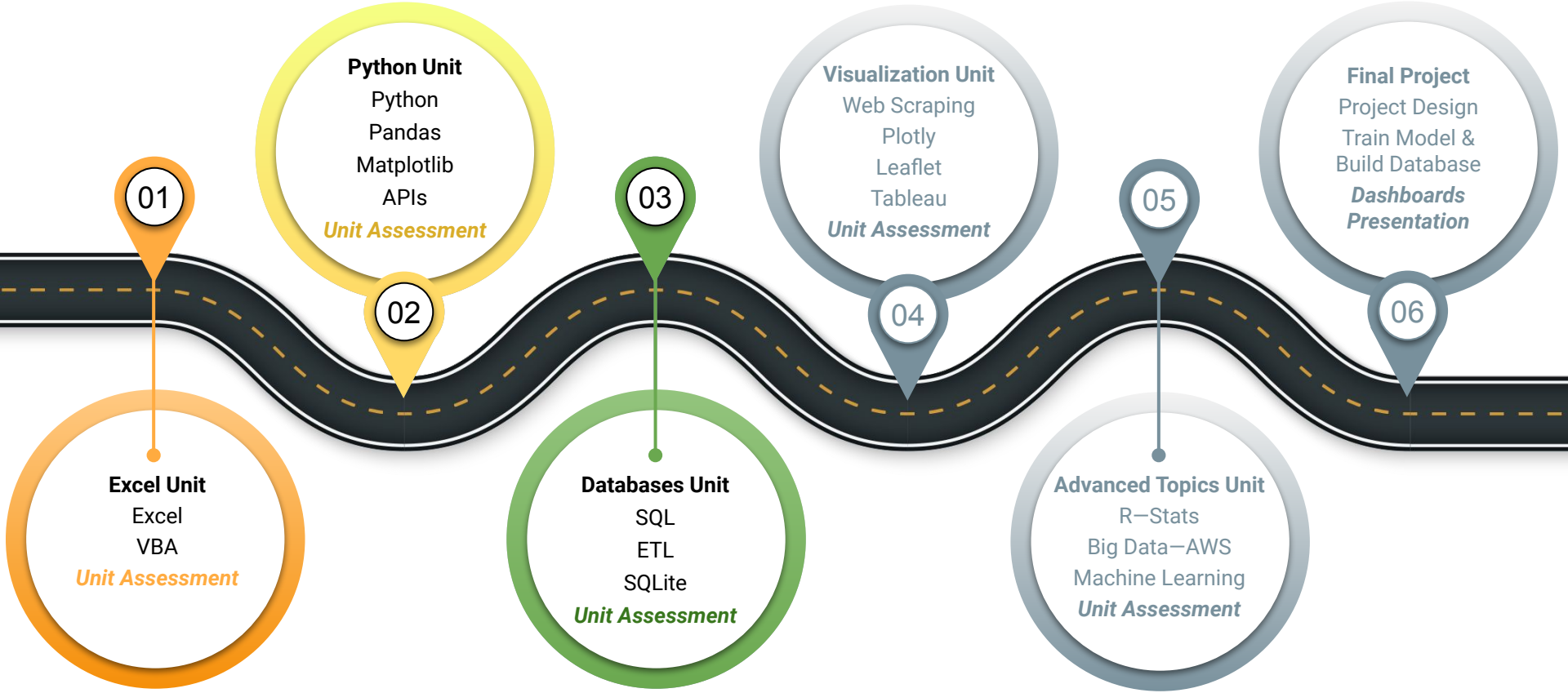
Working with Flask to Create a Web Application

Data Boot Camp

Lesson 9.2



The Big Picture



This Week: SQLAlchemy

By the end of this week, you'll know how to:

Use Flask to create and run a server

Retrieve data from a database using GET requests

Execute database queries on behalf of the client

Return JSONified query results from API endpoints and populate a webpage

This Week's Challenge

Using the skills learned throughout the week, students will query a SQLite database table, retrieve all the temperatures for the months of June and December, create DataFrames from these queries, and then generate summary statistics.

