

Um einen Knoten (bzw. ein HTML Element) zu entfernen, muss man das Elternelement und das zu löschende Element kennen und selektieren.

JS



.removeChild() ← Löscht ein HTML Element.

Im Gegensatz zu der CSS Anweisung **display: none;** wird das HTML Dokument gänzlich gelöscht und nicht nur ausgeblendet.

```
<ol id="einkaufsListe">
  <li id="e1">Milch</li>
  <li id="e2">Schokolade</li>
  <li id="e3">Äpfel</li>
</ol>

<button onClick="Entfernen();">Löschen</button>

<script>
  function Entfernen () {
    var elternElement = document.getElementById("einkaufsListe");
    var kindElement = document.getElementById("e2");
    elternElement.removeChild(kindElement);
  }
</script>
```



Warum braucht man das Elternelement? Diese Frage ist berechtigt, da das zu löschende Element sowieso direkt selektiert wird.
Die Antwort ist: Der DOM benötigt es!



Die Lösung jedoch: mit **.parentNode** wird automatisch das Elternelement selektiert. Das Script zum Beispiel oben, sieht dann wie folgt aus:

```
<script>
  function Entfernen() {
    var elternElement = document.getElementById("e2").parentNode;
    var kindElement = document.getElementById("e2");
    elternElement.removeChild(kindElement);
  }
</script>
```

JS



.parentNode ← Selektiert ein Elternelement!

Mit **.parentNode** wird das Elternelement selektiert. Im Beispiel ist das **<div>** das Elternelement von **<button>** - ausgeblendet wird damit das **<div>** und damit auch das **<h1>** und das **<p>** Element.

```
<div>
  <h1>Informationen</h1>
  <p>Vieles zu lesen und noch mehr</p>
  <button onClick="ausblenden(this);">Ausblenden</button>
</div>

<script>
  function ausblenden(kindObjekt) {
    kindObjekt.parentNode.style.display = 'none';
  }
</script>
```