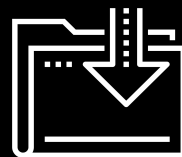




APIs

FinTech  
Lesson 5.1



# Class Objectives

---

By the end of today's class, you will be able to:



Describe the client-server model.



Read documentation and identify endpoints from a given API.



Perform a GET request and view the JSON response using the Postman client.



Interpret the JSON structure from a GET request response.



Use the requests library to request JSON data from an API within Python.



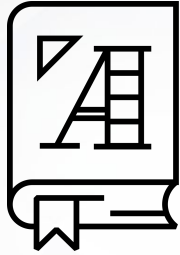
Parse a JSON response and print a selected field using Python.



Query an API using URL Parameters with the requests library.

**Disclaimer:** The response and content of live data cannot be censored or predicted.

# Introduction to APIs

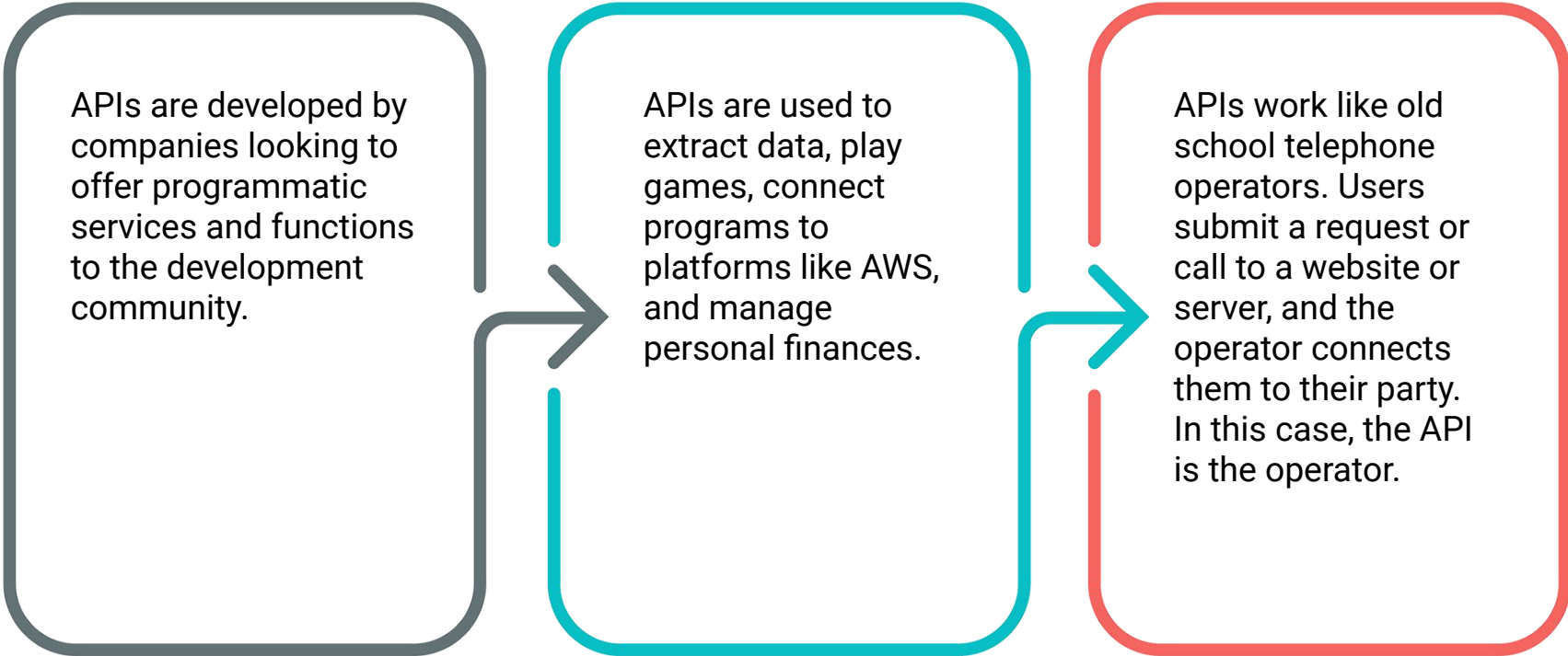


## **Application Programming Interfaces (APIs)**

are a set of functions packaged together that provide developers with a means of communicating with a server and integrating third-party software and technology into new applications.

