



Tópicos de Polymer: Observers y Notificaciones



Acerca de mi



José Luis Iturbide

jose.iturbide@softtek.com

joseluis.iturbide@gmail.com

Ingeniero en Computación de la F.I, UNAM

Arquitecto de Software Certificado con experiencia en proyectos del sector Bancario y de Seguros

Tópicos de Observers y Notificaciones

Objetivo

- Mostrar aquellos detalles del uso de observadores y notificaciones que pueden causar efectos no deseados por su desconocimiento
- Mejorar el conocimiento del uso de conceptos fundamentales para un mayor aprovechamiento de Polymer

Temas de Observadores

Cambios observables

// Modificar directamente la propiedad del elemento

```
this.owner = 'Jane';
```

// Cambiar una subpropiedad del element usando two-way data binding.

```
<local-dom-child name="{{hostProperty.subProperty}}"></local-dom-child>
```

Cambios no Observables

// Cambiar una subpropiedad de un objeto

```
this.address.street = 'Elm Street';
```

// Modificar un arreglo:

```
this.users.push({ name: 'Maturin'});
```

Temas de Observadores

Cambios observables

// Hacer observable cambios en objetos y en arreglos

```
this.set('address.street', 'Half Moon Street');
```

// Hacer observable un cambio en un arreglo

```
this.push('users', { name: 'Maturin' });
```

// Notificar a Polymer que un valor ha cambiado

```
this.address.street = 'Elm Street';
```

```
this.notifyPath('address.street');
```


Temas de Notificaciones

Posibles configuraciones de propiedades

notify. Una propiedad puede notificar cambios en ella en un evento propiedad-changed, por default no esta habilitada. La notificación no va mas alla de los límites del shadowDom.

readOnly. Una propiedad read-only previene cambios desde capas superiores. Por default las propiedades son read/write y soportan downward data flow.

Configuración	Resultado	
notify: false, readOnly: true	No data flow	
notify: false, readOnly: false	One-way, downward	
notify: true, readOnly: true	One-way, upward	
notify: true, readOnly: false	Two-way	

Data binding

Two way, permite flujo de datos upward (target to host) y downwards (host to target). Usa `{{}}`
`<my-input value="{{name}}"></my-input>`

One-way downward, Solo permite flujo de datos downwards. El flujo de datos Upward data flow esta deshabilitado. Usa corchetes `[[]]`.
`<my-input value="[[name]]"></my-input>`

Demo

Aplicación color changer para mostrar el uso de Observers y Notificaciones

Demo de observers y notificaciones

Color generado: (119, 196, 215)

Component Red Value: 119

RED

Component Green Value: 196

GREEN

Component Blue Value: 215

BLUE

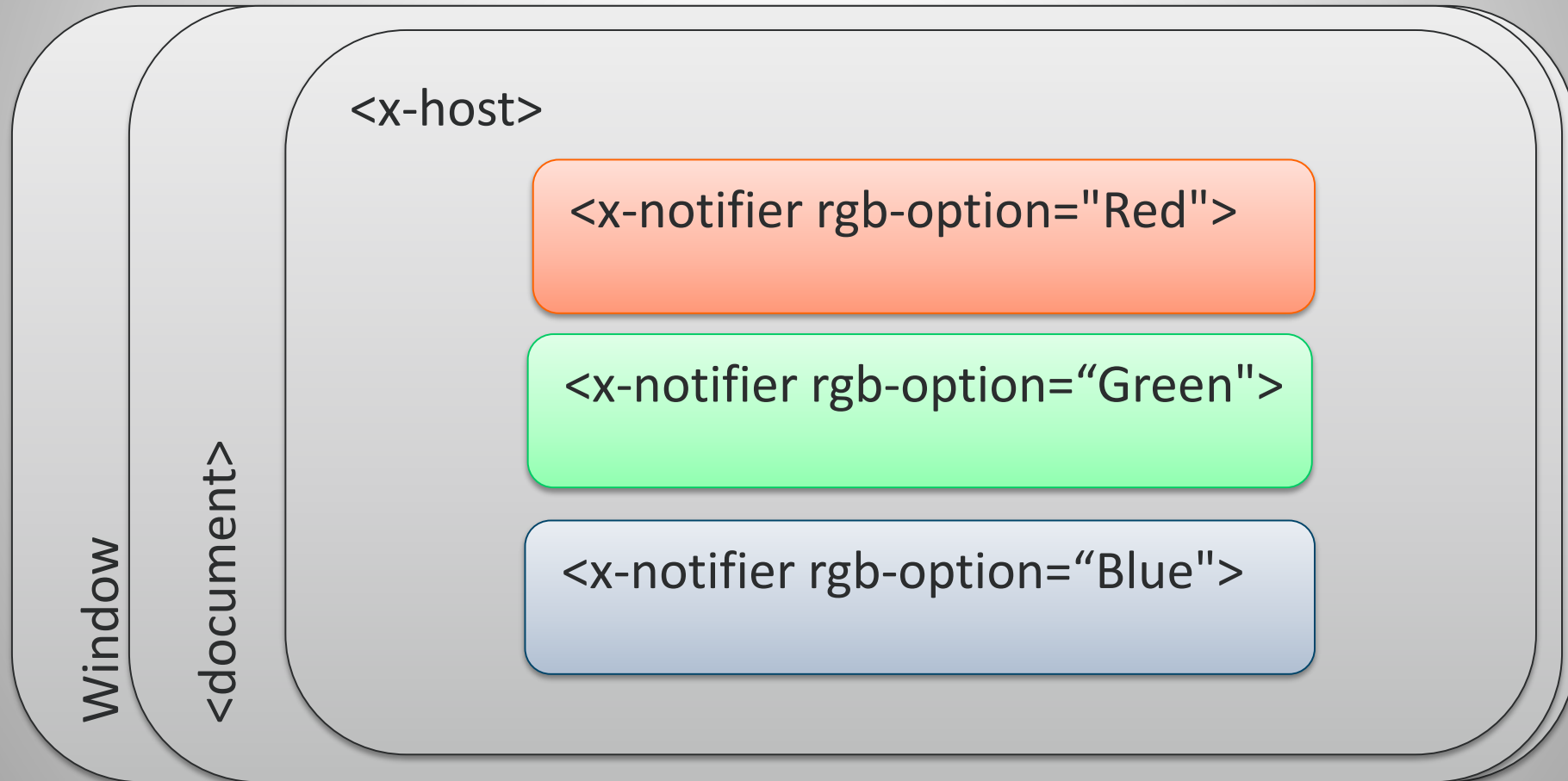
COPY

```
<x-host id="xhost"></x-host><br>
```

