Web Components, el futuro del desarrollo web

El mundo del diseño web avanza a una gran velocidad. Cada vez son mayores las demandas por parte de los usuarios, que buscan nuevas características para darle más funcionalidades a sus páginas web. El estándar HTML 5 ha supuesto una revolución a la hora de llevar a cabo un proyecto gracias a la llegada de nuevas etiquetas. Pero esto no ha quedado ahí, ya que gracias a los Web Components es posible personalizarlas. A lo largo de este White Paper hablaremos más sobre esta nueva tecnología que permite crear etiquetas HTML.

¿Qué son los Web Components?

La especificación de los Web Components empezó a desarrollarse hace aproximadamente un par de años, aunque algunas de las tecnologías que hay que utilizar para crearlos nacieron mucho antes. Hablar de Web Components es hablar de una revolución para el campo del desarrollo web ya que ofrece un mecanismo para reutilizar y encapsular código HTML, CSS y JavaScript, que luego poder utilizar una y otra vez en cualquier proyecto.

<gangnam-style></gangnam-style>

De una forma más simplificada, se puede decir que se trata de nuevas etiquetas HTML personalizadas, creadas por los desarrolladores para sus proyectos. Cada una de estas etiquetas ejecutará un determinado código de programación y se diferenciarán del resto de etiquetas que ya conocemos, porque en su nombre siempre llevarán un guión. A continuación os dejamos lo que sería un ejemplo de este tipo de etiquetas, que se encargaría de pintar un mapa de Google en nuestra página web.

<google-map latitude="37.77493" longitude="-122.41942"></google-map>

Si nos fijamos, este Web Component contiene una serie de propiedades para indicar la latitud y la longitud que queremos representar en el mapa, propiedades que se definirían en el proceso de creación del Web Component.

Elementos que intervienen a la hora de crear un Web Component

A la hora de diseñar un Web Component, es necesario utilizar 4 elementos complementarios pero independientes entre sí. Veamos cuáles son estos elementos.

1.- Custom Elements o etiquetas personalizadas

El primero que vamos a explicar hace referencia a la creación de las nuevas etiquetas personalizadas para dar solución a cualquier problema, con la ventaja de que podrán ser reutilizadas siempre que queramos.

Pongamos un ejemplo para que se pueda ver con mayor claridad lo que acabamos de indicar. Sigamos con el caso de implementar un mapa de Google. Si nos vamos al tutorial de desarrolladores de Google Maps, en su parte final aparece que el código necesario para mostrar un mapa sería algo como lo que os dejamos a continuación.

```
<!DOCTYPE html>
<html>
 <head>
  <style>
   #map {
    width: 500px;
    height: 400px;
   }
  </style>
 </head>
 <body>
  <div id="map"></div>
  <script>
   function initMap() {
    var mapDiv = document.getElementById('map');
    var map = new google.maps.Map(mapDiv, {
     center: {lat: 44.540, lng: -78.546},
     zoom: 8
    });
   }
  </script>
  <script src="https://maps.googleapis.com/maps/api/js?callback=initMap"
    async defer></script>
 </body>
</html>
```

Ahora bien, gracias al desarrollo de los Web Components y la creación de las etiquetas personalizadas, todo ese código se podría simplificar de la siguiente forma:

```
<google-map latitude="37.77493" longitude="-122.41942"></google-map>
```

Es decir, una vez que lo tengamos creado, nos olvidaremos de toda la programación y sólo haremos uso del nuevo tag.

Se pueden crear todos los elementos personalizados que queramos, pero todos ellos deben cumplir un requisito, y no es otro que su nombre contenga un guión para evitar conflictos con las actuales etiquetas o las que puedan llegar en un futuro, formadas por una sola palabra.

2.- Templates

Las especificaciones de los Web Components incorporan un sistema de plantillas que contienen tanto código HTML como CSS y que inicialmente no se mostrará en la página. El objetivo de estas plantillas, es que mediante el uso de JavaScript, se acceda al código que hay dentro de ellas. La ventaja de utilizar un sistema de plantillas radica en que se podrá utilizar las veces que haga falta en otro lugar de la página.

Explicado de otra forma, una plantilla se puede entender como un esqueleto que utilizar para poder mostrar los datos que queramos al usuario mediante la inyección de la información que se haría por medio de JavaScript.

