

Swiper

Most Modern Mobile Touch Slider

[Get Started](#)[API](#)[Demos](#)[Forum](#)

Like 248

Tweet 376

Follow @idangerous

783 followers

g+1

420

9.6K



Star 5,223

Fork 2,067

Swiper API

Swiper Full HTML Layout

```
<!-- Slider main container -->
<div class="swiper-container">
    <!-- Additional required wrapper -->
    <div class="swiper-wrapper">
        <!-- Slides -->
        <div class="swiper-slide">Slide 1</div>
        <div class="swiper-slide">Slide 2</div>
        <div class="swiper-slide">Slide 3</div>
        ...
    </div>
    <!-- If we need pagination -->
    <div class="swiper-pagination"></div>

    <!-- If we need navigation buttons -->
    <div class="swiper-button-prev"></div>
    <div class="swiper-button-next"></div>

    <!-- If we need scrollbar -->
    <div class="swiper-scrollbar"></div>
</div>
```

Initialize Swiper

Now, when we have Swiper's HTML, we need to initialize it using the following function:

`new Swiper(swiperContainer, parameters)` - initialize swiper with options

- `swiperContainer` - `HTMLElement` or `string` (with CSS Selector) of swiper container HTML element. Required.
- `parameters` - `object` - object with Swiper parameters. Optional.

Method returns initialized Swiper instance

For example:

```
var mySwiper = new Swiper('.swiper-container', {
    speed: 400,
    spaceBetween: 100
});
```

Swiper Parameters

Let's look on list of all available parameters:

Parameter	Type	Default	Description
initialSlide	number	0	Index number of initial slide.
direction	string	'horizontal'	Could be 'horizontal' or 'vertical' (for vertical slider).
speed	number	300	Duration of transition between slides (in ms)
setWrapperSize	boolean	false	Enabled this option and plugin will set width/height on swiper wrapper equal to total size of all slides. Mostly should be used as compatibility fallback option for browser that don't support flexbox layout well
virtualTranslate	boolean	false	Enabled this option and swiper will be operated as usual except it will not move, real translate values on wrapper will not be set. Useful when you may need to create custom slide transition
width	number		Swiper width (in px). Parameter allows to force Swiper width. Useful only if you initialize Swiper when it is hidden. Setting this parameter will make Swiper not responsive
height	number		Swiper height (in px). Parameter allows to force Swiper height. Useful only if you initialize Swiper when it is hidden. Setting this parameter will make Swiper not responsive
Autoplay			
autoplay	number	-	delay between transitions (in ms). If this parameter is not specified, auto play will be disabled
autoplayDisableOnInteraction	boolean	true	Set to false and autoplay will not be disabled after user interactions (swipes), it will be restarted every time after interaction
Progress			
watchSlidesProgress	boolean	false	Enable this feature to calculate each slides progress
watchSlidesVisibility	boolean	false	<code>watchSlidesProgress</code> should be enabled. Enable this option and slides that are in viewport will have additional visible class
Freemode			
freeMode	boolean	false	If true then slides will not have fixed positions
freeModeMomentum	boolean	true	If true , then slide will keep moving for a while after you release it
freeModeMomentumRatio	number	1	Higher value produces larger momentum distance after you release slider
freeModeMomentumBounce	boolean	true	Set to false if you want to disable momentum bounce in free mode
freeModeMomentumBounceRatio	number	1	Higher value produces larger momentum bounce effect
freeModeSticky	boolean	false	Set to true to enable snap to slides positions in free mode
Effects			
effect	string	'slide'	Could be "slide", "fade", "cube" or "coverflow"
fade	object	<pre> fade: { crossFade: false } </pre>	Fade effect parameters
cube	object	<pre> cube: { slideShadows: true, } </pre>	Cube effect parameters. For better performance you may disable shadows

		<pre>shadow: true, shadowOffset: 20, shadowScale: 0.94 }</pre>	
overflow	object	<pre>overflow: { rotate: 50, stretch: 0, depth: 100, modifier: 1, slideShadows: true }</pre>	Overflow effect parameters. For better performance you may disable shadows
Parallax			
parallax	boolean	false	Enable, if you want to use "parallaxed" elements inside of slider
Slides grid			
spaceBetween	number	0	Distance between slides in px.
slidesPerView	number or 'auto'	1	<p>Number of slides per view (slides visible at the same time on slider's container).</p> <p>If you use it with "auto" value and along with loop: true then you need to specify loopedSlides parameter with amount of slides to loop (duplicate)</p> <p>slidesPerView: 'auto' is currently not compatible with multirow mode, when slidesPerColumn > 1</p>
slidesPerColumn	number	1	Number of slides per column, for multirow layout
slidesPerColumnFill	string	'column'	Could be 'column' or 'row'. Defines how slides should fill rows, by column or by row
slidesPerGroup	number	1	Set numbers of slides to define and enable group sliding. Useful to use with slidesPerView > 1
centeredSlides	boolean	false	If true, then active slide will be centered, not always on the left side.
Grab Cursor			
grabCursor	boolean	false	This option may a little improve desktop usability. If true , user will see the "grab" cursor when hover on Swiper
Touches			
touchRatio	number	1	Touch ration
touchAngle	number	45	Allowable angle (in degrees) to trigger touch move
simulateTouch	boolean	true	If true, Swiper will accept mouse events like touch events (click and drag to change slides)
shortSwipes	boolean	true	Set to false if you want to disable short swipes
longSwipes	boolean	true	Set to false if you want to disable long swipes
longSwipesRatio	number	0.5	Ratio to trigger swipe to next/previous slide during long swipes
longSwipesMs	number	300	Minimal duration (in ms) to trigger swipe to next/previous slide during long swipes
followFinger	boolean	true	If disabled, then slider will be animated only when you release it, it will not move while you hold your finger on it