<https://github.com/jiturbide/DemoPolythonObservers>

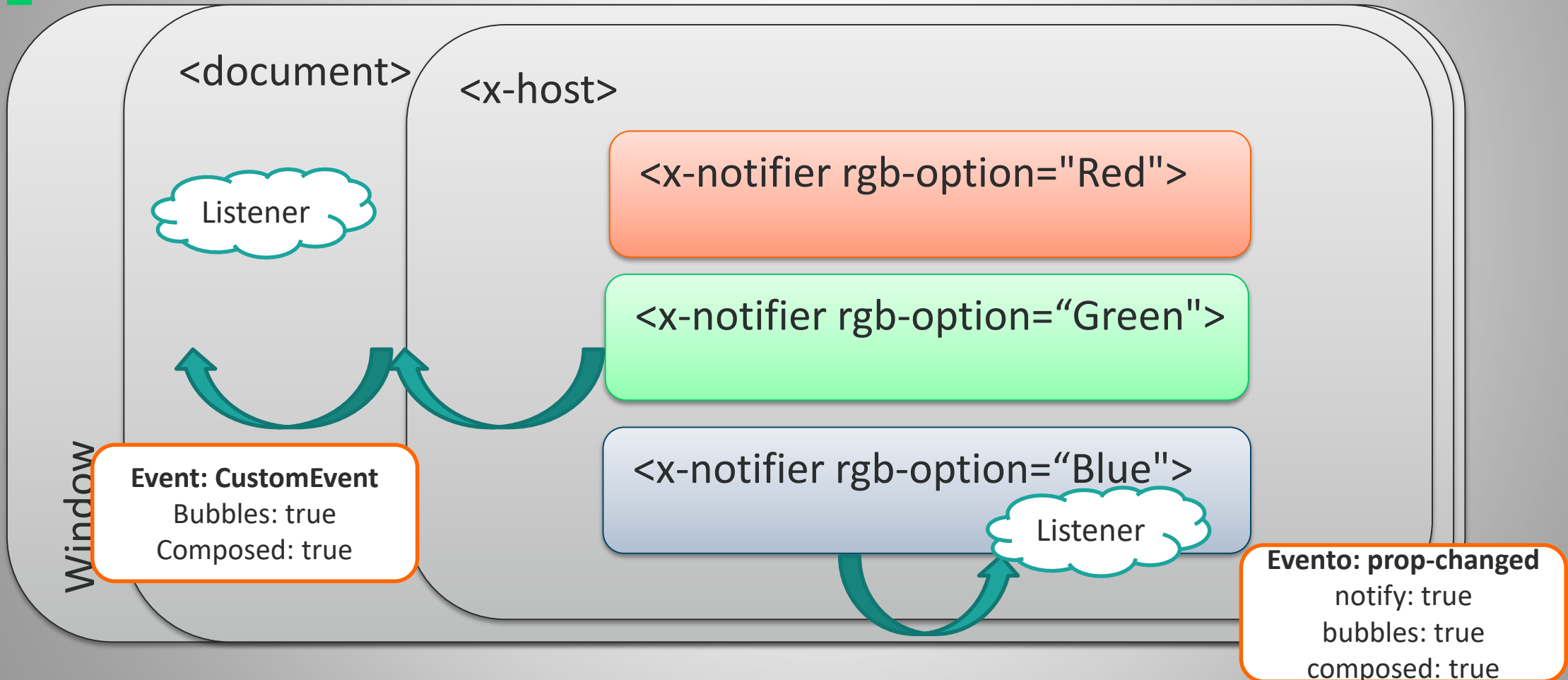
Componentes

Aplicación color changer para mostrar el uso de Observers y Notificaciones



Componentes

Custom Event vs Evento property-Changed



Conclusiones

Que cuidar?

1. Configuración de las propiedades
 - Encapsular, **readOnly** por defecto, no usar **reflectToAttribute**
 - No lanzar notificaciones innecesarias
2. Ubicación adecuada de Listeners
 - **connectedCallback**, **disconnectedCallback**
3. Elegir los eventos adecuados para notificación de datos
 - Elegir **composed** y **bubbles** adecuado
 - Preferentemente con objetos



Polymer

Plática Demo Polyton

Value
THROUGH
TECHNOLOGY