

Learn jQuery: Traversing the DOM

jQuery children

The jQuery `.children()` method returns all child elements of a selected parent element.

This method only applies to the direct children of the parent element, and not deeper descendants.

In the example code,

`$('.parent').children()` would select all the `.item` elements.

```
<div class="parent">
  <div class="item">Child 1</div>
  <div class="item">Child 2</div>
  <div class="item">Child 3</div>
</div>
```

jQuery .parent

The jQuery `.parent()` method returns the parent element of a jQuery object.

```
<ul>ul <!-- this is the parent of li's
one, two, six and ul three -->
  <li class="one">li</li>
  <li class="two">li</li>
  <ul class="three"> <!-- this is the
parent of li's four and five -->
    <li class="four">li</li>
    <li class="five">li</li>
  </ul>
  <li class="six">li</li>
</ul>
```

jQuery .next

The jQuery `.next()` method targets the next element that shares the same parent element as the original element.

In the following HTML code, the element returned by

`$('.two').next()` would be `<li class="three">Item three`.

```
<ul>
  <li class="one">Item one</li>
  <li class="two">Item two</li>
  <li class="three">Item three</li>
</ul>
```

jQuery .find()

In jQuery, the `.find()` method will find and return all descendent elements that match the selector provided as an argument.

This code block shows a snippet of HTML that has a simple shopping list. Using jQuery, the list items inside the shopping list can be selected. The `listItems` variable will be a jQuery object that contains the two list items from the shopping list.

```
/*
In HTML:
<ul id='shopping-list'>
  <li class='list-item'>Flour</li>
  <li class='list-item'>Sugar</li>
</ul>
*/

// jQuery:
const listItems = $('#shopping-
list').find('.list-item');
```

jQuery .siblings

The jQuery `.siblings()` method targets all of the sibling elements of a particular element.

`.siblings()` can be used to add a `selected` class to an element on click and remove it from all of its sibling elements, ensuring that only one element appears as “selected” at one time.

```
$('.choice').on('click', event => {
  // Remove the 'selected' class from
  any siblings
  $(event.currentTarget).siblings().rem
oveClass('selected');
  // Adds 'selected' class to that
  element only.
  $(event.currentTarget).addClass('sele
cted');
});
```

jQuery .closest

The jQuery `.closest()` method travels up through the DOM tree to find the first (and closest) ancestor element matching a selector string.