



JS: Proyecto final

Descripción:

Nota: Este es un ejercicio está basado en el ejercicio 2

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](#)
 - a. Este paso no se evalúa y , es posible compartir una misma cuenta entre alumnos. El profesor entregará unas credenciales de prueba.
Hay un límite de 1000 llamadas diarias por cada cuenta.
 - b. Es recomendable testear la llamada con cURL, postman u otra herramienta a gusto del alumno

Pasos HTML:

2. Debemos mostrar los siguientes datos para que quede una frase tipo a *La previsión meteorológica en su ubicación (**Latitud, longitud**) es de "**nubes dispersas**". La temperatura es de **29°C** , con una humedad relativa del **65%***
3. 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> La previsión meteorológica en su ubicación `

... `` es de `"` ... `` "

Pasos Javascript

4. En vez de recibir el feedback del usuario cuando pulse el botón, usaremos las API Web para consultar la ubicación del usuario (latitud y longitud)
 - a. Esta API, que vimos en clase se puede consultar en el recurso de preferencia del alumno
(https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API/Using_the_Geolocation_API,
https://www.w3schools.com/html/html5_geolocation.asp, ...)
5. Debemos comprobar que hemos obtenido unas coordenadas. Si la API de geolocalización devolviese un error deberemos avisar al usuario, por ejemplo, con un `alert()` .

En principio si las coordenadas que nos devuelve son (0,0) no lo consideramos un error. Simplemente será que nuestro navegador no sabe o no nos quiere dar la ubicación.
6. En vez de esperar a que se dispare el evento (ejercicio 2) esta vez, directamente usamos las coordenadas obtenidas para llegar al objetivo.
 - a. Realizar la llamada (`fetch`)

Esta vez la llamada no usará el clave "q" para la ubicación, sino que usará "lat" y "long"
 - b. Recibir asíncronamente la respuesta a la llamada
 - c. Identificar dentro de la respuesta los campos que necesitamos:
 - i. La localidad registrada bajo esas coordenadas (si no existe ninguna localidad la api nos devolvera el nombre del pais, del continente o la palabra "globe")

Vendrá bajo la clave `respuesta.name`
 - ii. Latitud y longitud para la que estamos haciendo la petición
 - iii. La previsión meteorológica en texto. "Chubascos leves", "Parcialmente nublado", "Soleado", ...
 - iv. Temperatura en °C
 - v. Humedad relativa en %
 - d. Llevar estos elementos a los correspondientes elementos que hemos creado en el documento HTML

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
}
```

```
}
```

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.

Entrega:

En la plataforma Moodle.

Deberá entregarse dentro de un fichero comprimido ZIP/RAR