# Application Programming Interfaces (APIs)

---



APIs are developed by companies looking to offer programmatic services and functions to the development community.

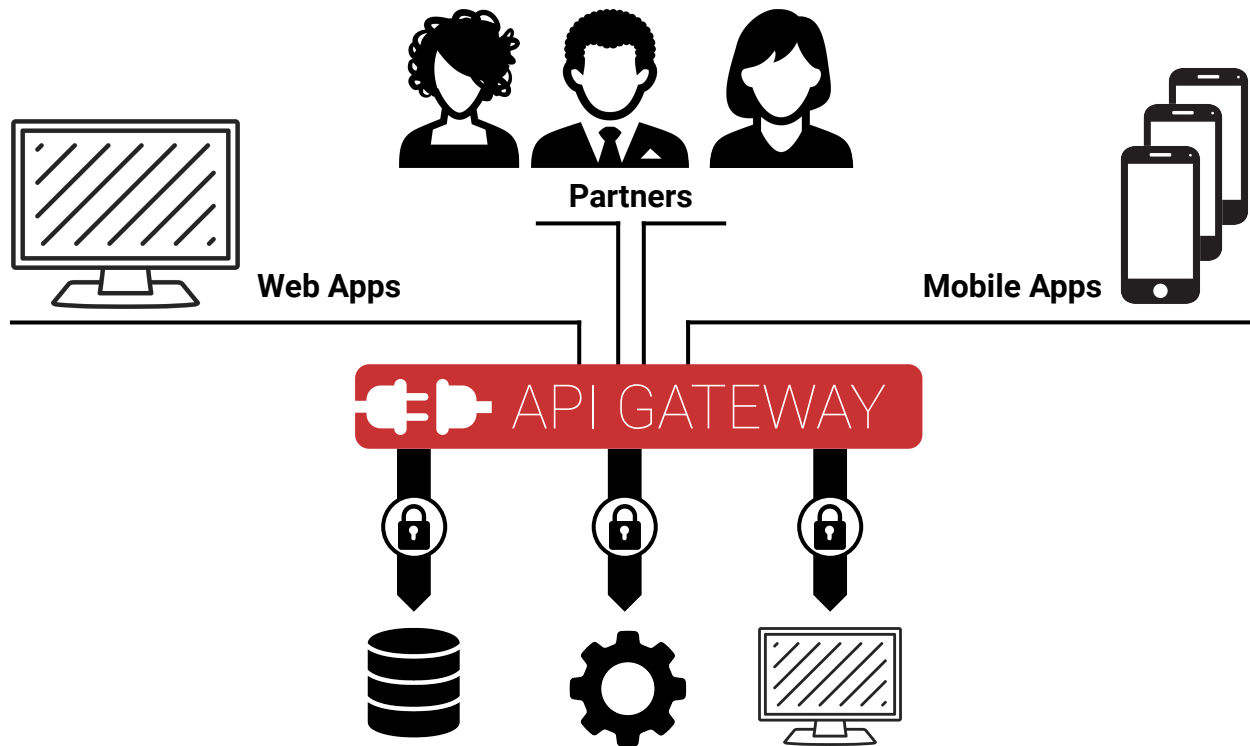
APIs are used to extract data, play games, connect programs to platforms like AWS, and manage personal finances.

APIs work like old school telephone operators. Users submit a request or call to a website or server, and the operator connects them to their party. In this case, the API is the operator.

