# Jquery bind double click and single click separately

Asked 9 years, 6 months ago Active 2 years, 7 months ago Viewed 101k times

97

40

Is there something in jquery that would allow me to differentiate between behavior on double click and single click?

When I bind both to same element only the single click gets executed.

Is there a way that wait for some time before execution of the single click to see if the user clicks again or not?

Thanks :)

jquery jquery-selectors

edited Aug 19 '13 at 6:47
Melanie
2,352 ●6 ●26 ●46

asked Jun 13 '11 at 12:31
kritya
3,112 ●6 ●35 ●58

1 stackoverflow.com/questions/1472433/… – Haim Evgi Jun 13 '11 at 12:37

## 14 Answers

Active Oldest Votes

145

✓

I found that John Strickler's answer did not quite do what I was expecting. Once the alert is triggered by a second click within the two-second window, every subsequent click triggers another alert until you wait two seconds before clicking again. So with John's code, a triple click acts as two double clicks where I would expect it to act like a double click followed by a single click.

I have reworked his solution to function in this way and to flow in a way my mind can better comprehend. I dropped the delay down from 2000 to 700 to better simulate what I would feel to be a normal sensitivity. Here's the fiddle: http://jsfiddle.net/KpCwN/4/.

Thanks for the foundation, John. I hope this alternate version is useful to others.

```
var DELAY = 700, clicks = 0, timer = null;

$(function(){

    $("a").on("click", function(e){

        clicks++;  //count clicks

        if(clicks === 1) {

            timer = setTimeout(function() {

                alert("Single Click");  //perform single-click action
                clicks = 0;             //after action performed, reset counter

            }, DELAY);
        } else {

            clearTimeout(timer);    //prevent single-click action
            alert("Double Click");  //perform double-click action
            clicks = 0;             //after action performed, reset counter
        }

    })
    .on("dblclick", function(e){
        e.preventDefault();  //cancel system double-click event
    });

});
```

edited May 7 '13 at 18:03
agrublev
669 ● 1 ● 8 ● 21

answered Oct 21 '11 at 5:22
Garland Pope
3,103 ● 1 ● 21 ● 19

1 Use .delegate() over .live() though. – John Strickler Nov 3 '11 at 18:08

24 use `.on()` over both of those now – Claudiu Sep 4 '12 at 15:40

2 Using this solution delays the click event in 700ms! It is very annoying to user... – uriel Jun 25 '15 at 5:39

1 @uriel, if you think 700 ms is too long to wait for a double click, just make the number smaller until it suits your preference. – Garland Pope Jul 17 '15 at 20:32

1 as of jQuery 1.7: In order to support later-on loaded content (ajax stuff) use `$("document").on("click","a" function(e){` instead in line 5. This construct replaces deprecated `delegate()` usage. Instead of `document` you can/should use any parent object of the `a` down the DOM as far as possible that persists trough time. – Hafenkranich Oct 20 '16 at 17:41

15

The solution given from "Nott Responding" seems to fire both events, click and dblclick when doubleclicked. However I think it points in the right direction.

I did a small change, this is the result :

```
$("#clickMe").click(function (e) {
    var $this = $(this);
    if ($this.hasClass('clicked')){
        $this.removeClass('clicked');
        alert("Double click");
        //here is your code for double click
    }else{
        $this.addClass('clicked');
        setTimeout(function() {
            if ($this.hasClass('clicked')){
                $this.removeClass('clicked');
                alert("Just one click!");
                //your code for single click
            }
        }, 500);
    }
});
```

Try it

http://jsfiddle.net/calterras/xmmo3esg/

edited Apr 18 '18 at 21:14
Wictor Chaves
535 ● 1 ● 9 ● 16

answered Oct 10 '14 at 16:48
calterras
198 ● 1 ● 4

10

Sure, bind two handlers, one to `click` and the other to `dblclick`. Create a variable that increments on every click. then resets after a set delay. Inside the setTimeout function you can do something...

```
var DELAY = 2000,
    clicks = 0,
    timer = null;

$('a').bind({
    click: function(e) {
        clearTimeout(timer);

        timer = setTimeout(function() {
            clicks = 0;
        }, DELAY);

        if(clicks === 1) {
            alert(clicks);
             //do something here

            clicks = 0;
        }

        //Increment clicks
        clicks++;
    },
    dblclick: function(e) {
        e.preventDefault(); //don't do anything
    }
});
```

edited Jun 13 '11 at 13:12

answered Jun 13 '11 at 12:41
John Strickler
23.4k ● 4 ● 47 ● 67

Thanks a lot. I tried to add the if(clicks==1) but it didnt can u please add the the single click function. Again thanks :) – kritya Jun 13 '11 at 12:55

@kritya I edited my answer - this should work better for you. See fiddle here - jsfiddle.net/VfJU4/1 – John Strickler Jun 13 '11 at 13:13

OOOOh i got where i was wrong i didnt put that ',' being addicted to dreamweaver it didnt highlighted it :P wtv Thanks :) – kritya Jun 13 '11 at 13:14

9

You could probably write your own custom implementation of click/dblclick to have it wait for an extra click. I don't see anything in the core jQuery functions that would help you achieve this.

