



## 6.Z. Para saber más

Te proponemos echar un vistazo a la tecnología [Mutation observer](#) y al pequeño ejemplo de código mostrado abajo ya que permite modificar el DOM utilizando eventos de mutación, una forma muy potente y descriptiva de programar.

- Pequeño ejemplo que usa mutation observer y técnicas de programación funcional en JavaScript

```
<!DOCTYPE html>
<meta charset="utf-8"/>
<ol contenteditable oninput="">
  <li>Haz clic en este párrafo y presiona la tecla intro</li>
</ol>
<script>
  var MutationObserver = window.MutationObserver ||
                        window.WebKitMutationObserver ||
                        window.MozMutationObserver;
  var list = document.querySelector('ol');
  var observer = new MutationObserver(function(mutaciones) {
    mutaciones.forEach(function(mutacion) {
      if (mutacion.type === 'childList') {
        var listaDeValores = [].slice.call(list.children)
          .map(function(node) { return node.innerHTML; })
          .filter(function(s) {
            return (s !== '<br>');
          });
        console.log(listaDeValores);
      }
    });
  });

  observer.observe(list, {
    attributes: true,
    childList: true,
    characterData: true
  });
</script>
```

Última modificación: viernes, 31 de mayo de 2019, 10:10

### ◀ 6.Y. Mapa conceptual

Ir a...



Presentación DWEC06 por profesora ▶

