Usage Documentation

For Version 1.0

By Vladimir Kharlampidi
@nolimits4web
The iDangero.us
http://www.idangero.us/

iDangero.us S6 Slider
@idangerous
http://www.idangero.us/sliders/s6/

Thank you for purchasing the S6 Slider! If you have any questions that are beyond the scope of this help file, please please feel free to contact us via our support ticket form [here](#).

Table of Contents

1. S6 Integration

1.1. Slider Integration (HTML and CSS)

It is very easy to integrate S6 into your page or mobile/web app. So let's first of all look at the HTML code we need to use:

First of all you need to attach the jQuery (or Zepto) library and S6's scripts and styles to your document. Add the following code to the HEAD section or in the end of the BODY section:

```
<head>
......
<link rel="stylesheet" src="path_to/s6/idangerous.s6-1.0.css" />
<script src="path_to/jquery.js"></script>
<script src="path_to/s6/idangerous.s6-1.0.min.js"></script>
......
</head>
```

S6 is intended to use with 4 slides, so insert the following code inside of BODY:

```
<div class="s6-container">
  <div class="s6-wrapper">

    <div class="s6-slide">
      <!-- Any HTML of the First slide -->
    </div>

    <div class="s6-slide">
      <!-- Any HTML of the Second slide -->
    </div>

    <div class="s6-slide">
      <!-- Any HTML of the Third slide -->
    </div>

    <div class="s6-slide">
      <!-- Any HTML of the Fourth slide -->
    </div>

  </div>
</div>
```

If you want to use pagination you need to add somewhere the following div:

```
<div class="my-pagination"></div>
```

At the example above we used slider with three images. As you can see the code is pretty simple.

Classes:

- In the example above we use the following classes: 's6-container', 's6-wrapper', 's6-slide', 'my-pagination'. These are default S6's classes, but if you need you can change them on slider initialization via options.

Pagination:

- in the example above we add "my-pagination" div for pagination. S6 will automatically add there 4 elements for slides and full HTML will be:

```
<div class="my-pagination">
  <span class="s6-pagination-switch s6-active-switch"></span>
  <span class="s6-pagination-switch"></span>
  <span class="s6-pagination-switch"></span>
  <span class="s6-pagination-switch"></span>
</div>
```

Every pagination element will have an "s6-pagination-switch" class, "active" pagination will have the "s6-active-switch" class. These are default class names for S6, but if you need you can change them on slider initialization via options.

Don't create full HTML of pagination, S6 will add them automatically, you just need to know their names to style them via CSS.

Now, let's look at the CSS code

After you add required HTML, you need to set slider's size, open idangerous.s6-1.0.css, you will need to change the following rule:

```
/* Set Dimensions */
.s6-container, .s6-slide {
      width:700px;
      height:300px;
}
```

Change the width and height of your slider in this rule. If you want it to be responsive, you can set dimensions in percents (%).

That is all about required HTML and CSS code. As you can see it's very simple

1.2. Slider Initialization (Java Script)

We need to initialize the S6 after the document is ready (is loaded). Use the following formatting inside of <script> tag (if you use it inside of your document file):

```
$(document).ready(function(){
      var myS6 = $(".s6-container").s6(parameters)
})
```

As you can see it was pretty simple! Now let's look at the list of available parameters:

Parameter	Type of variable	Example	Required	Default	Description
speed	*number*	`800`	optional	`500`	duration of animation between slides (in ms)
autoPlay	*number*	`5000`	optional	`-`	delay between transitions (in ms). If this parameter is not specified, auto play will be disabled
mode	*string*	`vertical'`	optional	`'horizontal'`	Rotation will be horizontal in "horizontal' mode and 'vertical' in 'vertical' mode
freeMode	*boolean*	`true`	optional	`false`	If 'true' then slider will not have fixed rotate positions
freeModeFluid	*boolean*	`true`	optional	`false`	If true, then if you release the slider it will keep rotating for a while
simulateTouch	*boolean*	`false`	optional	`true`	If true, S6 will accept mouse events like touch events (click and drag to rotate the slider)
onlyExternal	*boolean*	`true`	optional	`false`	If true, then the only way to switch the slide is use of external API functions like swipeRight or swipeLeft.

