



**GeeksHubs**  
academy \_

# Web Component

# Web Component V1



# Introducción a Web Componentes

Los Web Components son 4 especificaciones que nos permiten extender el desarrollo para la web por medio de componentes web que cualquier desarrollador podría realizar.

- Custom Elements
- HTML Templates
- HTML Imports
- Shadow DOM

El W3C está encargado de mantener los estándares, pero lo cierto es que sus procedimientos para permitir la evolución de la web son un poco pesados.



# ¿Por qué de los Web Components?



Custom Elements



Templates



Shadow DOM



Modules



CSS Scopes

Los Web Components nos ofrecen un estándar que va enfocado a la creación de todo tipo de componentes utilizables en una página web, para realizar interfaces de usuario y elementos que nos permitan presentar información.

Los propios desarrolladores serán los que puedan, en base a las herramientas que incluye Web Components crear esos nuevos elementos y publicarlos para que otras personas también los puedan usar.



# Ejemplos

Tienen como intención reflejar:

1. Es como si estuviéramos inventando etiquetas nuevas.
2. No tenemos el HTML por un lado, el CSS y el Javascript por otro. Es simplemente la etiqueta nueva y ésta ya es capaz de lanzar el comportamiento.
3. Obviamente, en algún lugar habrá un Javascript que se encargará de procesar esa etiqueta, pero será genérico para cualquier tipo de mapa y reutilizable.
4. Lo que además debe verse es que en el HTML estás colocando información que antes estaría en el Javascript.

```
<google-calendar  
  calendarId="TU_ID_CAL"  
  apiKey="TU_LLAVE_API"  
  busyLabel="busy"  
  freeLabel="Estoy libre">  
</google-calendar>
```



# Versiones del estándar V0 y V1

Este estándar ha sido impulsado principalmente por Google, empresa donde comenzaron el diseño de Web Components, según como ellos mismos consideraban que debería ser.

Sin embargo, para la creación de estándares abiertos, no solamente opina una empresa, sino un conjunto de empresas y profesionales bien posicionados en el sector.

Web Components V1 trajo con sí una novedad principal con respecto la versión V0, la sustitución de los HTML Imports por los **ES6 Module Imports**. Enseguida hablaremos de ello con más detalle.



# Kata I: Investiga librerías que usen Web Components



# Librerías para Web Components

- **Polymer:** Es una librería impulsada por Google que actualmente es el mayor referente en cuanto a Web Components.
- **Stencil:** Es la librería desarrollada por el equipo de Ionic, que pretende ser lo más abierta posible. Se integra muy bien con el novedoso Ionic 4, aunque lo podemos usar donde queramos.
- **X-Tag:** Es la apuesta de Mozilla para la creación de Web Components, específicamente los custom elements, al alcance de todos los navegadores modernos.
- **Bosonic:** Herramientas y utilidades para crear web components incluso en navegadores no tan nuevos, como Internet Explorer 9.





# Custom Elements



# ¿Qué son los Custom Elements?

Básicamente un elemento personalizado es un nuevo elemento de HTML creado por nosotros mismos para resolver cualquier tipo de necesidad que no esté implementada en el HTML estándar.

Los componentes pueden ser sencillos o complejos. Los complejos a menudo se apoyan en componentes más sencillos para resolver las tareas. Una aplicación generalmente será un componente que contiene a su vez un árbol de componentes que se relacionan entre sí y trabajan para resolver un conjunto de necesidades de un negocio.



# Ejemplo de Custom Component con Web Component

Básicamente un elemento personalizado es un nuevo elemento de HTML creado por nosotros mismos para resolver cualquier tipo de necesidad que no esté implementada en el HTML estándar.

```
class HolaComponentes extends HTMLElement {  
  constructor() {  
    super();  
    this.attachShadow( {mode: 'open'} ).innerHTML =  
    '<p>Hola! soy un Custom Element!</p>';  
  }  
}  
  
customElements.define('hola-componentes', HolaComponentes);
```



# Kata II: Crea un CustomComponent