# API Recap

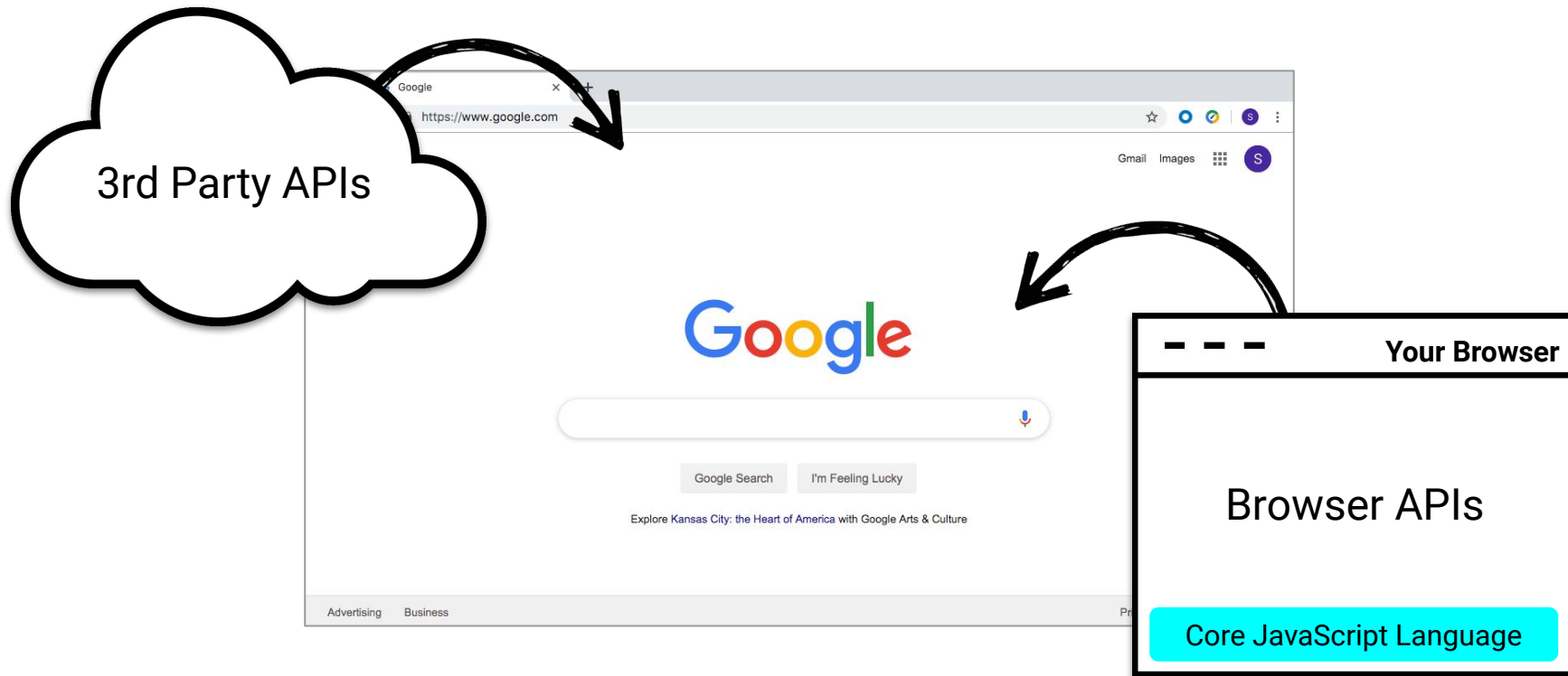
---

In software development APIs are often the bridge between different components



# API Recap

In software development, APIs are often the bridge between different components.





# Nasdaq Data Link

Nasdaq Data Link is a data mart of financial data. Quandl collects data from various sources, consolidates the data, and then makes it available to users.

Nasdaq Data Link is a great product to use to extract financial data to calculate ROI, risk-to-reward ratio, etc. The data can be accessed by users via their API. The API supports multiple programming languages, including Python.



Nasdaq Data Link A premier source for financial, economic and alternative datasets.