Un ejemplo de plantilla sería el siguiente código.

```
<template id="itTemplate">
  <dl>
    <template>
        <dt></dt><dd></dd>
    </template>
  </dl>
</template>
```

Mediante la ejecución del siguiente código JavaScript, podríamos rellenar esa plantilla con los datos que queramos mostrar. Por ejemplo, nombre y apellido de personas.

```
<script>
actores = [
      {'nombre' : 'Chris', 'apellido' : ' O\'Dowd '},
      {'nombre' : 'Richard', 'apellido' : ' Ayoade '},
      {'nombre' : 'Katherine', 'apellido' : ' Parkinson '}
   ];

 var t = document.querySelector('#itTemplate');
 var t2 = t.content.querySelector('template');
  var dt = t2.content.querySelector('dt');
  var dd = t2.content.querySelector('dd');
  var dl = t.content.querySelector('dl');
  actores.forEach(function(actor) {
    dtClone = dt.cloneNode(true);
    ddClone = dd.cloneNode(true);
    dtClone.textContent = actor.nombre;
    ddClone.textContent = actor.apellido;
    dl.appendChild(dtClone);
    dl.appendChild(ddClone);
  });

 var clone = document.importNode(t.content, true);
```

```
    document.body.appendChild(clone);

</script>
```

3.- Shadow DOM

Se trata de un sistema que permite tener una parte del DOM oculta a otros bloques de la página, lo que comúnmente se conoce como encapsulamiento, para que no pueda interferir con otras partes de la página. Gracias a esto, se soluciona uno de los mayores quebraderos de cabeza para los diseñadores, que tenían problemas a la hora de aplicar estilos en sus diseños ya que afectaban a otros elementos de la página, cogiendo una apariencia que no le correspondían o descolocando elementos que no deberían.

Gracias a esto, ahora se podrá colocar estilos que solo afectarán al Shadow DOM de un Web Component, evitando que estilos externos puedan afectar a su apariencia.

4.- HTML Imports

Por último tendríamos los HTML Imports, que nos permitirían importar un trozo de código para poder utilizarlo en un lugar de tu página. Ese código importado podrá contener código HTML, CSS y JavaScript. El uso de este código importado, podrá ser llevado a cabo mediante el uso de etiquetas “link”.

```
<body>
  <div class="header">
    <link rel="import" href="/ruta/al/header.html">
  </div>
</body>
```

Librerías para crear Web Components

Aunque hay librerías desarrolladas para facilitar la creación de Web Components, cabe decir que realmente no hacen falta, pero como suele ser habitual con este tipo de librerías, sí que ayudan en el proceso de desarrollo.

A pesar de esto, algunas empresas como Google o Mozilla se han apresurado a crear las suyas propias para ofrecer a los usuarios un mecanismo más sencillo para la creación de etiquetas personalizadas. A continuación os nombramos algunas de las más utilizadas.

Polymer
Posiblemente sea la librería más utilizada y conocida para la creación de este tipo de etiquetas. Se trata de un proyecto desarrollado por Google, al que le está dando un gran protagonismo lo que ha ayudado a adquirir parte de su fama. Está pensada para aprovechar la tecnología del estándar, facilitando la creación de etiquetas reutilizables.

X-Tag
Esta otra librería lleva el sello de Mozilla, que mantiene una disputada batalla con la anterior por ser la más utilizada. Ofrece resultados que son compatibles con todos los navegadores modernos que permiten el uso de los Web Components.

Bosonic
Esta última librería que os presentamos, ofrece un gran número de herramientas y utilidades para crear elementos personalizados, incluso para que sean compatibles para navegadores desfasados como podría ser el caso de Internet Explorer 9.

¿Cómo se utilizan los Web Components?

El uso de estos elementos en el diseño de una página web es muy sencillo una vez que lo tengamos desarrollado, tarea más tediosa ya que es necesario tener conocimientos de programación tanto de HTML como CSS y JavaScript. Por suerte, en la red nos podemos encontrar Web Components desarrollados por terceras personas que podemos utilizar en nuestro proyecto.

Para ello, lo primero que deberíamos hacer será indicar al proyecto que queremos utilizar ese tipo de etiquetas, cosa que haremos dentro del HEAD mediante el uso de la etiqueta "**link**". Por ejemplo, en el caso de querer utilizar el elemento para pintar un mapa de Google, lo haríamos de la siguiente manera.

```
<link rel="import" href="components/custom/google-map.html">
```

En el ejemplo anterior hemos supuesto que ese componente lo hemos almacenado en la ruta "components/custom" dentro de nuestro desarrollo, al igual que hemos supuesto que el nombre del archivo era "google-map.html".

Una vez que lo hemos incluido, ya podríamos utilizarlo en cualquier parte de nuestra página de la forma que hemos explicado anteriormente.

```
<google-map latitude="37.77493" longitude="-122.41942"></google-map>
```

Como hemos podido ver, la llegada de los Web Components abre un amplio abanico a los desarrolladores de páginas web, que podrán crearse su propia librería de etiquetas que luego utilizar en cualquiera de sus proyectos.