* Overview
* Arguments

jQuery UI Position Utility

## Overview

Utility script for positioning any widget relative to the window, document, a particular element, or the cursor/mouse.

*Note: jQuery UI does not support positioning hidden elements.*

Does not need ui.core.js or effects.core.js.

### Dependencies

* *none (only jQuery core)*

### Example

* Demo
* View Source

Clicking on the green element transfers to the other.

$("#position1").position({  
 my: "center",  
 at: "center",  
 of: "#targetElement"  
});  
$("#position2").position({  
 my: "left top",  
 at: "left top",  
 of: "#targetElement"  
});  
$("#position3").position({  
 my: "right center",  
 at: "right bottom",  
 of: "#targetElement"  
});  
$(document).mousemove(function(ev){  
 $("#position4").position({  
 my: "left bottom",  
 of: ev,  
 offset: "3 -3",  
 collision: "fit"  
 });  
});

<!DOCTYPE html>  
<html>  
<head>  
 <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="stylesheet" type="text/css"/>  
 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>  
 <script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"></script>  
 <style type="text/css">  
#targetElement { width:240px;height:200px;background-color:#999;margin:30px auto; }  
.positionDiv { width:50px;height:50px;opacity:0.6; }  
#position1 {background-color:#F00;}  
#position2 {background-color:#0F0;}  
#position3 {background-color:#00F;}  
#position4 {background-color:#FF0;}  
</style>  
  
 <script>  
 $(document).ready(function() {  
   
$("#position1").position({  
 my: "center",  
 at: "center",  
 of: "#targetElement"  
});  
$("#position2").position({  
 my: "left top",  
 at: "left top",  
 of: "#targetElement"  
});  
$("#position3").position({  
 my: "right center",  
 at: "right bottom",  
 of: "#targetElement"  
});  
$(document).mousemove(function(ev){  
 $("#position4").position({  
 my: "left bottom",  
 of: ev,  
 offset: "3 -3",  
 collision: "fit"  
 });  
});  
  
 });  
 </script>  
</head>  
<body style="font-size:62.5%;">  
   
<div id="targetElement">  
 <div class="positionDiv" id="position1"></div>  
 <div class="positionDiv" id="position2"></div>  
 <div class="positionDiv" id="position3"></div>  
 <div class="positionDiv" id="position4"></div>  
</div>  
  
</body>  
</html>

## Arguments

### **my**Type: String Defines which position on **the element being positioned** to align with the target element: "horizontal vertical" alignment. A single value such as "right" will default to "right center", "top" will default to "center top" (following CSS convention). Acceptable values: "top", "center", "bottom", "left", "right". Example: "left top" or "center center"

### **at**Type: String Defines which position on **the target element** to align the positioned element against: "horizontal vertical" alignment. A single value such as "right" will default to "right center", "top" will default to "center top" (following CSS convention). Acceptable values: "top", "center", "bottom", "left", "right". Example: "left top" or "center center"

### **of**Type: Selector, Element, jQuery, Event Element to position against. If you provide a selector, the first matching element will be used. If you provide a jQuery object, the first element will be used. If you provide an event object, the pageX and pageY properties will be used. Example: "#top-menu"

### **offset**Type: String Add these left-top values to the calculated position, eg. "50 50" (left top) A single value such as "50" will apply to both.

### **collision**Type: String

* When the positioned element overflows the window in some direction, move it to an alternative position. Similar to my and at, this accepts a single value or a pair for horizontal/vertical, eg. "flip", "fit", "fit flip", "fit none".
  + **flip**: to the opposite side and the collision detection is run again to see if it will fit. If it won't fit in either position, the center option should be used as a fall back.
  + **fit**: so the element keeps in the desired direction, but is re-positioned so it fits.
  + **none**: not do collision detection.

### **using**Type: Function When specified the actual property setting is delegated to this callback. Receives a single parameter which is a hash of top and left values for the position that should be set.