# Instructor Demonstration

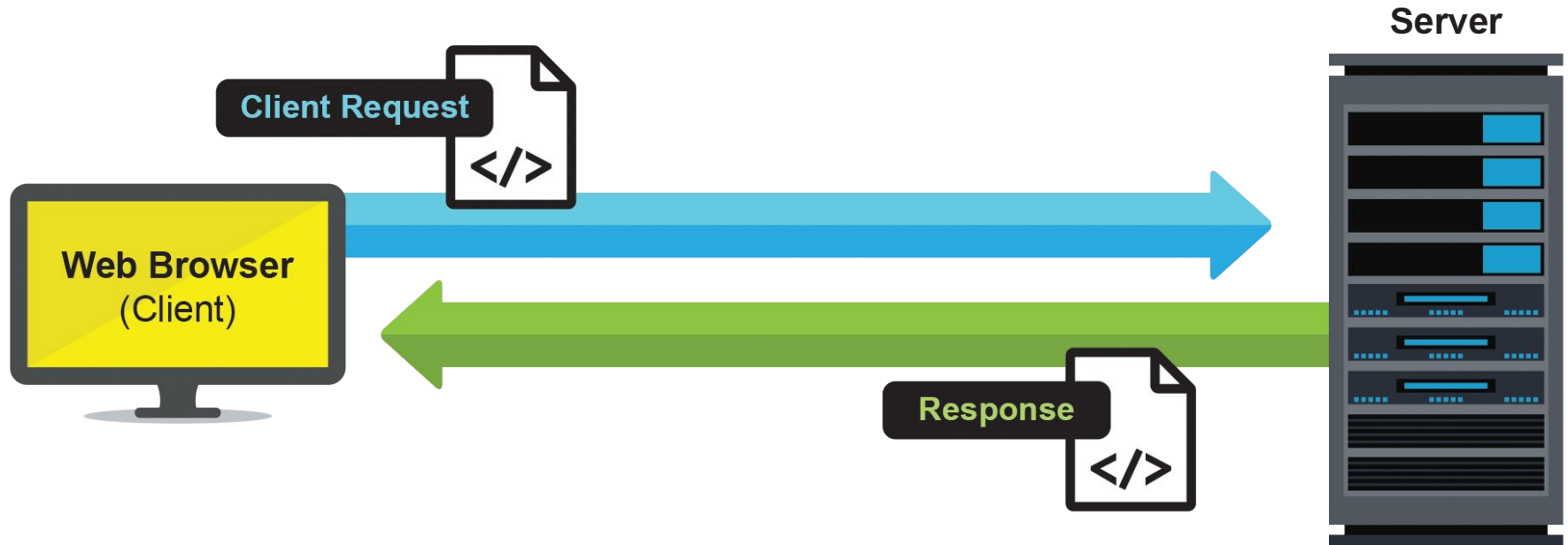
## Review Homework

# Client-Server Model

# Client-Server Model

---

The Client-Server Model is a structure that outlines the relationship and flow of communication between two components: a client and a server.



# The Client-Server Model

---

A client is any tool or application that is used to connect to or communicate with a server. Example **clients** include:



Web browsers

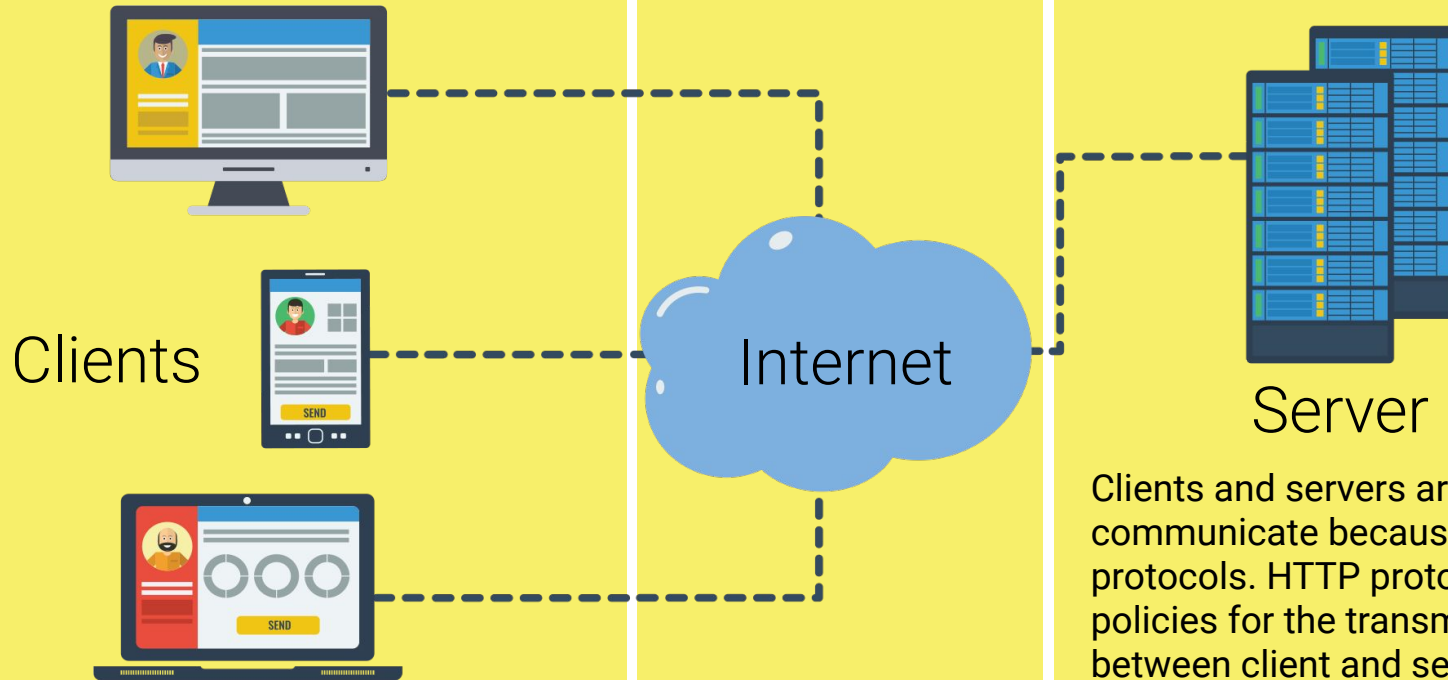


Mobile devices



Command-line interfaces

# The Client-Server Model



Clients submit requests to servers and then wait for a response from the server.

Clients and servers are able to communicate because of HTTP protocols. HTTP protocols are policies for the transmission of data between client and server. HTTP protocols allow users to execute server functions and clients and servers to exchange data.



## Activity: Eavesdropping on the Server

In this activity, you will surf the web with the browser's developer console open, visit websites like Facebook and Yahoo, and complete a Google search. You will read the standard output from the console to get a better idea of what data is sent between client and server as you navigate sites.

(Instructions sent via Slack.)

**Suggested Time:**  
10 Minutes

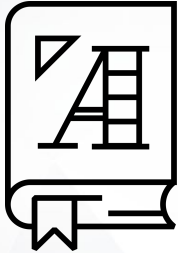




**Time's Up!** Let's Review.



# Postman



**Postman** is a service that provides users with a UI to submit and store API calls and requests.



# POSTMAN



API requests have to be submitted in some type of development environment. Postman offers an API specific development environment that is free to users.



Postman is a great tool to use when onboarding onto a new API. All that is needed to execute an API with Postman is the request URL.



Because Postman is a development environment, users have the ability to save API requests, configure environments, and even create mock servers.



## Activity: I Spy an API

In this activity, you will go through a list of FinTech APIs and test out their functionality using Postman. This will give you a better understanding of what Postman is and how it should be used, and it will expose you to one of the most common tools used in the FinTech industry.

(Instructions sent via Slack.)

**Suggested Time:**  
15 Minutes





## Activity: Parlez vous le JSON

In this activity, you will choose a sub-selection of the JSON output to decipher. You will then explain the sub-selection to a peer.

(Instructions sent via Slack.)

