Search...

Getting started
Components

M4Q
About
Population
Constructor
Ajax
Animation
Loops
Visibility
Effects
Subtree functions
Attributes
Html, text and value
Css and classes
Position and size
Manipulation
DataSet
Events
Utils

Base
Containers
Grid system
Typography
Tables
Forms
Buttons
Images
Figures
Lists

Form components
Checkbox
File input
Input
Input material
Keypad
Rating
Radio
Select
Slider
Double Slider
Spinner
Switch
Tag input
Textarea

Menus
App bar
Bottom navigation
Bottom sheet
Menu
Ribbon Menu
Side bar
Side navigation

Controls
Accordion
Badge
Carousel
Cards
Cube
Counter
Charms
Chat
Donut
Image compare
Image magnifier
Gravatar
List
ListView
Master
NavView
Panels
Progress & Activity
Streamer
Stepper
Splitter
Tabs
Tabs material
Table
Tiles
TreeView
Wizard

Information
Dialogs
Info box
Hints
Notify system
Popovers
Toasts
Windows

Date & time
Calendar
Calendar picker
Date picker
Time picker
Countdown
Formatting date

M4Q Subtree functions

Use subtree functions to search sub-elements in the set.

Table of contents

About

M4Q implements a set of functions to search for sub-items in the set: `index()`, `get()`, `eq()`, `contains()`, `is()`, `same()`, `find()`, `children()`, `parent()`, `parents()`, `closest()`, `siblings()`, `prev()`, `prevAll()`, `next()`, `nextAll()`, `filter()`, `last()`, `first()`, `even()`, `odd()`, `clone()`.

index()

Return value is an integer indicating the position of the first element.

```
<ul>
  <li id="foo">foo</li>
  <li id="bar">bar</li>
  <li id="baz">baz</li>
</ul>
```

```
var listItem = $( "#bar" );
alert( "Index: " + listItem.index() ); // Index: 1
```

```
var listItem = document.getElementById( "bar" );
alert( "Index: " + $( "li" ).index( listItem ) ); // Index: 1
```

```
var listItem = $( "#bar" );
alert( "Index: " + $( "li" ).index( listItem ) ); // Index: 1
```

eq()

Reduce the set of matched elements to the one at the specified index.

```
<ul>
  <li id="foo">foo</li>
  <li id="bar">bar</li>
  <li id="baz">baz</li>
</ul>
```

```
console.log( $( "li" ).eq(1) ); // [li#bar]
console.log( $( "li" ).eq(-1) ); // [li#baz]
```

get()

Return element as HTMLElement by index. When index is negative, function return element from and of array of elements

```
<ul>
  <li id="foo">foo</li>
  <li id="bar">bar</li>
  <li id="baz">baz</li>
</ul>
```

```
console.log( $( "li" ).get(1) ); // <li id="bar">
console.log( $( "li" ).get(-1) ); // <li id="baz">
```

find()

Get the descendants of each element in the current set of matched elements, filtered by a selector.

```
<ul>
    <li id="foo"><span class="foo">foo</span></li>
    <li id="bar"><span class="bar">bar</span></li>
    <li id="baz"><span class="baz">baz</span></li>
</ul>
```

```
console.log( $( "li" ).find("span") ); // [span, span, span]
console.log( $( "ul" ).find(".bar, .baz") ) // [span.bar, .span.baz];
```

children()

Get the children of each element in the set of matched elements, optionally filtered by a selector.

```
<ul>
    <li id="foo"><span class="foo">foo</span></li>
    <li id="bar"><span class="bar">bar</span></li>
    <li id="baz"><span class="baz">baz</span></li>
</ul>
```

Media
Video player
Audio player
Tools
Collapse
Color module
Draggable
Dropdown
Form validator
Hotkeys
Micro templates
Ripple
Storage
Session storage
Sorter
Touch and swipe

Utilities
Animations
Additional

```
console.log( $( "li" ).children(".bar, .foo") ); // [span.foo, span.bar]
console.log( $( "ul" ).children("#bar") ) // [li#bar];
```

contains()

Return `true`, if m4q object contains elements filtered by selector.

```
<ul>
    <li id="foo"><span class="foo">foo</span></li>
    <li id="bar"><span class="bar">bar</span></li>
    <li id="baz"><span class="baz">baz</span></li>
</ul>
```

```
console.log( $( "li" ).contains(".bar") ); // true
console.log( $( "li" ).contains("#bar") ); // false
```

is()

Check the current matched set of elements against a selector, element, or m4q object and return true if at least one of these elements matches the given arguments. Also, for selectors: `:selected`, `:checked` and `:hidden` return true, if first element in set matches to it.

match selector

```
<ul>
    <li id="foo"><span class="foo">foo</span></li>
    <li id="bar"><span class="bar">bar</span></li>
    <li id="baz"><span class="baz">baz</span></li>
</ul>
```

```
console.log( $( "#bar" ).is("li") ); // true
```

:checked

```
<input type="checkbox" id="check1" checked>
<input type="checkbox" id="check2" >
```

```
console.log( $( "#check1" ).is(":checked") ); // true
console.log( $( "#check2" ).is(":checked") ); // false
```

