Search...

M4Q Events

Attach or remove an event handler function for one or more events to the selected elements.

About

You can use built in methods: `on`, `one`, `off`, `trigger` and `fire` for works with events.

on()

Attach an event handler function for one or more events to the selected elements. You can define more than one event, with space-separated event types and optional namespaces. Also, you can add `options` to you your event with last argument `option`. Argument `option` must be a plain object and can contains:

- `ns` - namespace for event
- `once` - false (default) or true to define once fired event

```
var data = {...};

$(...).on("click", function(e){...});
$(...).on("mousedown touchstart", function(e){...});
$(...).on("click", function(e){...}, {once: true});
$(...).on("click", "selector", function(e){...});
$(...).on("click", "selector", function(e){...}, {ns: "unique_ns"});
```

Also, you can define `ns` for event with point notation: `event_name.ns_name`.

```
$(...).on("click.ns_name", function(e){...});
```

one()

Attach a handler to an event for the elements. The handler is executed at most once per element per event type. After event firing, the event handler will be removed. You can define more than one event, with space-separated event types and optional namespaces.

```
var data = {...};

$(...).one("click", function(e){...});
$(...).one("mousedown touchstart", function(e){...});
$(...).one("click", function(e){...}, options);
$(...).one("click", "selector", function(e){...});
$(...).one("click", "selector", function(e){...}, options);
```

off()

Remove an event handler. You can define more than one event, with space-separated event types and optional namespaces. If you can't define event name, will be removed all events.

```
$(...).off();
$(...).off("click");
$(...).off("mousedown touchstart");
$(...).off("click", "selector");
$(...).off("click", "selector", {ns: "ns_name"});
```

In M4Q, when you call `off` method, this method remove event handler, including, anonymous function for handler.

```
// add anonymous click handler
$(el).on("click", function(){
    console.log("ku");
});

$(el).off("click"); // This call remove anonymous click handler from element.
```

trigger()

Execute all handlers and behaviors attached to the matched elements for the given event type. Unlike jQuery, you can trigger the execution of not only predefined events, but also any. Also, you can define custom data and push it to the event with `data` argument. If you define custom event data, you can access to it with `e.data` property in the handler.

```
$(...).trigger("click");
...
$(...).trigger("click", {
    cls: "myClass"
});
...
$(...).on("click", function(e){
    $(e.target).addClass(e.data.cls);
})
```

fire()

Table of contents

Execute all handlers and behaviors attached to the matched elements for the given event type. Unlike jQuery, you can trigger the execution of not only predefined events, but also any. Also, you can define custom data and push it to the event with `data` argument. If you define custom event data, you can access to it with `e.detail` property in the handler.

```
$(...).fire("click");
...
$(...).fire("myCustomEvent", {
    cls: "myClass"
});
...
$(...).on("myCustomEvent", function(e){
    $(e.target).addClass(e.detail.cls);
})
```

getEvents()

Return all attached event with methods `on()` and `one()`.

```
console.log( $.getEvents() );
```

addEventHook()

You can define a `hook` that will trigger before or after the occurrence of an event. The hook is triggered for all elements that have the specified event and its handler was installed using the `on()`, `one()` methods. You can define more than one event, with space-separated event types.

```
$.addEventHook("click", function(e){...}); // add hook before (default) event click
$.addEventHook("mousedown touchstart", function(e){...}); // add hook before (default) event click
$.addEventHook("click", function(e){...}, "before"); // add hook before event click
$.addEventHook("click", function(e){...}, "after"); // add hook after event click
```

removeEventHook()

Remove an hook for event(s).

```
$.removeEventHook("click");
$.removeEventHook("click", "before");
$.removeEventHook("click", "after");
$.removeEventHook("mousedown touchstart");
```

removeEventHooks()

Remove all hook for event(s).

```
$.removeEventHooks();
```

Event aliases

M4Q contains short aliases for standard events: `blur`, `focus`, `resize`, `scroll`, `click`, `dblclick`, `mousedown`, `mouseup`, `mousemove`, `mouseover`, `mouseout`, `mouseenter`, `mouseleave`, `change`, `select`, `submit`, `keydown`, `keypress`, `keyup`, `contextmenu`, `touchstart`, `touchend`, `touchmove`, `touchcancel`.

```
var data = {...};

$(...).click(function(){})
$(...).click("selector", function(){})
$(...).click("selector", function(){}, data)
```

Document ready

Create document ready function. You can specify function as constructor argument and receive document ready function or use `$.ready(...)` method.

```
$(function(){
    ...
});

or

$.ready(function(){
    ...
});
```

equals to

```
document.addEventListener('DOMContentLoaded', function(){
    ...
});
```

$.load

Create `window.load` function.

```
$.load(function(e){
    ...
});

or
```

```
$(window).load(function(e){
    ...
});
```

equals to

```
window.onload = function(e){
    ...
});
```

$.unload

Create `window.unload` function.

```
$.unload(function(e){
    ...
});

or

$(window).unload(function(e){
    ...
});
```

equals to

```
window.onunload = function(e){
    ...
});
```

$.beforeunload

Create `window.beforeunload` function.

```
$.beforeunload("Do you really want to leave the page?");
```

```
$.beforeunload(function(e){
    var str = "Do you really want to leave the page?";
    e.returnValue = str;
    return str;
});
```

```
$(window).beforeunload("Do you really want to leave the page?");
```

```
$(window).beforeunload(function(e){
    var str = "Do you really want to leave the page?";
    e.returnValue = str;
    return str;
});
```

equals to

```
window.onbeforeunload = function(e){
    ...
});
```