**Suggested Time:**  
5 Minutes





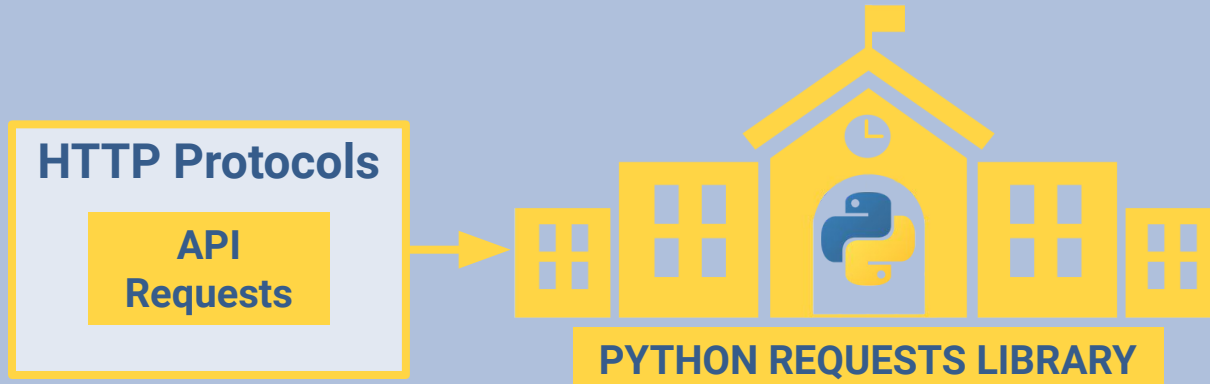
**Time's Up!** Let's Review.

# Python Requests

# Python Requests

---

Python has a requests library that can be used to make API calls. The requests library works similar to Postman. It allows for developers to submit API requests using HTTP protocols.



The requests library allows developers to use Python like glue, connecting their Python code with multiple third party APIs. This allows developers to create programs that are an amalgamation of multiple technologies!



# Python Requests

---

The requests library has its own functions, such as **GET** and **POST**. These can be used to execute API calls programmatically with Python.

The difference between Python's request library and Postman is that **developers can use Python objects (variables, data structures) to make APIs interact with one another** when they normally wouldn't. This allows developers to:

- Pass the output of one API as input to another API
- Utilize conditionals
- Leverage loops

# Python Requests

---

Each type of request serves a different purpose.

GET

**GET** requests are used to extract/acquire data from a server

POST

**POST** requests are used to push new or updated data to the server

PUT

**PUT** requests are used to overwrite content on the server



# Instructor Demonstration

## Python Requests



## Activity: Ice Breakers On Request

In this activity, you will be given a list of request URLs to execute using the Python requests library. You will also receive the opportunity to put your JSON knowledge to use by interpreting JSON output.

(Instructions sent via Slack.)

**Suggested Time:**  
20 Minutes





## Activity: Engagement

In this activity, you will reveal an interesting fact or joke you discovered while working with APIs.

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.





Countdown timer

**15:00**

(with alarm)

# URL Parameters



# URL Parameters

---

URL parameters serve as a means of configuring and changing API functionality.

**Parameters can be specified in one of two ways.** Parameters can follow / forward slashes or be specified by parameter name and then by parameter value.

Parameter provided after /

```
http://numbersapi.com/42
```

Parameter provided using  
parameter name and value

```
http://numbersapi.com/random?min=10?json
```

# URL Parameters

---

URL



```
graph TD; A[URL] --> B[http://www.site.com/page.html?parameter1=[@field:fieldname1]&parameter2=[@field:fieldname2]]
```

`http://www.site.com/page.html?parameter1=[@field:fieldname1]&parameter2=[@field:fieldname2]`

# URL Parameters

---

URL



```
graph TD; A[URL] --> B[http://www.site.com/page.html?parameter1=[@field:fieldname1]&parameter2=[@field:fieldname2]]
```

`http://www.site.com/page.html?parameter1=[@field:fieldname1]&parameter2=[@field:fieldname2]`

