

Wx360exe Win7 X64

Download

xbox 360 game download for free c/- SUBSCRIBE, YOU WILL RECEIVE!!! Wx360.exe Win7 X64 India Money Today 2017. apk jio4g low cost best sim free mobile apk Verbatim s-4c internal direct data lightgun power supply serial number Microsoft Office 2013 (x86 X64) V14.0.6298.1000 Download Wx360exe Win7 X64.Q: Toggling an element with a class I have an element in a page that is toggled by :target, but I want to control it with another element that has the class of.toggle (not all pages have it, only the one I'm working on now). When the element with the class of.toggle is activated by it's :target, I want to remove the :target (because of some code that doesn't work when it's active). Here's the code I've been using for this: `$('body').on('click', '.toggle', function() { $('.toggle', $(this).attr('href')).toggleClass('active'); $('.toggle').toggleClass('active'); $(this).toggleClass('active'); window.location.href =`

`$(this).attr('href'); });` The problem with this is that it removes the `:target` class from the element, but doesn't add the `:target` class to the element. Here's a live version of the code on jsFiddle: A: You can use `toggleClass()` again to add `:target`

```
$('body').on('click', '.toggle', function() { $('.toggle', $(this).attr('href')).toggleClass('active');  
$('.toggle').toggleClass('active');  
$(this).toggleClass('active'); window.location.href = $(this).attr('href');  
$('.toggle').add(':target').toggleClass('active'); });  
.active.active:target { color: red; }
```