<code>onlyExternal</code>	<code>boolean</code>	<code>false</code>	If true, then the only way to switch the slide is use of external API functions like <code>slidePrev</code> or <code>slideNext</code>
<code>threshold</code>	<code>number</code>	<code>0</code>	Threshold value in px. If "touch distance" will be lower than this value then swiper will not move
<code>touchMoveStopPropagation</code>	<code>boolean</code>	<code>true</code>	If enabled, then propagation of "touchmove" will be stopped
Touch Resistance			
<code>resistance</code>	<code>boolean</code>	<code>true</code>	Set to <code>false</code> if you want to disable resistant bounds
<code>resistanceRatio</code>	<code>number</code>	<code>0.85</code>	This option allows you to control resistance ratio
Clicks			
<code>preventClicks</code>	<code>boolean</code>	<code>true</code>	Set to <code>true</code> to prevent accidental unwanted clicks on links during swiping
<code>preventClicksPropagation</code>	<code>boolean</code>	<code>true</code>	Set to <code>true</code> to stop clicks event propagation on links during swiping
<code>slideToClickedSlide</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> and click on any slide will produce transition to this slide
Swiping / No swiping			
<code>allowSwipeToPrev</code>	<code>boolean</code>	<code>true</code>	Set to <code>false</code> to disable swiping to previous slide direction (to left or top)
<code>allowSwipeToNext</code>	<code>boolean</code>	<code>true</code>	Set to <code>false</code> to disable swiping to next slide direction (to right or bottom)
<code>noSwiping</code>	<code>boolean</code>	<code>true</code>	Set to <code>false</code> to disable swiping to next slide direction (to right or bottom)
<code>noSwipingClass</code>	<code>string</code>	<code>'swiper-no-swiping'</code>	If <code>true</code> , then you can add <code>noSwipingClass</code> class to swiper's slide to prevent/disable swiping on this element
<code>swipeHandler</code>	<code>string / HTMLElement</code>	<code>null</code>	String with CSS selector or HTML element of the container with pagination that will work as only available handler for swiping
Pagination			
<code>pagination</code>	<code>string / HTMLElement</code>	<code>null</code>	String with CSS selector or HTML element of the container with pagination
<code>paginationHide</code>	<code>boolean</code>	<code>true</code>	Toggle (hide/true) pagination container visibility when click on Slider's container
<code>paginationClickable</code>	<code>boolean</code>	<code>false</code>	If <code>true</code> then clicking on pagination button will cause transition to appropriate slide
<code>paginationBulletRender(index, className)</code>	<code>function</code>	<code>null</code>	<p>This parameter allows totally customize pagination bullets, you need to pass here a function that accepts <code>index</code> number of pagination bullet and required element class name (<code>className</code>).</p> <p>For example, with this code, we can add slide number into pagination bullet:</p> <pre>var swiper = new Swiper('.swiper-container', { ... paginationBulletRender: function (index, className) { return '' + (index + 1) + ''; } });</pre>
Navigation Buttons			
<code>nextButton</code>	<code>string / HTMLElement</code>	<code>null</code>	String with CSS selector or HTML element of the element that will work like "next" button after click on it
<code>prevButton</code>	<code>string / HTMLElement</code>	<code>null</code>	String with CSS selector or HTML element of the element that will work like "prev" button after click on it
Accessibility			
<code>a11y</code>	<code>boolean</code>	<code>false</code>	Option to enable keyboard accessibility to provide focusable navigation buttons and basic ARIA for screen readers
<code>prevSlideMessage</code>	<code>string</code>	<code>'Previous slide'</code>	Message for screen readers for previous button
<code>nextSlideMessage</code>	<code>string</code>	<code>'Next slide'</code>	Message for screen readers for next button