Module 9

Today’s Agenda

Today's Agenda

By completing today's activities, you'll learn the following skills:

01 Create a Flask web application with API routes

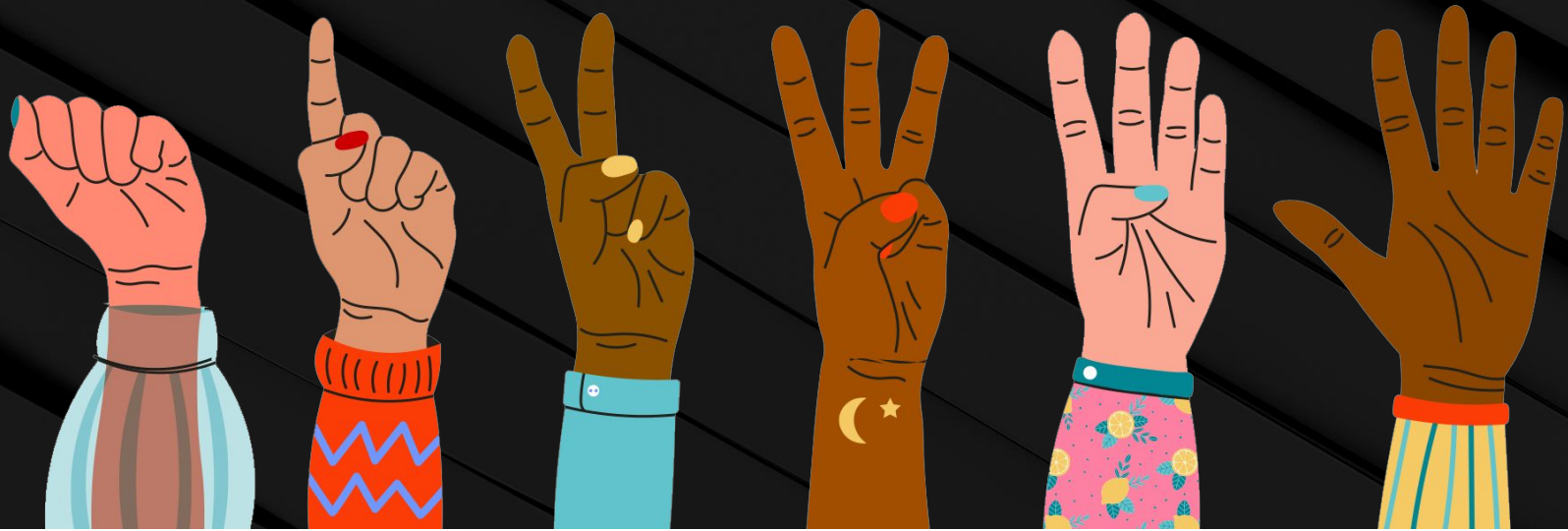
02 Jsonify data from a SQL database

03 Render Jsonified data to a webpage through the Flask application

Make sure you've downloaded any relevant class files!

FIST TO FIVE:

How comfortable do you feel with this topic?



