jQuery DOM Navigation

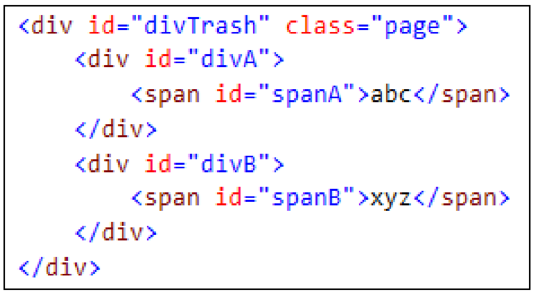
1 Navigation im DOM-Tree

Mit Selektoren kann man Elemente des DOM-Trees auswählen und das zugehörige jQuery-Objekt auf eine Variable zuweisen. Ausgehend davon kann man mit verschiedenen **jQuery-Methoden** zu beliebigen anderen Knoten des DOM-Trees **navigieren**. Sie sind stark an die Javascript-Array-Methoden angelehnt. Die wichtigsten dieser Funktionen sind hier aufgelistet.

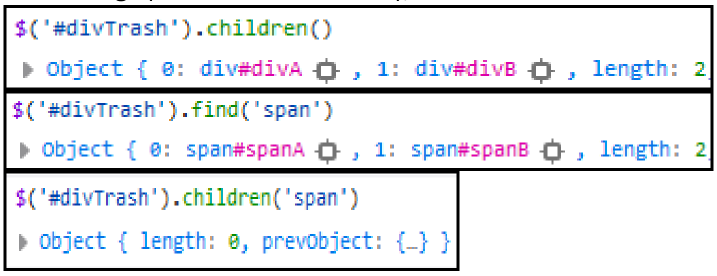
1.1 Descendants (Nachfolgeknoten)

`.find(selector)`	Descendant elements that match the selector.
`.contents()`	Child nodes (including text nodes).
`.children([selector])`	Child nodes, optionally filtered by a selector.

Bei **`children`** werden nur die **direkten** Kinder berücksichtigt (bei **`find`** aber nicht), also z.B.:



```
<div id="divTrash" class="page">
    <div id="divA">
        <span id="spanA">abc</span>
    </div>
    <div id="divB">
        <span id="spanB">xyz</span>
    </div>
</div>
```



```
$('#divTrash').children()
▶ Object { 0: div#divA , 1: div#divB , length: 2
$('#divTrash').find('span')
▶ Object { 0: span#spanA , 1: span#spanB , length: 2
$('#divTrash').children('span')
▶ Object { length: 0, prevObject: {…} }
```

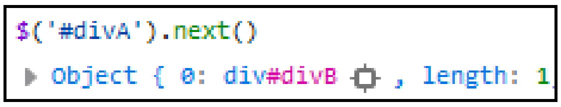
Also:

- children: direkte Unterelemente
- descendants: alle Unterelemente

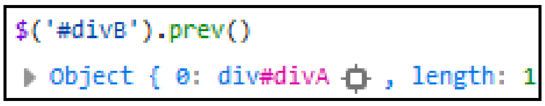
1.2 Siblings (Geschwisterknoten)

`.next([selector])`	The sibling immediately following each selected element, optionally filtered by a selector.
`.nextAll([selector])`	All siblings following each selected element, optionally filtered by a selector.
`.nextUntil([selector], [filter])`	All siblings following each selected element up to and not including the first element matching selector, optionally filtered by an additional selector.
`.prev([selector])`	The sibling immediately preceding each selected element, optionally filtered by a selector.
`.prevAll([selector])`	All siblings preceding each selected element, optionally filtered by a selector.
`.prevUntil([selector], [filter])`	All siblings preceding each selected element up to and not including the first element matching selector, optionally filtered by an additional selector.
`.siblings([selector])`	All siblings, optionally filtered by a selector.

Mit dem Beispiel von oben also:

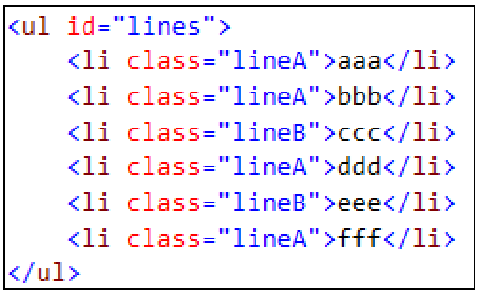


```
$('#divA').next()
▶ Object { 0: div#divB , length: 1
```

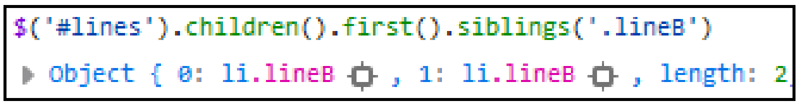


```
$('#divB').prev()
▶ Object { 0: div#divA , length: 1
```

Oder:



```
<ul id="lines">
    <li class="lineA">aaa</li>
    <li class="lineA">bbb</li>
    <li class="lineB">ccc</li>
    <li class="lineA">ddd</li>
    <li class="lineB">eee</li>
    <li class="lineA">fff</li>
</ul>
```



```
$('#lines').children().first().siblings('.lineB')
▶ Object { 0: li.lineB , 1: li.lineB , length: 2
```

1.3 Ancestors (Vorgängerknoten)

`.parent([selector])`	The parent of each selected element, optionally filtered by a selector.
`.parents([selector])`	All ancestors, optionally filtered by a selector.
`.parentsUntil([selector], [filter])`	All ancestors of each selected element up to and not including the first element matching selector, optionally filtered by an additional selector.
`.closest(selector)`	The first element that matches the selector, starting at the selected element and moving up through its ancestors in the DOM tree.
`.offsetParent()`	The positioned parent, either relative or absolute of the first selected element.

Hier verhält es sich also sehr ähnlich wie bei children/descendants:

- parent: direkter Vorgänger
- parents: alle Vorgänger