Quote from .dblclick() at the jQuery site

*It is inadvisable to bind handlers to both the click and dblclick events for the same element. The sequence of events triggered varies from browser to browser, with some receiving two click events before the dblclick and others only one. Double-click sensitivity (maximum time between clicks that is detected as a double click) can vary by operating system and browser, and is often user-configurable.*

answered Jun 13 '11 at 12:37

Mick Hansen
**2,495** ● 15 ● 14

4

Look at the following code

```
$("#clickMe").click(function (e) {
    var $this = $(this);
    if ($this.hasClass('clicked')){
        alert("Double click");
        //here is your code for double click
        return;
    }else{
         $this.addClass('clicked');
        //your code for single click
         setTimeout(function() {
                 $this.removeClass('clicked'); },500);
    }//end of else
});
```

Demo goes here http://jsfiddle.net/cB484/

answered Apr 23 '14 at 9:16

Muhammad Tahir
**1,796** ● 20 ● 23

4 i like your idea of adding class and using it as a count. – nikhil Apr 30 '14 at 19:48

if you add Alert(); instead of //your code for single click it is not working at all – DmitryBoyko Jun 10 '14 at 14:14

4

I've written a jQuery plugin that allow also to delegate the click and dblclick events

```
// jQuery plugin to bind both single and double click to objects
// parameter 'delegateSelector' is optional and allow to delegate the events
// parameter 'dblclickWait' is optional default is 300
(function($) {
$.fn.multipleClicks = function(delegateSelector, clickFun, dblclickFun, dblclickWait) {
    var obj;
    if (typeof(delegateSelector)==='function' && typeof(clickFun)==='function') {
        dblclickWait = dblclickFun; dblclickFun = clickFun; clickFun =
delegateSelector; delegateSelector = null; // If 'delegateSelector' is missing reorder
arguments
    } else if (!(typeof(delegateSelector)==='string' && typeof(clickFun)==='function'
&& typeof(dblclickFun)==='function')) {
        return false;
    }
    return $(this).each(function() {
        $(this).on('click', delegateSelector, function(event) {
            var self = this;
            clicks = ($(self).data('clicks') || 0)+1;
            $(self).data('clicks', clicks);
            if (clicks == 1) {
                setTimeout(function(){
                    if ($(self).data('clicks') == 1) {
                        clickFun.call(self, event); // Single click action
                    } else {
                        dblclickFun.call(self, event); // Double click action
                    }
                    $(self).data('clicks', 0);
                }, dblclickWait || 300);
            }
        });
    });
};
})(jQuery);
```

answered Sep 27 '14 at 11:53

IlNidoDelChiurlo

41 ● 1

This plugin is great! But somehow it doesn't seem to work in IE9? I don't care for older versions (and you shouldn't anymore nowadays IMO) but I try to achieve functionality at least on IE9 - this is the browser anyone can have (no OS limitations) and which has good JS support. – Dennis98 Jul 4 '15 at 14:37

Usage: `$("body").multipleClicks('#mySelector', function(){ /* do something on click */},function(){/* do something on doubleclick */},300);` – Hafenkranich Oct 20 '16 at 22:56

This one actually enables you to *use* both – click and doubleclick– at the same time for different actions. Thank you! That one works like a charm. – Hafenkranich Oct 20 '16 at 22:57

3

```
var singleClickTimer = 0; //define a var to hold timer event in parent scope
jqueryElem.click(function(e){ //using jquery click handler
    if (e.detail == 1) { //ensure this is the first click
        singleClickTimer = setTimeout(function(){ //create a timer
            alert('single'); //run your single click code
        },250); //250 or 1/4th second is about right
    }
});

jqueryElem.dblclick(function(e){ //using jquery dblclick handler
    clearTimeout(singleClickTimer); //cancel the single click
    alert('double'); //run your double click code
});
```

answered Nov 6 '11 at 0:44

Gregory Ray

157 ● 1 ● 3

Simple and effective. +1 – Bruce Pierson Sep 12 '17 at 22:08

3

**i am implementing this simple solution , [http://jsfiddle.net/533135/VHkLR/5/](http://jsfiddle.net/533135/VHkLR/5/)**

*html code*

```
<p>Click on this paragraph.</p>
<b> </b>
```

*script code*

```
var dbclick=false;
$("p").click(function(){
setTimeout(function(){
if(dbclick ==false){
```

```
$("b").html("clicked")
}

},200)

}).dblclick(function(){
dbclick = true
$("b").html("dbclicked")
setTimeout(function(){
dbclick = false


},300)
});
```