Time to Code

Emoji Plotting

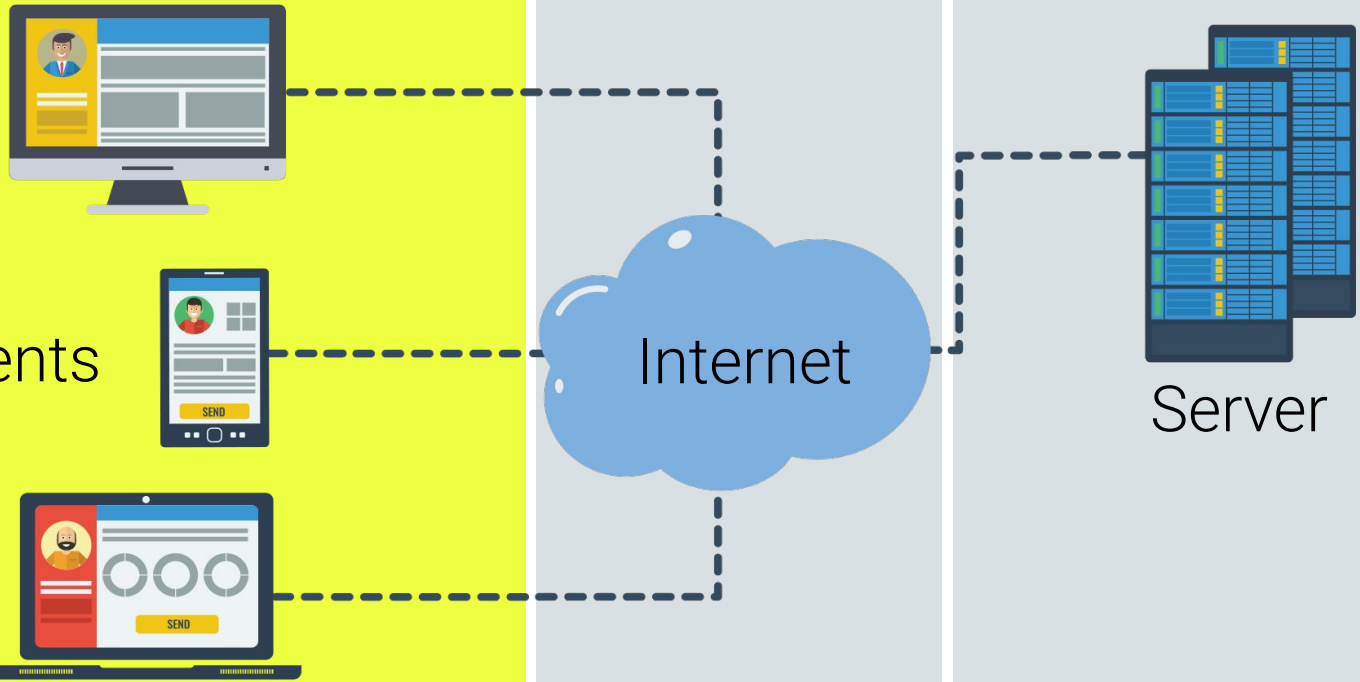
Suggested Time:

15 minutes

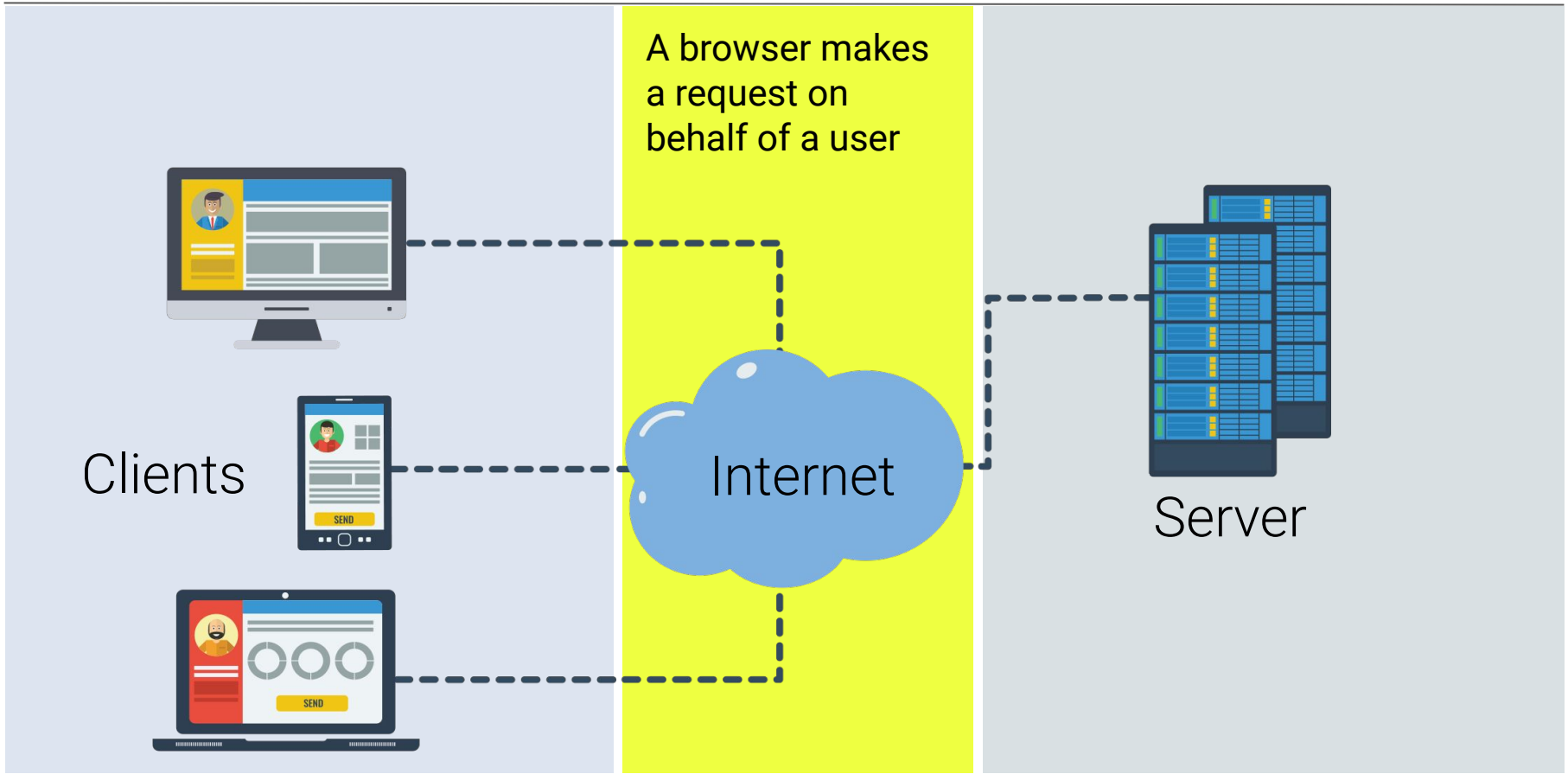
Introduction to Flask

Internet is Built from Clients and Servers

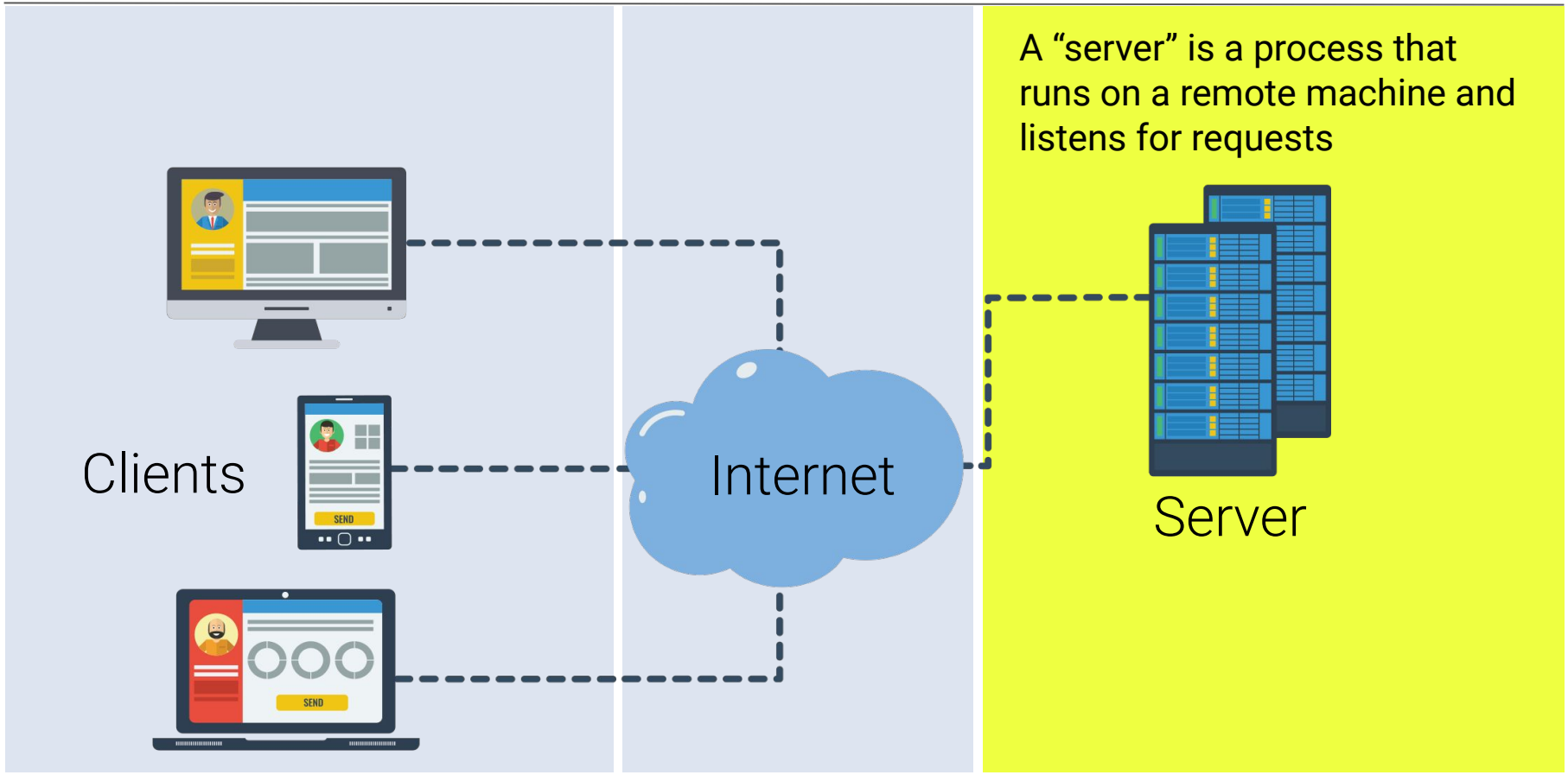
The application or device that is asking for information is called a “client”



