

Existen una serie de helpers muy útiles que debemos conocer: (es importante saber que todos estos helpers son anidables)

dom-repeat

Nos permite iterar arrays para repetir una template determinada, por ejemplo:

```
<template is="dom-repeat" items="{{family}}" as="familymember" index-as="count">
  #{{count}} - {{familymember.name}} {{familymember.lastName}}
</template>

...

properties: {
  family: {
    type: Array,
    value: [
      {name: "Víctor", lastName: "Sánchez", type: "me", hobbies: ["videogames", "tvshows"], isMe: true},
      {name: "Carlos", lastName: "Sánchez", type: "brother", hobbies: ["tvshows", "music"], isMe: false},
      {name: "Diego", lastName: "Sánchez", type: "parent", hobbies: ["gadgets", "music"], isMe: false},
      {name: "María Lucía", lastName: "Belmar", type: "parent", hobbies: ["music", "reading"], isMe: false},
      {name: "Antonia", lastName: "Bonet", type: "grandparent", hobbies: ["pictures", "cooking"], isMe: false},
      {name: "Mar", lastName: "Gómez", type: "godmother", hobbies: ["animals", "concerts"], isMe: false}
    ]
  }
},
```

En este ejemplo vemos como utilizamos el array `family` para iterar cada uno de sus elementos al cual con `as="familymember"` nos permite darle nombre a cada elemento dentro de la iteración. Podemos conocer la posición actual con el atributo `index-as="count"`

En cada vuelta podemos usar la cuenta o referirnos a subpropiedades con `familymember.name` o `familymember.lastName`

Como se explica en el video es muy útil aplicar el modelo de eventos para conocer cual es el item de la template que se ha seleccionado, el uso del modelo puede verse por ejemplo en la función print:

```
_print: function(e) {
  var model = e.model;
  console.log(model);
  console.log(model.familymember);
  console.log(model.familymember.name + " " + model.familymember.lastName);
},
```

dom-if

Nos permite renderizar o no una template en base a si cumple o no con el booleano del atributo `if="{{condición}}"`

```
<template is="dom-if" if="{{familymember.isMe}}">
  <h2>Being you is cool!</h2>
</template>
```

dom-bind

Seguramente el helper menos intuitivo pero también uno de los más prácticos:

```
<body>
  <template is="dom-bind">
    Regular input: <input value="{{name::input}}">

    <my-first-element name="{{name}}">
      <img src='../imgs/avatar.png' />
    </my-first-element>
  </template>
</body>
```

En esta ocasión estamos usando el helper para unir diversos elementos (uno nativo y uno creado con polymer). Las dom-bind template nos permiten que estos elementos se comuniquen entre ellos compartiendo propiedades o subpropiedades que ambos tienen a la 'vista'.