Parameter	Type of variable	Example	Required	Default	Description
followFinger	*boolean*	`true`	optional	`false`	If false, then slider will be animated only when you release it, it will not rotate while you hold your finger on it
Perspective Properties					
perspective	*number*	`800`	optional	`1000`	Perspective of slider
setPerspective	*boolean*	`false`	optional	`true`	If 'false' S6 will not set 'perspective' property to slider. Use it if you want to use S6 in custom perspective scene
createBackFaces	*boolean*	`false`	optional	`true`	If true, S6 will create additional 2 faces for more realistic appearance.
Pagination					
pagination	*string*	`".my-pagination"`	optional	`-`	Selector of the container where S6 will create (and track) pagination elements

Parameter	Type of variable	Example	Required	Default	Description
createPagination	*boolean*	`false`	optional	`true`	If "true" S6 will create pagination spans inside of container, specified in previous option. If "false" you will need to create appropriate elements manually.
Shadow					
dynamicShadow	*boolean*	`true`	optional	`false`	Set to 'true' if you want to use dynamic shadows
dynamicShadowOffset	*number*	`40`	optional	`20`	Vertical offset of shadow in px
dynamicShadowScale	*number*	`0.7`	optional	`0.9`	Scale factor of dynamic shadow
Call Back Functions					
onTouchStart	*function*	`function(){alert("Hello")}`	optional	-	function will be executed when you touch the slider
onTouchMove	*function*	`function(){alert("Hello")}`	optional	-	function will be executed when you touch and move the finger over the slider
onTouchEnd	*function*	`function(){alert("Hello")}`	optional	-	function will be executed when you release the slider

Parameter	Type of variable	Example	Required	Default	Description
onSlideChangeStart	*function*	`function(){alert("Hello")}`	optional	–	function will be executed when you release the slider and it going to change the slide. **Don't work with freeMode**
onSlideChangeEnd	*function*	`function(){alert("Hello")}`	optional	–	function will be executed after the animation to other slide (next or previous). **Don't work with freeMode**
onSlideReset	*function*	`function(){alert("Hello")}`	optional	–	function will be executed when you release the slider and it going to be reseted to currently active slide. **Don't work with freeMode**
Classes					
wrapperClass	*string*	`'my-wrapper'`	optional	`'s6-wrapper'`	CSS class of the S6's wrapper.
slideClass	*string*	`'my-slide'`	optional	`'s6-slide'`	CSS class of the S6's slide.
paginationClass	*string*	`'my-switch'`	optional	`'s6-pagination-switch'`	CSS class of the S6's pagination element.
paginationActiveClass	*string*	`'my-active-switch'`	optional	`'s6-active-switch'`	CSS class of the S6's active pagination element.

-

Example with parameters

Ok, let's look at the Slider initialization example again but with parameters:

```
$(document).ready(function(){
      var myS6 = $(".s6-container").s6({
            //Speed
            speed:300,

            //Mode
            mode : 'horizontal',
            freeMode: true,
            freeModeFluid : true,

            // Perspective
            perspective:900,
            setPerspective : true,
            createBackFaces : true,

            //Pagination
            pagination : '.my-pagination',

            //Dynamic Shadows
            dynamicShadow : true,
            dynamicShadowOffset : 30,
            dynamicShadowScale: 0.8,

            //Callbacks
            onTouchStart: function(){ alert('Oops, you touch me') },
            onTouchEnd: function(){ alert('Phuf') }
      })
})
```

That was example with almost all possible parameters. Of course there is no sense to specify parameters that you do not need.

2. S6 API

After you init the slider, it returns special S6 object which is contain a lot of useful parameters and methods. In the example above we used:

```
$(document).ready(function(){
      var myS6 = $(".s6-container").s6(parameters)
})
```

And now, "myS6" variable contains "S6 object". Here its structure:

2.1 s6.activeSlide

Contains the index of currently active (front) slide.

Usage example:

```
alert ( myS6.activeSlide )
```

2.2 s6.height / s6.width

Contains width and height of slider.

Usage example:

```
alert ( "S6 dimensions is "+myS6.width+" x "+myS6.height+"px" )
```

2.3 s6.isSupport3D

Returns "true" if browser supports CSS3 3D transforms.

Usage example:

```
if (myS6.isSupport3D) {
  alert ( "Your browser supports CSS3 3D transforms" )
}
```

2.4 s6.isSupportTouch()

Function. Returns "true" if browser supports touch events.

