**Task 4: Python y Fetch()**

Para poder disponer de los datos requeridos directamente del servidor, debemos crear un servidor local en nuestra pc debido a que AJAX solo se comunica entre servidores. Para esto, podemos utilizar Python. Dado que en Windows no viene preinstalado, tenemos que obtenerlo de <http://python.org> y descargar la versión ejecutable. Al instalarlo **es importante marcar la casilla PATH** para que funcione correctamente.

A continuación, para crear nuestro servidor local, nos dirigimos al **Windows PowerShell**, al cual se puede acceder haciendo click derecho en el ícono de inicio de Windows. Dentro de esta aplicación, buscamos el directorio donde se encuentra nuestro trabajo escribiendo **cd** seguido de la dirección de nuestra carpeta:

**cd “C:\Users\rodri\Google Drive\MindHub\2. Introduction to JavaScript Development\Task 4”**

Es importante encerrar entre **“comillas”** la dirección si alguno de nuestros directorios contiene espacios en blanco.

Para iniciar el servidor, a continuación debemos escribir la propiedad:

**Python -m http.server**

A partir de ahora podemos acceder a nuestra página tipeando <http://localhost:8000> en nuestro browser.

**Fetch ()**

Utilizamos fetch para extraer los datos directamente de propublica, en vez de descargarlo a nuestro equipo. Para esto debemos asignarle una url de donde extrarlos y a continuación un objeto que describa el método a utilizar (en este caso GET, aunque no es necesario explicitarlo debido a que este método es el que se realiza por defecto) y la API key que habíamos obtenido en la task 2 de propublica. Si la comunicación con el servidor es exitosa, la función .then() se activará; sino, la función catch entrará en juego. Debido a que fetch nos devuelve una promesa a realizar cuando el documento esté cargado, debemos llamar a las funciones que queremos realizar una vez que tenemos los datos. En mi caso, solo una vez que la variable data está cargada con los datos del servidor, se disparan las funciones table() para crear la tabla en el HTML, dropDownMenu() para el menu con todos los estados y mainFilter() para los filtros, que se encuentran escritas en el resto del código, pero que solo se activan una vez que el fetch extrae los datos del servidor y los carga en la variable data.

var data;

$(function() {

fetch("https://api.propublica.org/congress/v1/113/senate/members.json",{

method: 'GET',

headers: new Headers({

"X-API-Key": "1pS4P2QU0p0J3PlmSeM9PmZeMG4tj1Z5DdQKktG4"

})

} ).then(function(response){

return response.json();

}).then (function(json){

data = json;

table();

dropDownMenu();

mainFilter();

}).catch(function(){

if (data == undefined){

console.log("Fail")

} else {

table();

dropDownMenu();

mainFilter();

}

})

});