Internet is Built from Clients and Servers



Internet is Built from Clients and Servers

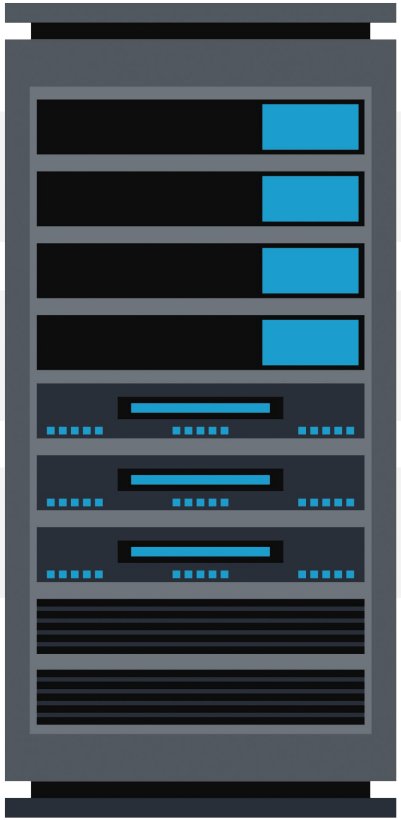


A server is essentially a program.

Server

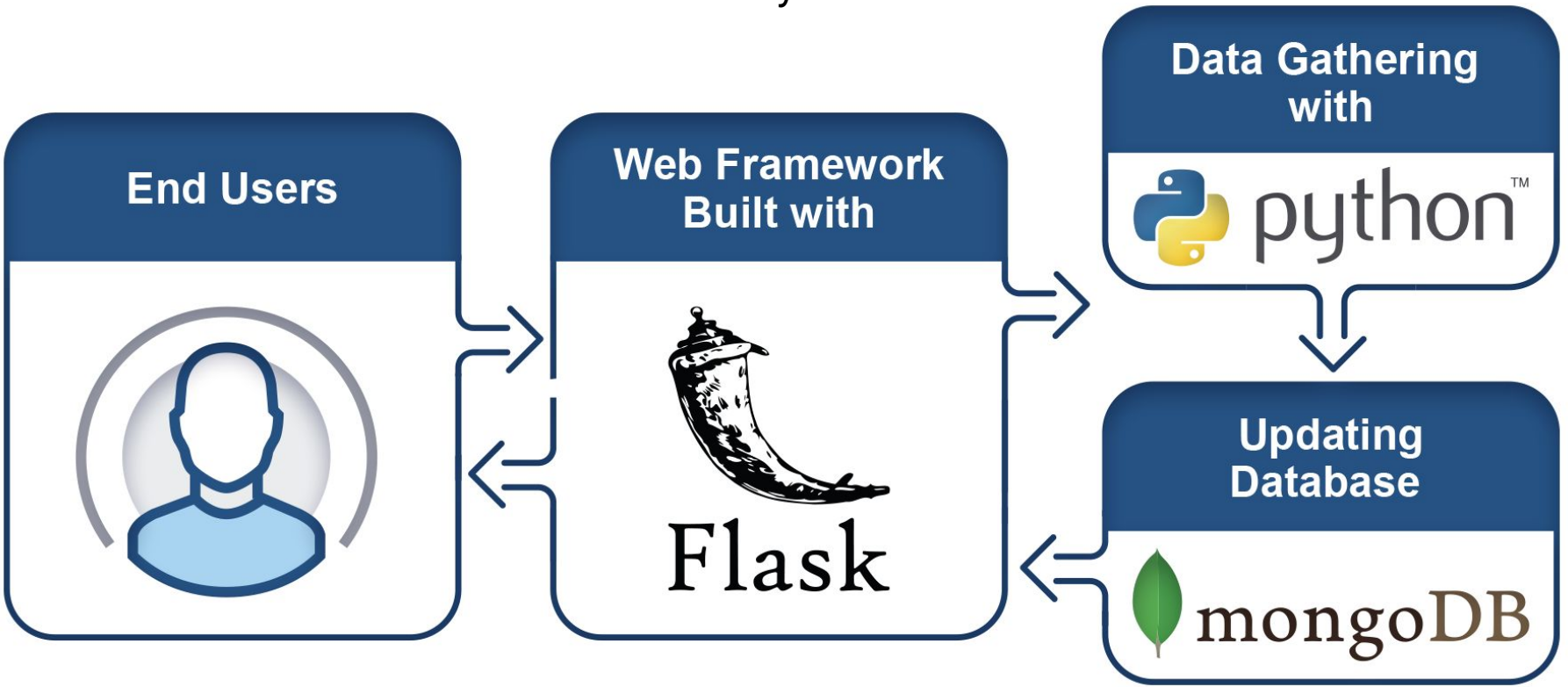
A server is essentially a program.

