|  |  |  |  |
| --- | --- | --- | --- |
| Sigla Asignatura | DWY4101 | Nombre de la Asignatura | DESARROLLO DE APLICACIONES WEB Y MOBILE |
| Nombre del Recurso Didáctico | **1.1.13 Actividad - ¿Cómo está el tiempo?** | | |
| Unidad de Aprendizaje N° 1 | Front End de una Aplicación Web | | |
| Unidades de Competencia | Desarrolla aplicaciones utilizando patrones de diseño y buenas prácticas de programación que soporten los requerimientos de la organización (3) | | |

**Objetivos**:

Construir una página web que consuma información desde un API, utilizando AJAX y jQuery para mostrar el contenido en la pantalla.

**Instrucciones**:

Los estudiantes realizarán una página que obtenga desde un sitio externo datos meteorológicos de ciudades de Chile y mostrarán el estado actual del tiempo para una ciudad.

**Material Complementario:**

Sin material complementario

**Forma de Trabajo:**

Individual

**Tiempo estimado:**

1,5 horas

**Descripción de la actividad:**

|  |
| --- |
| Los estudiantes realizarán una página que obtenga desde un sitio externo datos meteorológicos de ciudades de Chile y mostrarán el estado actual del tiempo para una ciudad.  **INSTRUCCIONES** PASO 1: La estructura Construye una estructura de directorios adecuada para soportar el proyecto, puedes utilizar la que se muestra a continuación como ejemplo. PASO 2: Los datos Para obtener los datos meteorológicos, utilizaremos el API que ofrece gratuitamente Wunderground, pero puedes utilizar cualquier otro que conozcas.  Un API, es un set de funciones y procedimientos escritas por un tercero con el fin de ser utilizadas en un software. En este caso, utilizaremos el API del Clima que ofrece Wunderground.  Para acceder a la información meteorológica de una ciudad, prueba ingresando a la siguiente URL a través de tu navegador:  <http://api.wunderground.com/api/e069d13c9432ad2f/conditions/q/CL/Santiago.json>  Reemplazando el nombre de la ciudad por la que desees consultar. Deberías ver algo como esto:    El API entrega la información en JSON, un formato de texto ligero para simplificar el intercambio de datos entre sistemas. PASO 3: El JavaScript En el archivo script.js, construiremos una función que se encargue de leer los datos JSON entregados por el API y escribir la información que seleccionemos en elementos HTML.  La función tiene tres partes:   1. La lectura de JSON 2. La captura de los datos necesarios 3. La impresión de los datos capturados en un HTML  PASO 3.1: Lectura de JSON Utilizando un objeto XMLHttpRequest, capturaremos los datos entregados por el archivo JSON.  Si observas el archivo entregado por el API, notarás que los datos están estructurados en bloques agrupados por paréntesis de llave, {}. Cada uno de esos bloques son Objetos JSON, que se componen de una serie de valores de diferentes tipos identificados por un nombre, arreglos y otros objetos JSON.  Por ejemplo, el objeto JSON **current\_observation**, contiene en su interior un objeto llamado **display\_location** que contiene la información de la zona geográfica que se está observando:  display\_location": {  "full":"Santiago, Chile",  "city":"Santiago",  "state":"RM",  "state\_name":"Chile",  }  El nombre del país es accedido a través del elemento **state\_name**, tal como se accede a un valor en una variable o un atributo en los otros lenguajes de programación que conoces.  Volviendo a nuestro código, luego de leer el archivo JSON, JavaScript carga los datos en la variable **datos.** Desde este punto en adelante, el objeto **datos** contiene toda la información disponible en el archivo JSON.   PASO 3.2: La captura de los datos necesarios Ahora que tenemos volcado en un elemento JS la información del API, podemos capturar los datos que necesitemos respetando la estructura original del archivo JSON.  Por ejemplo, podremos capturar la ciudad observada en una variable JS de la siguiente forma:  var ciudad = datos.current\_observation.display\_location.full  Capturemos algunos valores para mostrar en nuestra página. En este ejemplo, trabajaremos con la ciudad, temperatura, humedad relativa y la condición del tiempo actual (soleado, lluvioso, despejado, etc.) pero tú puedes experimentar con todas las demás opciones que entrega el API.   PASO 3.3: La impresión de los datos capturados en un HTML Ahora solo queda escribir, en elementos HTML, los valores capturados en las variables JS. Crearemos un documento HTML para ello y volveremos a este punto más adelante.  **PASO 4: El HTML**  Construyamos una estructura HTML sencilla que permita mostrar el contenido de las variables JavaScript creadas anteriormente. En este caso, se utilizando elementos img, p, span y encabezados para ayudar a la jerarquía visual de los elementos. Más allá de los elementos que elijas para tu estructura, es importante que éstos tengan asociado un ID para poder manipularlos con jQuery en el último paso. ¡Adelante! Construye tu propio archivo HTML o utiliza el que proponemos.    Ahora, volvamos al paso 3.3… PASO 3.3 La impresión de los datos capturados en un HTML (Segunda parte) Otra vez volvamos a nuestra función JS. Utilizaremos jQuery para acceder a los elementos HTML construidos en el paso anterior mediante el ID que asignamos a cada uno e inyectaremos el contenido de las variables JavaScript en los elementos HTML.   PASO 5: Últimos detalles Para terminar, debemos asegurarnos de que el archivo HTML se encuentre enlazado con el archivo de script donde construimos la función.  Además, no podemos olvidar incluir la librería de jQuery en el encabezado del documento HTML.  Por último, debemos invocar la función que escribimos en la función ready.    ¡Listo! Revisa el resultado en tu navegador y corrige posibles errores con ayuda del inspector y la consola del navegador. |