**its not much laggy**

edited Nov 28 '12 at 16:07

answered Nov 28 '12 at 15:44
Chetan Sandeep Renu
163 ● 3 ● 10

2

This solution works for me

```
var DELAY = 250, clicks = 0, timer = null;

$(".fc-event").click(function(e) {
    if (timer == null) {
        timer = setTimeout(function() {
           clicks = 0;
            timer = null;
            // single click code
        }, DELAY);
    }

    if(clicks === 1) {
         clearTimeout(timer);
         timer = null;
         clicks = -1;
         // double click code
    }
    clicks++;
});
```

answered Feb 6 '12 at 6:10
Vitali Korol
21 ● 2

0

```
(function($){

$.click2 = function (elm, o){
    this.ao = o;
    var DELAY = 700, clicks = 0;
    var timer = null;
    var self = this;

    $(elm).on('click', function(e){
        clicks++;
        if(clicks === 1){
            timer = setTimeout(function(){
                self.ao.click(e);
            }, DELAY);
        } else {
            clearTimeout(timer);
            self.ao.dblclick(e);
        }
    }).on('dblclick', function(e){
        e.preventDefault();
    });

};

$.click2.defaults = { click: function(e){}, dblclick: function(e){} };

$.fn.click2 = function(o){
    o = $.extend({},$.click2.defaults, o);
    this.each(function(){ new $.click2(this, o); });
    return this;
};

})(jQuery);
```

And finally we use as.

```
$("a").click2({
    click : function(e){
        var cid = $(this).data('cid');
        console.log("Click : "+cid);
    },
    dblclick : function(e){
        var cid = $(this).data('cid');
        console.log("Double Click : "+cid);
    }
});
```

answered Jul 19 '13 at 5:03

Sergio FG
9 ● 1

What is the difference between $.click and $.fn.click? – FrenkyB Apr 3 '17 at 5:31

0

I made some changes to the above answers here which still works great: http://jsfiddle.net/arondraper/R8cDR/

answered Oct 10 '12 at 14:29

azzy81
2,150 ● 24 ● 37

0

This is a method you can do using the basic JavaScript, which is works for me:

```
var v_Result;
function OneClick() {
    v_Result = false;
    window.setTimeout(OneClick_Nei, 500)
```

```
        function OneClick_Nei() {
            if (v_Result != false) return;
            alert("single click");
        }
    }
    function TwoClick() {
        v_Result = true;
        alert("double click");
    }
```

edited Jun 27 '14 at 3:32
Michael Yaworski
12.1k ● 16 ● 55 ● 90

answered Jun 27 '14 at 3:04
user3781586
1

---

0

Below is my simple approach to the issue.

JQuery function:

```
jQuery.fn.trackClicks = function () {
    if ($(this).attr("data-clicks") === undefined) $(this).attr("data-clicks", 0);

    var timer;
    $(this).click(function () {
        $(this).attr("data-clicks", parseInt($(this).attr("data-clicks")) + 1);

        if (timer) clearTimeout(timer);

        var item = $(this);
        timer = setTimeout(function() {
            item.attr("data-clicks", 0);
        }, 1000);
    });
}
```

Implementation:

```
$(function () {
    $("a").trackClicks();

    $("a").click(function () {
        if ($(this).attr("data-clicks") === "2") {
            // Double clicked
        }
    });
});
```

Inspect the clicked element in Firefox/Chrome to see data-clicks go up and down as you click, adjust time (1000) to suit.

answered Oct 9 '15 at 13:33
mhapps
624 ● 5 ● 10

---

Same as the above answer but allows for triple click. (Delay 500) [http://jsfiddle.net/luenwarneke/rV78Y/1/](http://jsfiddle.net/luenwarneke/rV78Y/1/)

0

```
    var DELAY = 500,
    clicks = 0,
    timer = null;

$(document).ready(function() {
    $("a")
    .on("click", function(e){
        clicks++;  //count clicks
        timer = setTimeout(function() {
        if(clicks === 1) {
           alert('Single Click'); //perform single-click action
        } else if(clicks === 2) {
           alert('Double Click'); //perform single-click action
        } else if(clicks >= 3) {
           alert('Triple Click'); //perform Triple-click action
        }
         clearTimeout(timer);
         clicks = 0;  //after action performed, reset counter
       }, DELAY);
    })
    .on("dblclick", function(e){
        e.preventDefault();  //cancel system double-click event
    });
});
```

answered Mar 12 '14 at 2:16

Luen

3 ● 5

**Highly active question**. Earn 10 reputation in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.