- We can write the code that runs a server
- We can determine what data is displayed
- We can determine what data is shared



Flask

Flask is a micro web framework to build your own APIs!



Instructor Demonstration

First Steps with Flask

Questions?

JSON APIs with jsonify

JSON APIs with jsonify

String Responses

All of the routes that were written in the previous activity have returned string responses.

Return JSON Data

The APIs we've dealt with in this course do not return raw text; instead, they return JSON data.

Flask has a built-in method, called `jsonify`, to automatically convert a dictionary into a properly formatted JSON response.

Flask Has a Function to Create JSON Responses

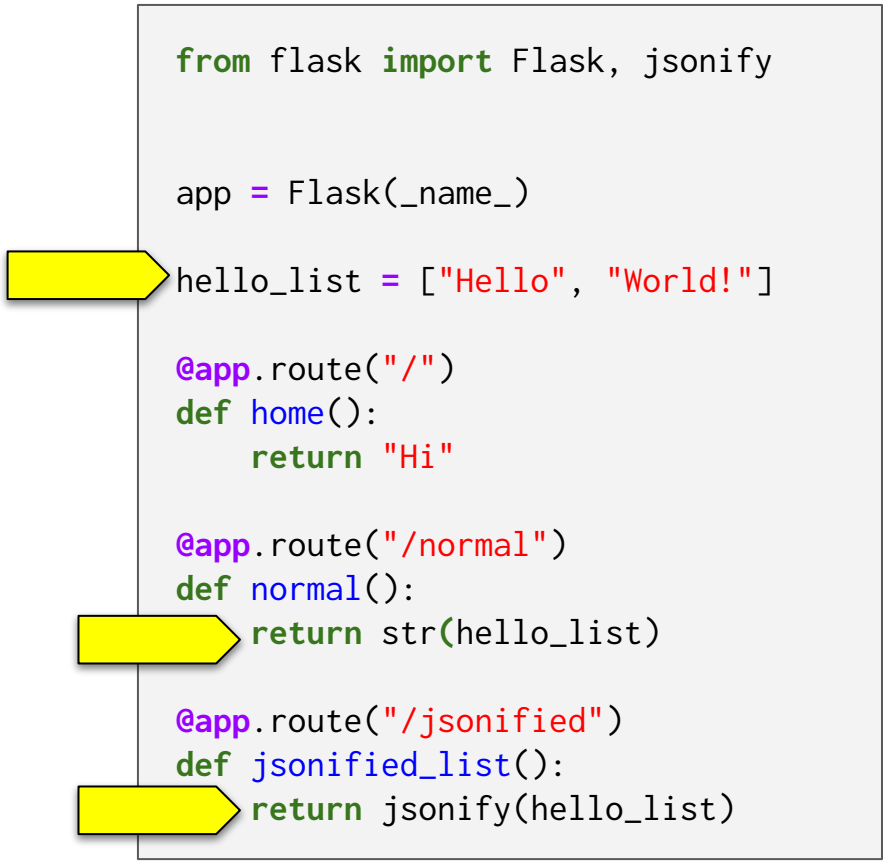
We cannot simply return a list response directly through Python.

