

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

Utils

Use utilities functions to improve your work.

\$.merge()

Merge the contents of two arrays together into the first array.

```
var newArray = $.merge( [], oldArray );
var newMatches = $.merge( $(), $( "div" ) );
var arrayMatches = $.merge( [], $( "div" ) );
var combineMatches = $.merge( $( "span" ), $( "div" ) );
```

\$.import()

Import any elements array like objects into `m4q` object.

```
var obj = $.import( jQuery( "span" ) );
var obj = $.import( document.querySelectorAll( "span" ) );
```

\$.uniqueId()

Get unique id in GUID format `xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx` .

```
var GUID = $.uniqueId();
```

\$.toArray()

Convert any array like object to array.

```
var arr = $.toArray( document.querySelectorAll( "span" ) );
var arr = $.toArray( $(span) );
```

\$.type()

Get object symple type name.

```
console.log( $.type( $( ".inner" ) ) ); // Outputs: object
console.log( $.type( [] ) ); // Outputs: array
console.log( $.type( "" ) ); // Outputs: string
console.log( $.type( 123 ) ); // Outputs: number
console.log( $.type( document.querySelectorAll( "span" ) ) ); // Outputs: nodelist
```

\$.sleep()

Suspend script execution for specified number of milliseconds.

```
console.log("Step 1 and sleep 1 sec");
$.sleep(1000);
console.log("Step 2");
```

\$.isSelector()

Check if string is a valid simple query selector.

```
$.isSelector( ".inner" ); // true
$.isSelector( "#nner" ); // true
$.isSelector( ".inner.first" ); // true
$.isSelector( "<div>" ); // false
$.isSelector( "<p>This is paragraph</p>" ); // false
```

\$.remove()

Remove matched elements from DOM.

```
$.remove( "span" );
$.remove( ".inner" );
$.remove( "#inner" );
```

\$.camelCase()

Convert dashed name to camel-case name.

```
console.log( $.camelCase( "data-animate" ) ); // Outputs: dataAnimate
```

\$.isPlainObject()

Check if object is a plain object.



- 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

```
var obj = {prop: "value"};
var arr = [1, 2, 3];
var $obj = $("body");
var str = "body";

console.log( $.isPlainObject( obj ) ); // Outputs: true
console.log( $.isPlainObject( arr ) ); // Outputs: false
console.log( $.isPlainObject( $obj ) ); // Outputs: true
console.log( $.isPlainObject( str ) ); // Outputs: false
console.log( $.isPlainObject( String(str) ) ); // Outputs: false
```

\$.isEmptyObject()

Check if object is empty.

```
var obj = {}, obj2 = {prop: 1};
$.isEmptyObject( obj ); // true
$.isEmptyObject( obj2 ); // false
```

\$.isArrayLike()

Check if object is array like object.

```
var obj = {};
var $obj = $("body");
var arr = [];

console.log( $.isArrayLike(obj) ); // false
console.log( $.isArrayLike($obj) ); // true
console.log( $.isArrayLike(arr) ); // true
```

\$.acceptData()

Check if HTMLElement can accept data with `m4q` dataset routines.

```
$.acceptData( document.querySelector("#elem" ) );
```

\$.not()

Check if variable has a value.

```
var nul = null;
var val = 1;

function und(val) {
    console.log( $.not(val) );
}

und(); // Outputs true
console.log( $.not(nul) ); // Outputs: true
console.log( $.not(val) ); // Outputs: false
```

\$.unit(), \$.parseUnit()

Parse element style property value to array with value and unit name.

```
console.log( $.unit("1rem") ); // Outputs [1, "rem"]
console.log( $.unit(1) ); // Outputs [1, ""]
```

\$.isVisible

Check if HTMLElement can visible on page.

```
<div class="vis1"></div>
<div class="vis2" style="display: none"></div>
<div class="vis3" hidden></div>
```

```
console.log( $.isVisible( $(".vis1")[0] ) ); // Outputs true
console.log( $.isVisible( $(".vis2")[0] ) ); // Outputs false
console.log( $.isVisible( $(".vis3")[0] ) ); // Outputs false
```

\$.isHidden

Check if HTMLElement is hidden on page.

```
<div class="vis1">123</div>
<div class="vis2" style="display: none">123</div>
<div class="vis3" hidden>123</div>
<div class="vis4" style="visibility: hidden">123</div>
<div class="vis5" style="opacity: 0">123</div>
```

```
console.log( $.isHidden( $(".vis1")[0] ) ); // Outputs false
console.log( $.isHidden( $(".vis2")[0] ) ); // Outputs true
console.log( $.isHidden( $(".vis3")[0] ) ); // Outputs true
console.log( $.isHidden( $(".vis4")[0] ) ); // Outputs true
console.log( $.isHidden( $(".vis5")[0] ) ); // Outputs true
```

`$(...).items()`

Return array of elements from `m4q` object.

```
$("body"); // m4q object  
$("body").items(); // array
```

