



JS: Ejercicio 2

Descripción:

Nota: Este es un ejercicio sobre el que se basará el proyecto final.

Realizaremos un pequeño proyecto de actualización sin cambiar de página (AJAX), consultando a una API pública comercial con información meteorológica (OpenWeather).

En primer lugar, y de manera similar a la que aprendimos en el módulo de Core Javascript, haremos una llamada a una API externa. Como la llamada tiene un retardo, podemos usar una estrategia basada en promises y escuchar la respuesta.

Después, con la información recibida y con los conocimientos de JS y las [APIs HTML5](#) gestionaremos la respuesta para manipular el DOM creado con nuestro documento HTML.

Pasos previos (No evaluables)

1. Crear una cuenta en [OpenWeather](#)
 - Este paso no se evalúa y es posible compartir una misma cuenta entre alumnos.
 - Las llamadas a la api necesitan un identificador de cuenta llamado AppId. Un ejemplo de AppId sería : 2123b15abf5dbccb4b78d19ccea8dd7d
 - En el momento de escribir esto hay un límite de 1000 llamadas diarias por cada cuenta.
 - Es recomendable testear la llamada con cURL, postman u otra herramienta a gusto del alumno

Pasos **HTML**:

2. Crea una página web que contenga un "formulario" con dos elementos tal y como ha aprendido el alumno.
 - a. El primer elemento será un campo de texto(textfield) donde el usuario introduzca una localidad.
 - b. El segundo elemento será un botón que el usuario podrá clicar.

- c. Esta vez, no usaremos formulario `<form>` que nos envíe los datos recogidos a una URL, sino que utilizaremos eventos Javascript. Es decir, esperaremos a que el usuario pulse un botón para ejecutar un código determinado.
3. Junto al formulario, debemos mostrar los siguientes datos para que quede una frase tipo a *En **Tajonar**, **Es** la previsión meteorológica es de “**nubes dispersas**”. La temperatura es de **29°C** , con una humedad relativa del **65%***
4. Indudablemente deberemos usar un documento javascript al que haremos referencia con `<script src=“...”></script>`

Para esto será necesario que dejemos espacios en blanco identificados que después podamos rellenar según extraigamos datos de la API

```
<p> En <span id="localidad"> ... </span> la previsión meteorológica es de "<span id="previsionTexto"> ... </span> "
```

Pasos **Javascript**

5. Debemos recibir el feedback del usuario cuando pulse el botón. Para ello debemos
 - a. Identificar al botón
 - b. Asignarle un event_listener para el evento “click”
6. Una vez que se dispara el evento debemos gestionarlo para que realice las acciones pedidas:
 - a. Realizar la llamada (fetch)
 - b. Recibir asíncronamente la respuesta a la llamada
 - c. Identificar dentro de la respuesta los campos que necesitamos:
 - i. Localidad para la que estamos haciendo la petición
 - ii. La previsión meteorológica en texto. “Chubascos leves”, “Parcialmente nublado”, “Soleado”, ...
 - iii. Temperatura en °C
 - iv. Humedad relativa en %
 - d. Llevar estos elementos a los correspondientes elementos que hemos creado en el documento HTML

Notas:

- Los resultados de cada alumno serán diferentes dependiendo de cómo se apliquen los estilos. Los entregables podrán ser publicados y visibles por el resto de

alumnos. Podrán ser usados en clase para compartir experiencias y mejorar el conocimiento colectivo.

- Sí se valoran los comentarios sobre el código.
 - No está permitida cualquier práctica peligrosa con javascript. Esto incluye , entre otras, la lectura de cookies/localstorage/sessionstorage o similares, comunicación externa o cualquier otra práctica que impersona al usuario o ponga en riesgo su privacidad.
-
- Ejemplo de código devuelto por la API:

```
{
  "coord": {
    "lon": 0,
    "lat": 0
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "nubes",
      "icon": "04n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 299.59,
    "feels_like": 299.59,
    "temp_min": 299.59,
    "temp_max": 299.59,
    "pressure": 1012,
    "humidity": 78,
    "sea_level": 1012,
    "grnd_level": 1012
  },
  "visibility": 10000,
  "wind": {
    "speed": 5.85,
    "deg": 177,
    "gust": 5.31
  },
}
```

```
"clouds": {  
  "all": 100  
},  
"dt": 1673460850,  
"sys": {  
  "sunrise": 1673417038,  
  "sunset": 1673460667  
},  
"timezone": 0,  
"id": 6295630,  
"name": "Globe",  
"cod": 200  
}
```

Entrega:

Subir los diferentes archivos en la plataforma Moodle.