******API - Application Programe Interface******

An **API** lets two pieces of software talk to each other.
Just like a function, you don't have to know how the API works only its inputs and outputs.

Examples:

1. Establish communication between your program and other software component using API via inputs and outputs.

Using Pandas as an example to explain communication between your python program and other software components.
When you create a Pandas object with the Dataframe constructor in API lingo, this is an "instance".
The data in the dictionary is passed along to the pandas API.
You then use the dataframe to communicate with the API.

2. REST API
It allows you to communicate through the internet to take advantage of resources like cloud etc
REpresentational State Transfer - REST.
With REST API you can communicate your program (client) with Web services.
Input/request and Output/response
Web service is considered resource

3. REST API over HTTP
Transfering data over the internet.
a. The request is communicated via HTTP message where data is in the form of JSON file.
It is sent as request via API to Resource(Web service).

b. Sevice performs the operation.

c. API then returns a response via an HTTP message and the resultant data is the transfered in JSON format back to client(you).

******HTTP Protocol******

When the client uses a web page the browser sends an HTTP request to the server where page is hosted.
The server tries to find the desired resource by default "index.html".
If your request is successful, the server will send the object to the client in an HTTP response.
This includes information like the type of the resource, the length of the resource, and other information.

HTTP request is sent to server and response is returned by resource in the form of html,png file and txt file.

The HTTP protocol allows you to send and receive information through the web including webpages, images, and other web resources.

URL - Uniform Resource Locator
URL consists of 3 parts:
1. **Scheme -** The protocol eg. http://
2. **Base URL/Internet address -** eg.www.gitlab.com
3. **Route -** location on web server eg./image/logo.png

URL are actually a subset of **(Uniform Resource Identifier)** URIs.
Endpoint-is the URL of an operation provided by a Web server.
(http://www.gitlab.com/image/logo.png)

Request process - includes request start line, request Header and request body.

Request start line:
GET method-HTTP method The location of the resource-index.html the HTTP version

Response process - includes response start line, response Header and response body.

Response start line:
Response start line contains the version number HTTP/1.0, a status code (200) meaning success, followed by a descriptive phrase (OK)

Response header: useful information.

Response body: contains the requested file, an HTML document.

STATUS CODES:
1XX - Informational
2XX - Success
200 - OK
3XX - Redirection
300 - Multiple choices
4XX - Client Error
401 - Unauthorized
403 - Forbiden
404 - Not found
500 - Internal server error.

HTTP Methods:
GET - Retrive data from server.
POST - Submit data to server.
PUT - Update data already on server.
DELETE - Delete data from server.

********HTTP using request library in python********

Request module in python:
Check jupyter notebook.

Get Request with URL Parameters:
Use the GET method to modify the results of your query.
For retrieving data from an API, send a GET request to the server.

A **query string** is a part of a uniform resource locator (URL),
This sends other information to the web server.
The start of the query is a ?, followed by a series of parameter and value pairs.

Start of Query	Parameter Name		Value		Parameter Name		Value
?	name	=	Joseph	&	ID	=	123
http://httpbin.org/get? Name=Joseph&ID=123							

Post Request:
A POST is used to send data to a server.
The POST request sends the data in a request body.
Endpoint will expect data as a file or as a form.
A form is convenient way to configure an HTTP request to send data to a server.