



POLYMER

© 2017, ACTIBYTI PROJECT SLU, Barcelona
Autor: Ricardo Ahumada

Polymer

- La librería Polymer es un conjunto de polyfills (pieza de código o plugin que permite trabajar con una tecnología que no soporta el navegador de forma nativa) que le ayudan a crear Web Components en todos los principales navegadores. Proporciona el framework para definir, crear y renderizar elementos personalizados complejos de una manera simplista similar a las etiquetas que ya conocemos. Lo que quiero decir con esto, es que ayuda a simplificar la forma en que usamos componentes complejos mediante:
 - Encapsular gran parte del código y la estructura complejos
 - Permitir que los desarrolladores utilicen una convención de nombres de estilo de etiqueta simple de usar
 - Proporcionar un conjunto de elementos de interfaz de usuario predefinidos para aprovechar y ampliar
- Pero lo importante a recordar, es que el propio framework se desarrolla sobre la base de la dirección de las especificaciones individuales que son examinadas por el W3C, proporcionando así una base que debe evolucionar con la dirección del cuerpo normativo principal.

Polymer

- Los Web Components se basan en los siguientes pilares:
 - **Templates:** son fragmentos de HTML que representan al componente de modo que cuando se haga referencia al componente se usará esta plantilla.
 - **HTML Import:** es la posibilidad de incluir un HTML dentro de otro mediante un tag `<link rel="import" href="include.html">`. De este modo el template de un Web Component formará parte del otro.
 - **Shadow DOM:** permite encapsular el subarbol DOM de un componente, de modo que la interacción desde fuera y hacia dentro está controlada. Es importante para que los componentes no entren en conflictos al formar parte de la misma página.
 - **Custom Elements:** para la definición de Tags específicos que representan a los web components que se han creado. En este tutorial haremos el componente contador que estará representado por el tag `<c-contador>` por ejemplo.
- Como puedes imaginar todos estos elementos forman parte de las nuevas especificaciones que van incorporándose a HTML5 y a los navegadores. Aunque se intenta mantener la compatibilidad entre todos ellos no esperes que funcionen en versiones viejas o muy específicas. Antes de usar esta tecnología se deben repasar los requisitos funcionales.



Ejemplo simple de introducción

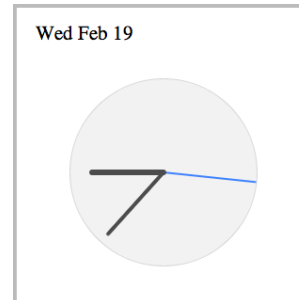
Ejemplo simple



- Esto se parece a la sintaxis de etiquetas HTML con la que ya hemos trabajado y es mucho más fácil de implementar, leer y mantener que algún código JavaScript complejo.

```
<polymer-ui-clock> </polymer-ui-clock>
```

- Y el resultado final se ve así:



- Y puesto que es un elemento normal en el DOM, puedes diseñarlo también utilizando CSS como este:

```
polymer-ui-clock {  
  width: 320px; //ancho  
  height: 320px; //altura  
  display: inline-block; //tipo de vista  
  background: url("../assets/glass.png") no-repeat; //fondo  
  background-size: cover; //tamaño de fondo  
  border: 4px solid rgba(32, 32, 32, 0.3); //bordes }
```

Instalación de Polymer

- Hay tres maneras de instalar y usar Polymer:
 - Utilice el administrador de paquetes de Bower (preferido)
 - Utilizar las librerías alojadas en los cdnjs de CloudFare
 - Instalar desde Git
- De los tres, la manera más fácil y recomendada es utilizar Bower porque no sólo es increíblemente fácil de hacer, sino que Bower también gestiona cualquier dependencia que Polymer pueda tener. Esto significa que si decide instalar un elemento de interfaz de usuario específico que tiene una dependencia de otro, Bower también podrá manejarlo.
- Bower se instala con un módulo empaquetado de node.js, por lo que necesitará tener Node.js instalado. En la línea de comandos, escriba lo siguiente:

```
npm install -g bower
```
- Esto debería extraer Bower del registro npm e instalarlo para que esté disponible a nivel global.

