

Crear un elemento con polymer es muy fácil:

```
<link rel="import" href="../bower_components/polymer/polymer.html">
<!--
Element providing solution to no problem in particular.

##### Example

<my-first-element></my-first-element>

@element my-first-element
-->
<dom-module id="my-first-element">
  <template>
    <h2>Hello</h2>
  </template>

  <script>
    Polymer({
      is: 'my-first-element',
    });
  </script>
</dom-module>
```

Usar este primer elemento en nuestro web (index.html) es también muy muy sencillo:

```
<!doctype html>
<html>
  <head>
    <title>Examples</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="bower_components/webcomponentsjs/webcomponents-lite.js"></script>
    <link rel="import" href="bower_components/polymer/polymer.html">
    <link rel="import" href="bower_components/paper-button/paper-button.html">
    <link rel="import" href="ejsT2/my-first-component.html">

    <style>
      body {
        margin: 0px;
        font-family: 'Roboto', 'Noto', sans-serif;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
      }
    </style>
  </head>

  <body>
    <my-first-element></my-first-element>
  </body>
</html>
```

Vamos a complicarlo un poco añadiendo nuestra primera propiedad:

```
<link rel="import" href="../bower_components/polymer/polymer.html">
<!--
Element providing solution to no problem in particular.

##### Example

    <my-first-element></my-first-element>

@element my-first-element
-->

<dom-module id="my-first-element">
  <template>
    <style>
      :host {
        display: block;
        text-align: center;
        background-color: #BC1E30;
        min-width: 300px;
        color: #fff;
        margin: 15px;
      }
    </style>
    <h2>Hello {{name}}</h2>
  </template>

  <script>
    Polymer({
      is: 'my-first-element',

      properties: {
        name: {
          type: String,
          value: "element",
          reflectToAttribute: true,
          readOnly: false,
          notify: false,
          observer: "_observeName"
        }
      },

      _observeName: function(e){
        console.log("_observeName");
        console.log(e);
      }
    });
  </script>
</dom-module>
```

Ahora nuestro elemento muestra diferentes nombres gracias al binding de la propiedad name a nuestra template con {{name}} y puedo usarla de diferentes modos:

```
<my-first-element name="Victor"></my-first-element>  
<my-first-element name="Carlos"></my-first-element>  
<my-first-element></my-first-element>
```

Esto mostrará respectivamente:

1. Hello Victor
2. Hello Carlos
3. Hello Element (ya que es el valor por defecto asignado en 'value')

Tipos de propiedades

Existen en polymer varios tipos de propiedades (type) que debemos conocer:

1. Boolean
2. Date
3. Number
4. String
5. Array
6. Object

Creando custom elements en js (o desde la consola)

```
var el = document.createElement('my-first-element')  
document.body.appendChild(el)  
el.name = "Peter"
```