

# *APIs*

012720 Data Science  
February 5th, 2020

// **FLATIRON SCHOOL**

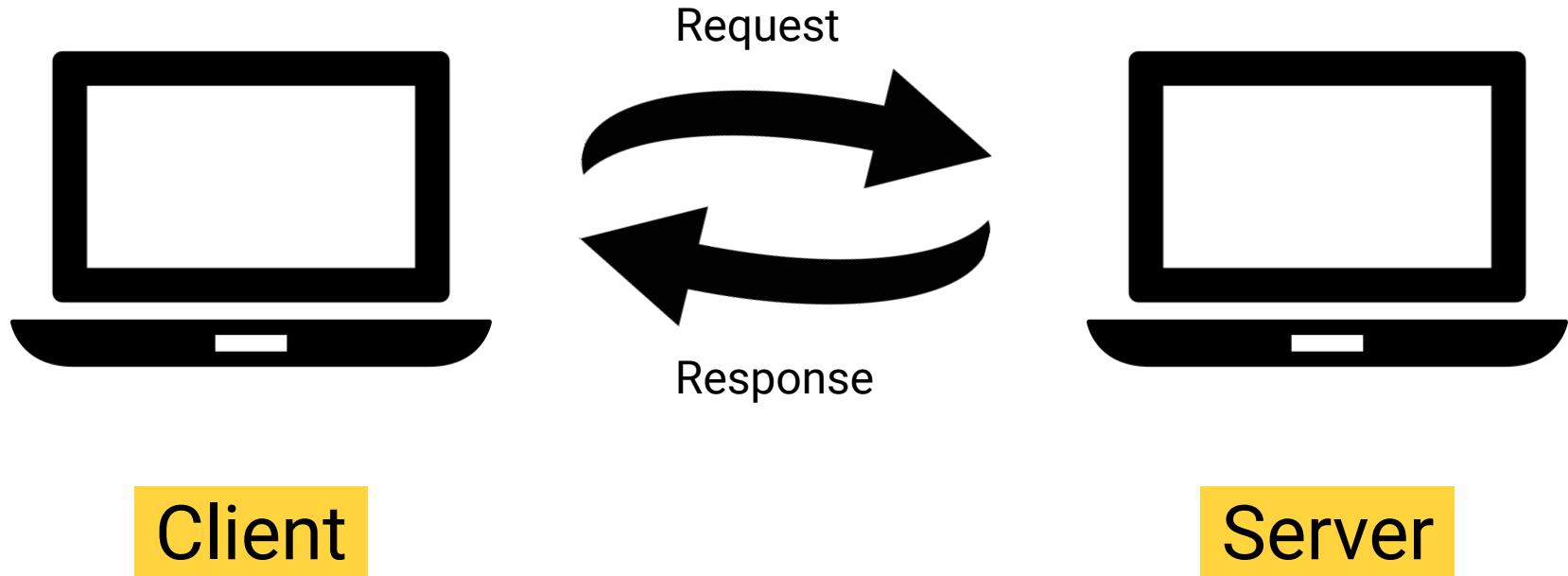


# Agenda

1. **Client-server model**
2. **HTTP protocol**

# 1. *Client-server* model

# Request-Response Cycle





Client

Makes the request and  
waits for the response

Examples:

- Browser
- Python code
- Postman HTTP client

Waits for requests, and responds to them

Examples:

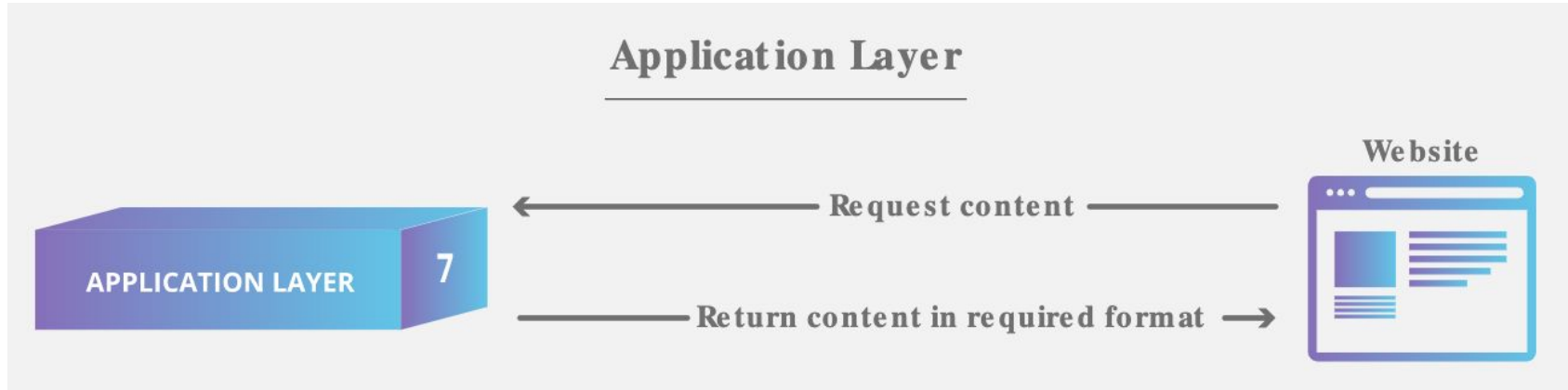
- Jupyter Notebook
- PostgreSQL
- Web server



Server

## 2. HTTP

# HyperText Transfer Protocol (HTTP)





# HTTP Request

| Component            | Example                                | Description  |
|----------------------|--|--|
| Method<br>(AKA Verb) | 'GET'                                  | What type of request this is. Conventionally <code>GET</code> means you want to retrieve information, <code>POST</code> means you want to submit new information |
| Path<br>(AKA URL)    | 'https://pokeapi.co/api/v2/pokemon/1/' | The address of the server, followed by the specific route, sometimes followed by params  |
| Headers              | '"Content-Type": "application/json"'   | Gives the server more information about how to interpret the request. Sometimes this includes authentication.  |
| Body                 | '{"trainer_id": 1}'                    | Used with a <code>POST</code> to submit data   |

# Web Server Interface

A web server is:

- Running constantly waiting for requests
- Able to accept requests and serve responses with an HTTP protocol
  - The client and server can “speak” totally different languages, so long as they both know how to use this protocol
- Usually configured for human users or code clients, but not both
  - For human users, return HTML/CSS/JS that will render something in a browser
  - For code, return JSON to be used by another program