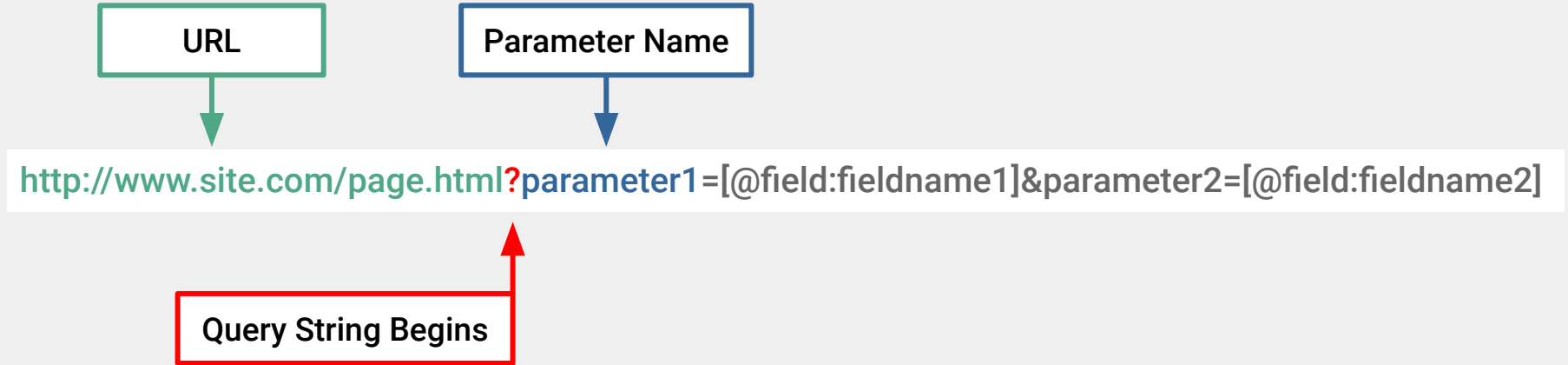
Query String Begins



```
graph BT; C[Query String Begins] --> D[http://www.site.com/page.html?parameter1=[@field:fieldname1]&parameter2=[@field:fieldname2]]
```

