

Veamos nuestro elemento actualizado y algunas nuevas funcionalidades de styling.

En nuestro elemento tengo el siguiente estilo actualizado:

```
<style>
:host {
  display: block;
  text-align: center;
  background-color: var(--my-bg-color, #BC1E30);
  min-width: 300px;
  color: #fff;
  margin: 15px;
}

:host ::content img{
  max-width: 100px;
  border-radius: 50px;
}

#byeP {
  @apply(--my-byeP-rules);
}
</style>
```

y en el elemento intermedio `my-app.html` que contiene toda nuestra aplicación tengo:

```
<link rel="import" href="../bower_components/polymer/polymer.html">
<link rel="import" href="my-first-element.html">

<!--
Element providing solution to no problem in particular.

##### Example

  <my-app></my-app>

@element my-app
-->

<dom-module id="my-app">
  <template>
    <style>
      .purple{
        --my-bg-color: #2F36A9;

        --my-byeP-rules:{
          color: #e813d5;
          text-align: right;
          margin-right: 10px;
          font-size: 12px;
        };
      }
    </style>
```

```

<my-first-element name="Victor" class="purple">
  <img src='../imgs/avatar.png' />
</my-first-element>

<my-first-element name="Carlos">
</my-first-element>
</template>

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

```

Finalmente ahora el índice `index.html` de la app queda de la siguiente manera:

```

<!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="ejsT3/my-app.html">
  </head>

  <body>
    <my-app></my-app>
  </body>
</html>

```

Ahora veo que en nuestro código del estilo del componente `background-color`

```
background-color: var(--my-bg-color, #BC1E30);
```

Utilizo una custom property css3, a la que yo mismo he bautizado como `--my-bg-color` y a la que asigno, para aquellos casos dónde no se indique lo contrario un valor por defecto: `#BC1E30`.

Sin embargo en esta ocasión vemos que los fondos de este elemento (y quizás de muchos otros que usen la misma variable) será ahora purpura si el elemento dispone de la clase `purple`.

```

.purple{
  --my-bg-color: #2F36A9;
  ...

```

En ocasiones no será suficiente personalizar un sólo valor y necesitaremos aplicar varias reglas para que el theming haga lo que nosotros queremos, es el caso de los custom mixins. En nuestro elemento:

```
#byeP {  
  @apply(--my-byep-rules);  
}
```

Y en `my-app.html` usamos este custom mixin de la siguiente manera:

```
--my-byep-rules:{  
  color: #e813d5;  
  text-align: right;  
  margin-right: 10px;  
  font-size: 12px;  
};
```