**Hacer una llamada a la API en Python**

**1. Primero usamos el módulo de solicitudes para enviar solicitudes HTTP (es decir, solicitar datos a través de Internet) usando Python.**

**2• Solicitudes: documentación de HTTP para humanos**

**3• Documentación de Python en el módulo de solicitudes**

**4• Módulo de Solicitudes de Python, Escuelas W3**

**5. La solicitud HTTP devuelve un objeto de respuesta que incluye los datos que se devuelven como parte de la llamada a la API.**

**6. El módulo de solicitudes incluye una gama de métodos que nos permiten interactuar con elementos de la API.**

| **Method** | **Description** |
| --- | --- |
| delete(url, args) | Sends as DELETE request to the specified URL |
| get(url, params, args) | Sends a GET request to the specified URL |
| head(url, args) | Sends a HEAD request to the specified URL |
| patch(url, data, args) | Sends a PATCH request to the specified URL |
| post(url, data, json, args) | Sends a POST request to the specified URL |
| put(url, data, args) | Sends a PUT request to the specified URL |
| request(method, url, args) | Sends a request of the specified method to the specified URL |

7. Un ejemplo muy básico del módulo de solicitudes en acción.

*# import requests module*

**import** requests

*# store URL*

x **=** requests**.**get('url')

*# print URL text*

print(x**.**text)

In [ ]:

*# importer libreerias*

**import** urllib

**import** urllib.request

*# asignar url a variable*

url **=** 'https://en.wikisource.org/wiki/John\_F.\_Kennedy%27s\_Third\_State\_of\_the\_Union\_Address'

*# open url*

response **=** urllib**.**request**.**urlopen(url)

*# read response*

webContent **=** response**.**read()

*#crear Nuevo archivo HTML*

f **=** open('Kennedy\_Third\_SOTU.html', 'wb')

*# assign response to html file*

f**.**write(webContent)

*# cerrar archivo*

f**.**close

**GitHub API in Python.**

In [ ]:

*# import requests module*

**import** requests

*# import json module*

**import** json

*# store API url*

url **=** 'https://api.github.com/search/repositories?q=language:python&sort=stars'

*# assign the headers- not always necessary, but something we have to do with the GitHub API*

headers **=** {'Accept': 'application/vnd.github.v3+json'}

*# assign the requests method*

r **=** requests**.**get(url, headers**=**headers)

*# print a status update for the requests command*

print(f"Status code: {r**.**status\_code}")

*# store API response to variable*

response\_dict **=** r**.**json()

*# process results*

print(response\_dict**.**keys())

8.La primera instrucción print() devuelve un mensaje de estado para la llamada a la API request.get().

9. Una vez que sabemos que la llamada API está funcionando, podemos continuar y solicitar los datos JSON disponibles a través de la API.

10. Luego podemos almacenar esos datos JSON como un diccionario de Python.

11: Describa cómo funciona una API web con sus propias palabras.