Routes must return HTTP responses.

```python
from flask import Flask, jsonify


app = Flask(_name_)

hello_list = ["Hello", "World"]

@app.route("/")
def home():
    return "Hi"

@app.route("/normal")
def normal():
    return str(hello_list)

@app.route("/jsonified")
def jsonified_list():
    return jsonify(hello_list)
```

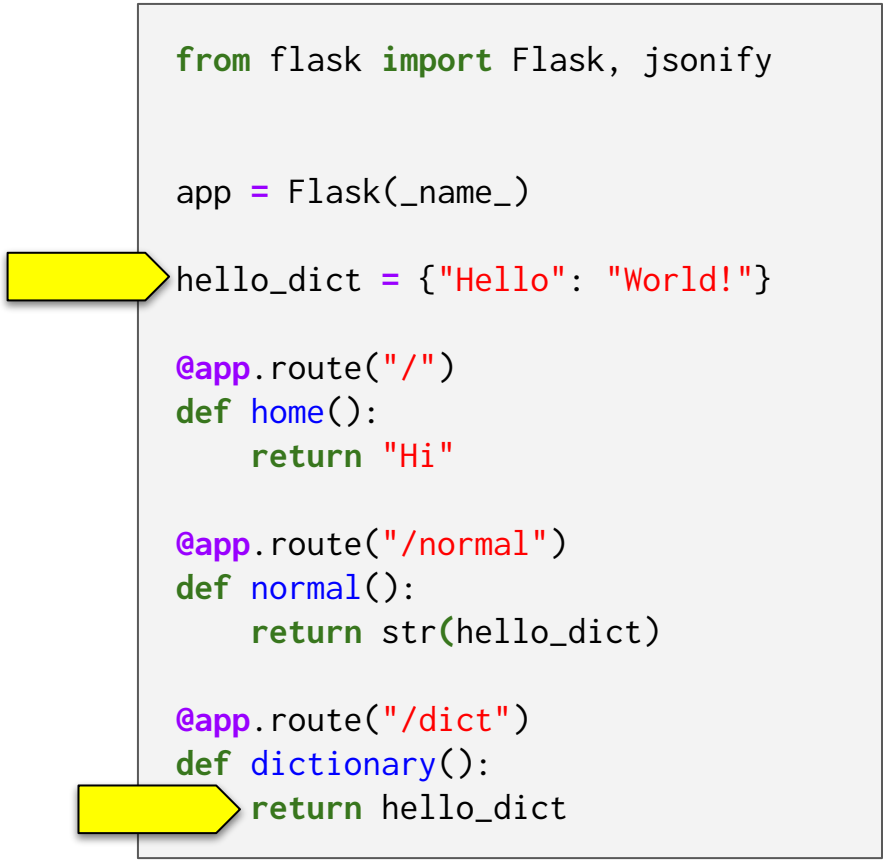
Flask Has a Function to Create JSON Responses

`jsonify` automatically converts Python lists into JSON responses.