<code>firstSlideMessage</code>	<code>string</code>	<code>'This is the first slide'</code>	Message for screen readers for previous button when swiper is on first slide
<code>lastSlideMessage</code>	<code>string</code>	<code>'This is the last slide'</code>	Message for screen readers for previous button when swiper is on last slide
Scrollbar			
<code>scrollbar</code>	<code>string / HTMLElement</code>	<code>null</code>	String with CSS selector or HTML element of the container with scrollbar.
<code>scrollbarHide</code>	<code>boolean</code>	<code>true</code>	Hide scrollbar automatically after user interaction
Keyboard / Mousewheel			
<code>keyboardControl</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to enable navigation through slides using keyboard right and left (for horizontal mode), top and bottom (for vertical mode) keyboard arrows
<code>mousewheelControl</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to enable navigation through slides using mouse wheel
<code>mousewheelForceToAxis</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to force mousewheel swipes to axis. So in horizontal mode mousewheel will work only with horizontal mousewheel scrolling, and only with vertical scrolling in vertical mode.
<code>mousewheelReleaseOnEdges</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> and swiper will release mousewheel event and allow page scrolling when swiper is on edge positions (in the beginning or in the end)
<code>mousewheelInvert</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to invert sliding direction
Hash Navigation			
<code>hashnav</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to enable hash url navigation to for slides
Images			
<code>preloadImages</code>	<code>boolean</code>	<code>true</code>	When enabled Swiper will force to load all images
<code>updateOnImagesReady</code>	<code>boolean</code>	<code>true</code>	When enabled Swiper will be reinitialized after all inner images (<code></code> tags) are loaded. Required <code>preloadImages: true</code>
<code>lazyLoading</code>	<code>boolean</code>	<code>false</code>	Set to "true" to enable images lazy loading. Note that <code>preloadImages</code> should be disabled
<code>lazyLoadingInPrevNext</code>	<code>boolean</code>	<code>false</code>	Set to "true" to enable lazy loading for the closest slides images (for previous and next slide images)
<code>lazyLoadingOnTransitionStart</code>	<code>boolean</code>	<code>false</code>	By default, Swiper will load lazy images after transition to this slide, so you may enable this parameter if you need it to start loading of new image in the beginning of transition
Loop			
<code>loop</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to enable continuous loop mode If you use it along with <code>slidesPerView: 'auto'</code> then you need to specify <code>loopedSlides</code> parameter with amount of slides to loop (duplicate)
<code>loopAdditionalSlides</code>	<code>number</code>	<code>0</code>	Addition number of slides that will be cloned after creating of loop
<code>loopedSlides</code>	<code>number</code>	<code>null</code>	If you use <code>slidesPerView: 'auto'</code> with loop mode you should tell to Swiper how many slides it should loop (duplicate) using this parameter
Controller			
<code>control</code>	<code>[Swiper Instance]</code>	<code>undefined</code>	Pass here another Swiper instance or array with Swiper instances that should be controlled by this Swiper
<code>controlInverse</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> and controlling will be in inverse direction
Observer			
<code>observer</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> to enable Mutation Observer on Swiper and its elements. In this case Swiper will be updated (reinitialized) each time if you change its style (like hide/show) or modify its child elements (like adding/removing slides)
<code>observeParents</code>	<code>boolean</code>	<code>false</code>	Set to <code>true</code> if you also need to watch Mutations for Swiper parent elements