Instalación de Polymer

- Las instalaciones subsiguientes basadas en Bower toman la siguiente forma:

```
bower install
```

- Una vez terminado el proceso de instalación de bower se continua con la intalacion de la plataforma de Polymer y los componentes básicos, ya que estos proporcionan la base para poder crear y ejecutar los elementos.

```
bower install --save Polymer/platform Polymer/polymer
```

- Polymer también viene con un amplio conjunto de elementos predefinidos que se pueden comenzar a aprovechar de inmediato. Consisten en elementos basados en UI y no UI que proporcionan funcionalidad como:
 - Animation
 - Accordions
 - Grid layout
 - Ajax capabilities
 - Menus
 - Tabs

Instalación de Polymer

- Hay una opción para instalar estos componentes. Puede instalar todo o sólo los que desee utilizar. Para instalar todo, escribe:

```
bower install Polymer/core-elements  
bower install Polymer/polymer-ui-elements
```

- Una vez que esté más familiarizado con el framework, se pueden seleccionar los componentes individuales que desee utilizar e instalarlos de esta manera:

```
bower install Polymer/polymer-ui-accordion
```


Estructura de una app usando Polymer

- En primer lugar podemos crear el fichero index.html, que contiene el código de la página que compone la aplicación Web de ejemplo:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Polymer WebApp</title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-scale=1.0, user-scalable=yes">
  <!-- Construcción:css styles/main.css -->
  <link rel="stylesheet" href="styles/main.css">
  <script src="bower_components/platform/platform.js"></script>
  <!-- Construcción:vulcanized elements/elements.vulcanized.html -->
  <link rel="import" href="elements/elements.html">
</head>
<body unresolved>
  <div class="hero-unit">
    <yo-greeting> </yo-greeting>
    <p>You now have</p>
    <yo-list> </yo-list>
  </div>
  <!-- Construcción:js scripts/app.js -->
  <script src="scripts/app.js"></script>
</body>
</html>
```

Estructura de una app usando Polymer

- Como se puede ver se trata de un código muy sencillo en el que, además de las etiquetas <meta>, hojas de estilo y etiquetas básicas que componen una página web clásica, destacan una serie de elementos:
 - Unos “link” que proporcionan a la página la funcionalidad necesaria para operar con Polymer. Estos include son:
 - La referencia a la plataforma de Polymer: `<script src="bower_components/platform/platform.js"></script>`
 - El script de inicio de la aplicación: `<script src="scripts/app.js"></script>`
 - La referencia a los componentes que se utilizan: `<link rel="import" href="elements/elements.html">`
- Para ver la estructura de la aplicación, nos centraremos en el último de los include indicados:

```
<link rel="import" href="elements/elements.html">
```

Estructura de una app usando Polymer

- Nótese que la etiqueta tipo link cuyo atributo rel tiene el valor "import". Esto es una de las características que los nuevos navegadores soportan, lo que permiten cargar el código HTML dentro de la página padre. Este archivo que se incluye "elements/elements.html" contiene las referencias a los elementos (o componentes) de polymer que se van a invocar en la aplicación utilizando las etiquetas de HTML que los representan:

```
<link rel="import" href="yo-list/yo-list.html">  
<link rel="import" href="yo-greeting/yo-greeting.html">  
<link rel="import" href="c-contador/c-contador.html">
```

- Cuando abrimos uno de estos link, por ejemplo yo-greeting, nos encontraremos con la estructura básica de un elemento de polymer.