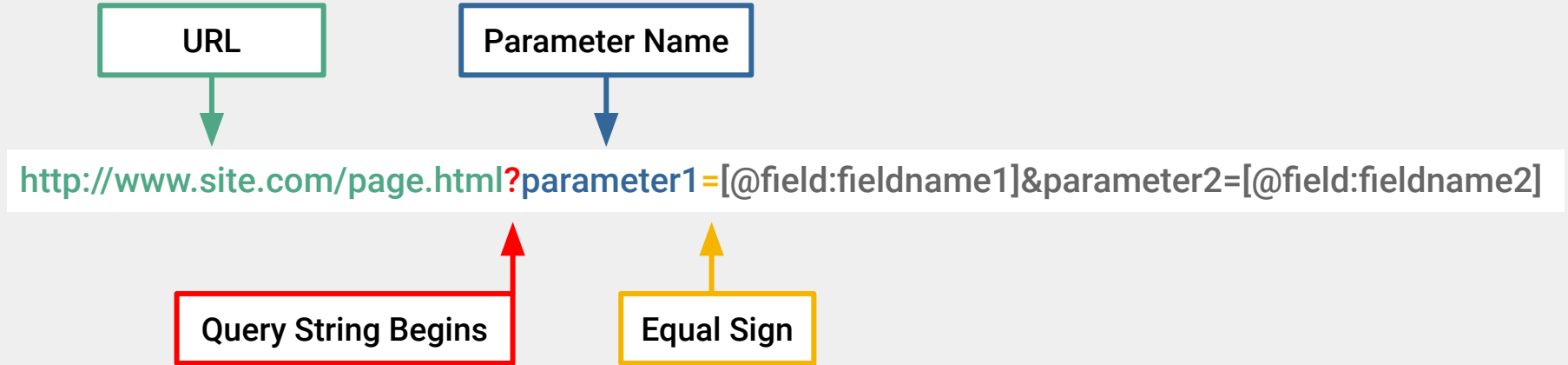
# URL Parameters

---



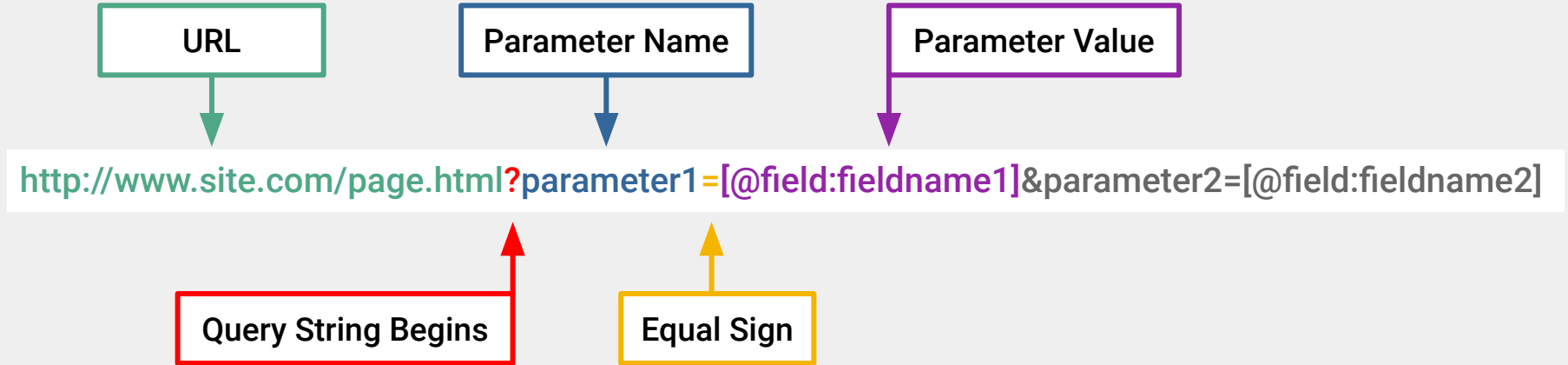
# URL Parameters

---



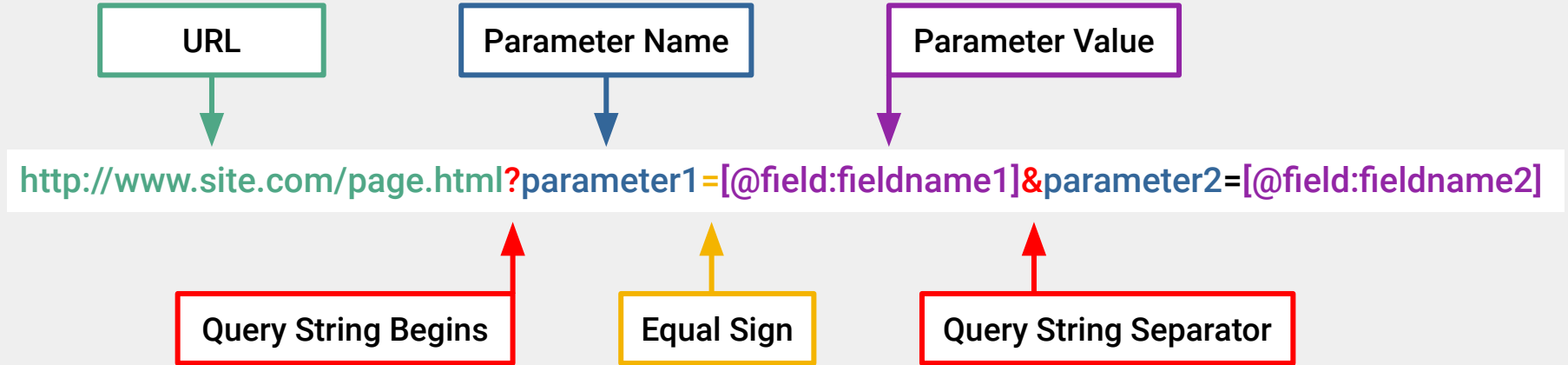
# URL Parameters

---



# URL Parameters

---





# Instructor Demonstration

## URL Parameters





## Activity: House of Requests

In this activity, you will play a game of Blackjack using the Deck of Cards API. You can play the game against a classmate or imaginary dealer.

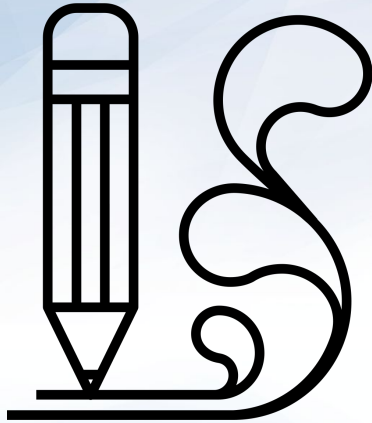
(Instructions sent via Slack.)

**Suggested Time:**  
25 Minutes





**Time's Up!** Let's Review.



## Homework: Sign Up for APIs

Sign up for these APIs before coming to the next class; you will use them in Lesson 5.2.

(Links sent via Slack.)

**Due Date:**  
**Next Class**



# Questions

*The  
End*