Usage example:

```
if ( myS6.isSupportTouch() ) {
  alert ( "Your browser supports touch events" )
}
```

2.5 s6.startAutoPlay()

Function. Will enable auto play, if auto play delay parameter was specified. Can be used to create external play/pause buttons.

Usage example:

```
$(".play-button").click(function(){
  myS6.startAutoPlay()
})
```

2.6 s6.stopAutoPlay()

Function. Will disable auto play, if auto play delay parameter was specified. Can be used to create external play/pause buttons.

Usage example:

```
$(".stop-button").click(function(){
  myS6.stopAutoPlay()
})
```

2.7 s6.swipeNext()

Function. Run transition to next slide. Can be used to create external previous/next buttons.

Usage example:

```
$(".next-button").click(function(){
  myS6.swipeNext()
})
```

2.8 s6.swipePrev()

Function. Run transition to previous slide. Can be used to create external previous/next buttons.

Usage example:

```
$(".prev-button").click(function(){
  myS6.swipePrev()
})
```

2.9 s6.touches

Object. Contain information about touches:

- **myS6.touches.startX** - number, X-position (left) of the first touch - on touch start
- **myS6.touches.startY** - number, Y-position (top) of the first touch - on touch start
- **myS6.touches.currentX** - number, X-position(left) of the touch (finger) while you moving it over the slider (ontouchmove) , and equal to final X-position when you release the slider (ontouchend)
- **myS6.touches.currentY** - number, Y-position(top) of the touch (finger) while you moving it over the slider (ontouchmove) , and equal to final Y - position when you release the slider (ontouchend)
- **myS6.touches.diffX** - equal to (myS6.touches.currentX - myS6.touches.startX)
- **myS6.touches.diffY** - equal to (myS6.touches.currentY - myS6.touches.startY)

These parameters may be useful with callback functions.

Usage example:

```
var myS6 = $('.s6-container').s6({
      onTouchEnd : function(){
            alert('You moved finger on ' + myS6.touches.diffX + ' pixels over the
slider' )
      }
})
```

2.10 s6.rotates

Object. Contain information about rotate angles of slider:

- **myS6.rotates.startX** - X-angle (in degrees) of slider when you start touch it, changing only on touch start event. Used in 'vertical' mode

- **myS6. rotates.startY** - Y-angle (in degrees) of slider when you start touch it, changing only on touch start event. Used in 'horizontal' mode
- **myS6. rotates.currentX** - number, X-angle (in degrees) of slider while you rotate (touch) it over with finger (ontouchmove) , and equal to final X-angle when you release the slider (ontouchend)
- **myS6. rotates.currentY** - number, Y-angle (in degrees) of slider while you rotate (touch) it over with finger (ontouchmove) , and equal to final Y-angle when you release the slider (ontouchend)

These parameters may be useful with callback functions.

Usage example:

```
var myS6 = $('.s6-container').s6({
      onSlideChangeEnd : function(){
            alert('Slider was rotate to new angle equal to '+myS6.rotates.currentY
+' degrees')
      }
})
```

2.11 s6.params

Object. Contain information about parameters of slider:

Usage example:

```
var myS6 = $('.s6-container').s6();
alert ('Slider animation speed is ' + myS6.params.speed)
```

Many of parametric options can be overwritten after slider's initialization, for example:

```
var myS6 = $('.s6-container').s6();

//Clicking on the element with 'change-speed' class, will change slider's speed to
1000ms
$(".change-speed").click(function(){
      myS6.params.speed = 1000;
})

//Enable freeMode by clicking something with "enable-free-mode" class:
$(".enable-free-mode").click(function(){
      myS6.params.freeMode = true;
})
```

3. Fallback and Degradation

S6 comes with amazing feature if you want to use it on many devices (browsers) which may not support 3D Transforms. In this case you need also to include Swiper (it's free) and if browser don't support 3D Transforms, S6 will detect it and switch the slider to Swiper automatically for graceful degradation!

Let's back to S6 integration, but now with Swiper:

```
<head>
......
<link rel="stylesheet" src="path_to/s6/idangerous.s6-1.0.css" />
<script src="path_to/jquery.js"></script>

<!-- Put Swiper before the S6 script for graceful degradation -->
<script src="path_to/idangerous.swiper-1.1.min.js"></script>

<script src="path_to/s6/idangerous.s6-1.0.min.js"></script>
......
</head>
```

That is all, S6 will do the rest part of work for you.

You can download Swiper (it's free) here: **http://www.idangero.us/sliders/swiper/**