:hidden

```
<div style="display: none">This is a hidden div</div>
<div style="opacity: 0">This is a hidden div</div>
<div style="visibility: hidden">This is a hidden div</div>
<div hidden>This is a hidden div</div>
```

```
console.log( $( "div" ).is(":hidden") ); // true
```

:selected

```
<select>
    <option>Item 1</option>
    <option selected>Item 2</option>
    <option>Item 3</option>
</select>
```

```
$.each($("option"), function(){
    if ($(this).is(":selected")) {
        console.log("Selected option:", this.value);
    }
});
```

same()

Checks the current matched set of elements with the m4q object and return true if matched set equivalent to checked object.

```
<ul>
    <li id="foo"><span class="foo">foo</span></li>
    <li id="bar"><span class="bar">bar</span></li>
    <li id="baz"><span class="baz">baz</span></li>
</ul>
```

```
console.log( $( "#bar" ).same( $("li#bar") ) ); // true
console.log( $( "#bar" ).same( $("div") ) ); // false
```

parent()

Get the parent of each element in the current set of matched elements, optionally filtered by a selector.

```
<ul class="level-1">
  <li class="item-i">I</li>
  <li class="item-ii">II
    <ul class="level-2">
```

```
      <li class="item-a">A</li>
      <li class="item-b">B
        <ul class="level-3">
          <li class="item-1">1</li>
          <li class="item-2">2</li>
          <li class="item-3">3</li>
        </ul>
      </li>
      <li class="item-c">C</li>
    </ul>
  </li>
  <li class="item-iii">III</li>
</ul>
```

```
$( "li.item-a" ).parent().css( "background-color", "red" );
```

parents()

Get the ancestors of each element in the current set of matched elements, optionally filtered by a selector.

```
<div>
    <p>
    <span>
      <b>My parents are: </b>
    </span>
    </p>
</div>
```

```
var parentEls = $( "b" ).parents().items().map(function(el){
    return el.tagName;
}).join(", ");

$( "b" ).append( "<strong>" + parentEls + "</strong>" );

// Outputs: My parents are: SPAN, P, DIV, BODY, HTML
```

closest()

For each element in the set, get the first element that matches the selector by testing the element itself and traversing up through its ancestors in the DOM tree.

```
<ul id="one" class="level-1">
  <li class="item-i">I</li>
  <li id="ii" class="item-ii">II
    <ul class="level-2">
      <li class="item-a">A</li>
      <li class="item-b">B
        <ul class="level-3">
          <li class="item-1">1</li>
          <li class="item-2">2</li>
          <li class="item-3">3</li>
        </ul>
      </li>
      <li class="item-c">C</li>
    </ul>
  </li>
  <li class="item-iii">III</li>
</ul>
```

Suppose we perform a search for <ul> elements starting at item A:

```
$( "li.item-a" )
  .closest( "ul" )
  .css( "background-color", "red" );
```

siblings()

Get the siblings of each element in the set of matched elements, optionally filtered by a selector.

```
<div>
    <div>
        <span><b>Bold text</b></span>
        <div>I'am div</div>
        <p>123</p>
    </div>
</div>
```

```
console.log($("span").siblings()); // Outputs: [div, p]
console.log($("span").siblings("p")); // Outputs [p]
```

prev()

Get the immediately preceding sibling of each element in the set of matched elements. If a selector is provided, it retrieves the previous sibling only if it matches that selector.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third-item">list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
</ul>
```

```
$( "li.third-item" ).prev().css( "background-color", "red" );
```

prevAll()

Get the preceding sibling of each element in the set of matched elements. If a selector is provided, it retrieves the sibling only if it matches that selector.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third-item">list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
</ul>
```

```
$( "li.third-item" ).prev().css( "background-color", "red" );
```

next()

Get the immediately following sibling of each element in the set of matched elements. If a selector is provided, it retrieves the next sibling only if it matches that selector.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third-item">list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
</ul>
```

```
$( "li.third-item" ).next().css( "background-color", "red" );
```

nextAll()

Get the following sibling of each element in the set of matched elements. If a selector is provided, it retrieves the sibling only if it matches that selector.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third-item">list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
</ul>
```

```
$( "li.third-item" ).nextAll().css( "background-color", "red" );
```

filter()

Reduce the set of matched elements to those that match the selector or pass the function's test.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
// get odd elements
$("li").filter(function(el, i){
    return i % 2 === 0;
}).css("background-color", "red");
```

last()

Reduce the set of matched elements to the last element in set.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
$("li").last().css("background-color", "red");
```

first()

Reduce the set of matched elements to the first element in set.

```
<ul>
```

```
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
$("li").first().css("background-color", "red");
```

even()

Reduce the set of matched elements to elements with even index.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
$("li").even().css("background-color", "red");
```

odd()

Reduce the set of matched elements to elements with odd index.

```
<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li>list item 3</li>
  <li>list item 4</li>
  <li>list item 5</li>
  <li>list item 6</li>
</ul>
```

```
$("li").odd().css("background-color", "red");
```