```
<link rel="import" href="../../bower_components/polymer/polymer.html">  
<polymer-element name="yo-greeting" attributes="">  
  <template>  
    <link rel="stylesheet" href="yo-greeting.css">  
    <h1>{{ greeting }}, {{ greeting }}!</h1>  
    <span>Update text to change the greeting.</span>  
    <input type="text" value="{{ greeting }}">  
  </template>  
  <script>  
    (function () {'use strict';Polymer({greeting : '\Allo'}); })();  
  </script>  
</polymer-element>
```

Estructura de una app usando Polymer

- Podemos destacar los siguientes aspectos del código:
 - Import de la librería de polymer.html, que en realidad hace referencia a polymer.js que es la verdadera librería.
 - La definición del nombre del elemento de polymer “yo-greeting”.
 - El template o representación del HTML del elemento que está formando por un h1, un span y un input. Además incluye su propia hoja de estilos que ha sido especificada en un fichero independiente.
 - Una sección de script que define el modelo y el controlador de ese elemento.
- Como se puede observar, un elemento de Polymer o Web Component está completamente autocontenido, es decir, tiene la vista, tiene el modelo de datos y tiene el controlador o acciones que operan sobre esos datos. Por tanto es totalmente independiente y reutilizables.

Primera App con Polymer

- Crearemos nuestro propio elemento de Polymer, que será un contador con un botón para incrementar el número y otro para disminuir. Por tanto se compondrá de:
 - Template: que contiene dos botones y un input par ver el elemento.
 - Controlador: que inicializa el valor y tiene una función par subir el elemento y otra para bajarlo.
- Si seguimos los pasos indicados en la descripción anterior sobre la estructura de la aplicación de Polymer básica, podremos deducir que tendremos que hacer:
 - Crear un directorio y fichero en elements para nuestro nuevo tag. Como se llamará c-contador, crearemos elements/c-contador/c-contador.html. Copiaremos el ya existente yo-greetings.html para ahorrarr trabajo y lo editaremos.
 - Editar elements.html para incluir la referencia al nuevo fichero que describe nuestro componente.
- Con estos dos sencillos pasos ya estaremos preparados para invocar el componente con el tag dentro de index.html.



Creando un nuevo elemento con Polymer

Creando el fichero c-contador.html



- Este fichero define el componente por completo, tanto apariencia como datos y comportamiento. Nos basamos en uno ya existente. El contenido será el siguiente:

```
<link rel="import" href="../../bower_components/polymer/polymer.html">
<polymer-element name="c-contador">
  <template>
    <link rel="stylesheet" href="c-contador.css">
    <button class="boton" on-click="{{decrementa}}">-</button>
    <input name="contador" type="text" value="{{numero}}" />
    <button href="" class="boton" on-click="{{incrementa}}">+</button>
  </template>
  <script>
(function){'use strict';Polymer('c-contador',{numero:0,decrementa:function() {
  this.numero--; }, incrementa:function() {
    this.numero++; } });
})();
</script>
</polymer-element>
```

Creando el fichero c-contador.html

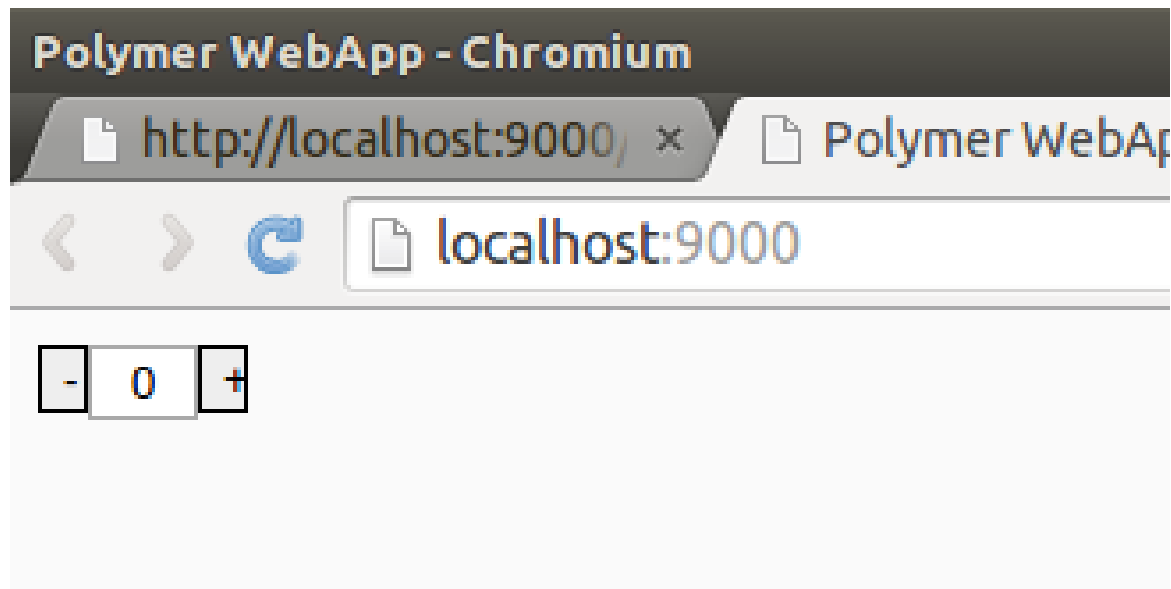


- Los puntos importantes de nuestro componente o elemento de Polymer son:
 - El nombre del componente indicado en la etiqueta *polymer-element* será “c-contador”. A la hora de elegir el nombre tenemos que tener en cuenta que este tiene que llevar un guión.
 - El apartado “template” donde se define la apariencia. En primer lugar tenemos la definición de la hoja de estilos en un fichero externo llamado c-contador.css. Es una buena práctica tenerlo definido fuera, aunque se hubiera podido incluir perfectamente en el mismo código. Por otra parte tenemos dos botones definidos con el tag de HTML “button”. Si te fijas en detalle, ambos tienen referencia a la clase “boton” que está declarada en el fichero CSS. Por otra parte también tienen referencia al atributo *on-click*.
 - La definición del controlador mediante la sección “script” del elemento, que contiene código JavaScript. Si observas bien, por una parte tenemos la definición de la función general que es común para todas las instancias del componente, y dentro de ella la llamada a Polymer pasando como parámetro el nombre del elemento.

Creando el fichero c-contador.html



- Para utilizarlo simplemente tenemos que ir al fichero raíz de la aplicación (index.html) e invocarlo dentro del código mediante la etiqueta. Si eliminamos la referencia a los otros dos elementos, yo-greeting y yo-list, tendremos una página vacía con el contador. Si pulsamos sobre los botones podremos ver cómo sube o baja el contador:



- Lo que acabamos de hacer es poner el componente aislado.

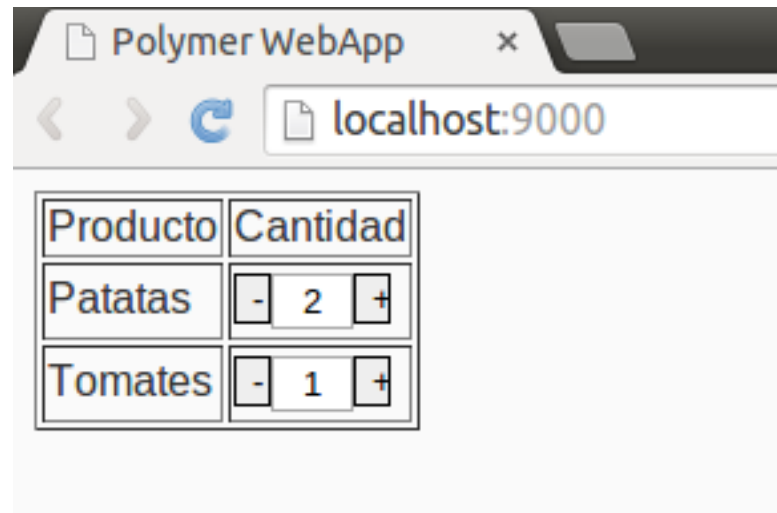
Creando el fichero c-contador.html



- Imagina ahora que tenemos una tabla de productos y cantidades y queremos reutilizar el contador.

```
<table border="1">
  <tr>      <td>Producto</td>      <td>Cantidad</td>      </tr>
  <tr>      <td>Patatas</td>      <td><c-contador id="contadorPatatas"></c-contador></td>      </tr>
  <tr>      <td>Tomates</td>      <td><c-contador id="contadorTomates"></c-contador></td>      </tr>
</table>
```

- Obtenemos la siguiente imagen:





BananaTube: Decisión en equipo

- Discute en equipo la conveniencia de usar Polymer para implementar la SPA de BananaTube
- Sería una buena alternativa a los frameworks que ya conoces?
- Tomad nota de las fortalezas y debilidades que hayas observado en el framework, así como, la dificultad de incorporarlo en el proyecto.



 **netmind**

WeKnowIT

Barcelona

C. Almogàvers, 123
08018 Barcelona
Tel. 93 304.17.20
Fax. 93 304.17.22

Madrid

Plaza Carlos Trías Bertrán, 7
28020 Madrid
Tel. 91 442.77.03
Fax. 91 442.77.07

www.netmind.es



MINISTERIO
DE ENERGÍA, TURISMO
Y AGENDA DIGITAL

red.es



UNIÓN EUROPEA

Fondo Social Europeo
"El FSE invierte en tu futuro"