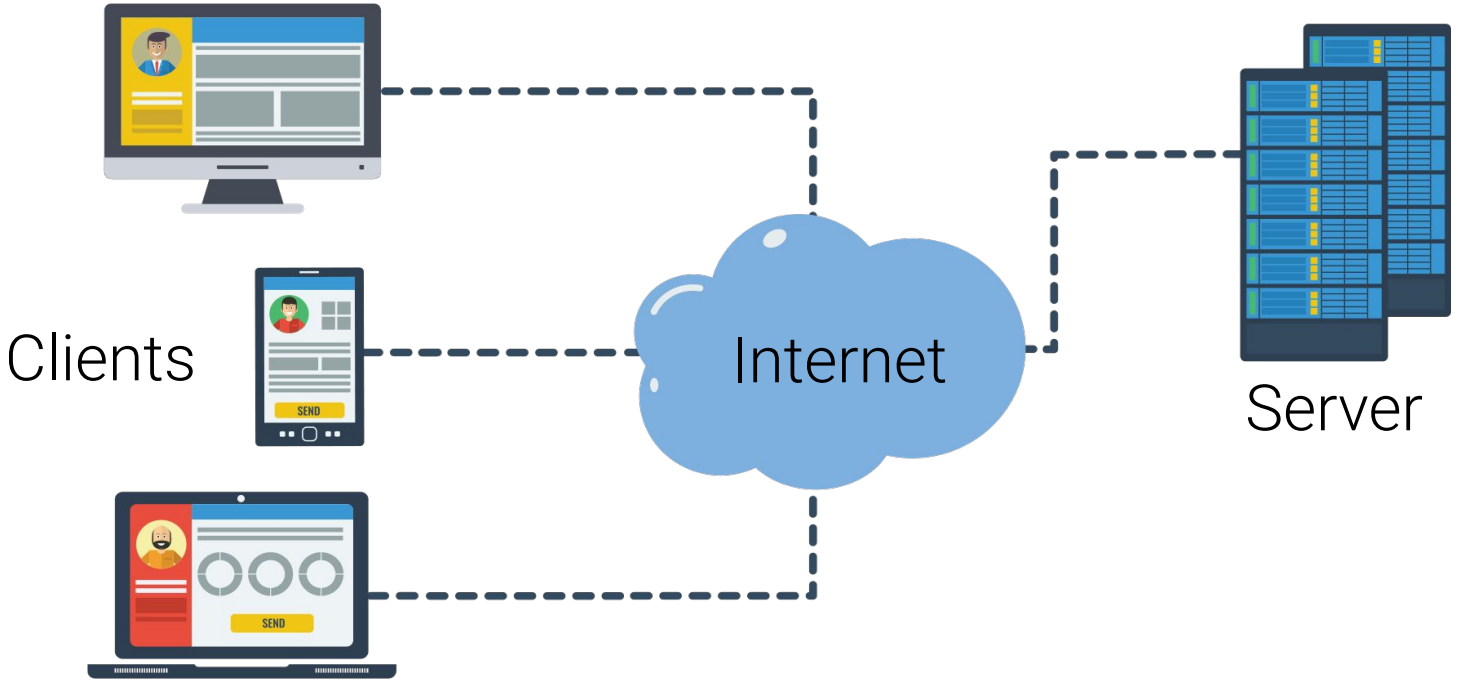
Flask has a function to create JSON responses

When converting Python dictionaries into JSON responses, `jsonify` is not needed.



Flask Has a Function to Create JSON Responses

The converted JSON responses are wrapped in HTTP to send back to the client.



Instructor Demonstration

Jsonify

Time to Code

Hello, Web

Suggested Time:

15 minutes

Questions?

Justice League jsonify

Suggested Time:

15 minutes

Routes with Variable Rules

Our Current API Is One-Dimensional

Our current API can only return the entire Justice League dataset.

```
@app.route("/api/v1.0/justice-league")
 def justice_league():
     """Return the justice league data as json"""

      return jsonify(justice_league_members)
```

Our Current API Is One-Dimensional

Ideally, clients can send a request for a character and expect either:

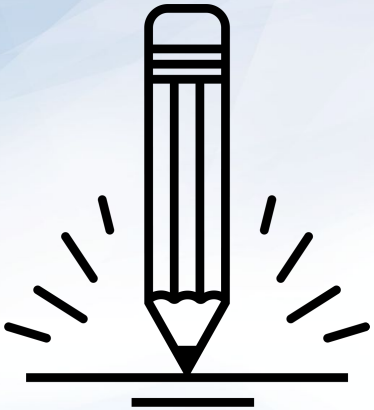
01 A JSON response with only specific character information; or

02 A detailed error response

```
return {"error": f"Character with real_name
{real_name} not found."}, 404
```

Instructor Demonstration

Routes with Variable Paths



Activity: Routes with Variable Rules

In this activity, you will practice reflecting an existing database by using SQLAlchemy on a SQLite table that contains demographic data.

Suggested Time:
10 minutes

Let's Review

Flask with ORM

Any useful API must make queries against datasets that are too large to load into memory.

Flask with ORM

SQLAlchemy can be used to perform queries based on a Flask route.

01

Convert the query into a dictionary, then into a JSON with `jsonify`

02

Return the JSON query to the endpoint

Instructor Demonstration

Flask with ORM

Questions?

Chinook Database Analysis

Suggested Time:

15 minutes