Callbacks

<code>runCallbacksOnInit</code>	<code>boolean</code>	<code>true</code>	Run on [Transition/SlideChange][Start/End] callbacks on swiper initialization. Such callbacks will be fired on initialization in case of your initialSlide is not 0, or you use loop mode
<code>onInit(swiper)</code>	<code>function</code>		Callback function, will be executed right after Swiper initialization
<code>onSlideChangeStart(swiper)</code>	<code>function</code>		Callback function, will be executed in the beginning of animation to other slide (next or previous). Receives swiper instance as an argument.
<code>onSlideChangeEnd(swiper)</code>	<code>function</code>		Callback function, will be executed after animation to other slide (next or previous). Receives slider instance as an argument.
<code>onTransitionStart(swiper)</code>	<code>function</code>		Callback function, will be executed in the beginning of transition. Receives swiper instance as an argument.
<code>onTransitionEnd(swiper)</code>	<code>function</code>		Callback function, will be executed after transition. Receives slider instance as an argument.
<code>onTouchStart(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user touch Swiper. Receives swiper instance and 'touchstart' event as an arguments.
<code>onTouchMove(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user touch and move finger over Swiper. Receives swiper instance and 'touchmove' event as an arguments.
<code>onTouchMoveOpposite(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user touch and move finger over Swiper in direction opposite to <code>direction</code> parameter. Receives swiper instance and 'touchmove' event as an arguments.
<code>onSliderMove(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user touch and move finger over Swiper and move it. Receives swiper instance and 'touchmove' event as an arguments.
<code>onTouchEnd(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user release Swiper. Receives swiper instance and 'touchend' event as an arguments.
<code>onClick(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user click/tap on Swiper after 300ms delay. Receives swiper instance and 'touchend' event as an arguments.
<code>onTap(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user click/tap on Swiper. Receives swiper instance and 'touchend' event as an arguments.
<code>onDoubleTap(swiper, event)</code>	<code>function</code>		Callback function, will be executed when user double tap on Swiper's container. Receives swiper instance and 'touchend' event as an arguments
<code>onImagesReady(swiper)</code>	<code>function</code>		Callback function, will be executed right after all inner images are loaded. <code>updateOnImagesReady</code> should be also enabled
<code>onProgress(swiper, progress)</code>	<code>function</code>		Callback function, will be executed when Swiper progress is changed, as second arguments it receives <code>progress</code> that is always from 0 to 1
<code>onReachBeginning(swiper)</code>	<code>function</code>		Callback function, will be executed when Swiper reach its beginning (initial position)
<code>onReachEnd(swiper)</code>	<code>function</code>		Callback function, will be executed when Swiper reach last slide
<code>onDestroy(swiper)</code>	<code>function</code>		Callback function, will be executed when you destroy Swiper
<code>onSetTranslate(swiper, translate)</code>	<code>function</code>		Callback function, will be executed when swiper's wrapper change its position. Receives swiper instance and current translate value as an arguments
<code>onSetTransition(swiper, transition)</code>	<code>function</code>		Callback function, will be executed everytime when swiper starts animation. Receives swiper instance and current transition duration (in ms) as an arguments
<code>onAutoplayStart(swiper)</code>	<code>function</code>		Callback function, will be executed when when autoplay started
<code>onAutoplayStop(swiper)</code>	<code>function</code>		Callback function, will be executed when when autoplay stopped
<code>onLazyImageLoad(swiper, slide, image)</code>	<code>function</code>		Callback function, will be executed in the beginning of lazy loading of image
<code>onLazyImageReady(swiper, slide, image)</code>	<code>function</code>		Callback function, will be executed when lazy loading image will be loaded

Namespace

<code>slideClass</code>	<code>string</code>	<code>'swiper-slide'</code>	CSS class name of slide
<code>slideActiveClass</code>	<code>string</code>	<code>'swiper-slide-active'</code>	CSS class name of currently active slide
<code>slideVisibleClass</code>	<code>string</code>	<code>'swiper-slide-visible'</code>	CSS class name of currently visible slide
<code>slideDuplicateClass</code>	<code>string</code>	<code>'swiper-slide-duplicate'</code>	CSS class name of slide duplicated by loop mode
<code>slideNextClass</code>	<code>string</code>	<code>'swiper-slide-next'</code>	CSS class name of slide which is right after currently active slide
<code>slidePrevClass</code>	<code>string</code>	<code>'swiper-slide-prev'</code>	CSS class name of slide which is right before currently active slide
<code>wrapperClass</code>	<code>string</code>	<code>'swiper-wrapper'</code>	CSS class name of slides' wrapper
<code>bulletClass</code>	<code>string</code>	<code>'swiper-pagination-bullet'</code>	CSS class name of single pagination bullet
<code>bulletActiveClass</code>	<code>string</code>	<code>'swiper-pagination-bullet-active'</code>	CSS class name of currently active pagination bullet
<code>paginationHiddenClass</code>	<code>string</code>	<code>'swiper-pagination-hidden'</code>	CSS class name of pagination when it becomes inactive
<code>buttonDisabledClass</code>	<code>string</code>	<code>'swiper-button-disabled'</code>	CSS class name of next/prev button when it becomes disabled

Slider Methods & Properties

After we initialize Slider we have its initialized instance in variable (like `mySwiper` variable in example above) with helpful methods and properties:

Properties	
<code>mySwiper.params</code>	Object with passed initialization parameters
<code>mySwiper.container</code>	Dom7/jQuery element with slider container HTML element. To get vanilla HTMLElement use <code>mySwiper.container[0]</code>
<code>mySwiper.wrapper</code>	Dom7/jQuery element with slider wrapper HTML element. To get vanilla HTMLElement use <code>mySwiper.wrapper[0]</code>
<code>mySwiper.slides</code>	Dom7/jQuery array-like collection of slides HTML elements. To get specific slide HTMLElement use <code>mySwiper.slides[1]</code>
<code>mySwiper.bullets</code>	Dom7/jQuery collection of pagination buttons HTML elements. To get specific slide HTMLElement use <code>mySwiper.bullets[1]</code>
<code>mySwiper.width</code>	Width of container
<code>mySwiper.height</code>	Height of container
<code>mySwiper.translate</code>	Current value of wrapper translate
<code>mySwiper.progress</code>	Current progress of wrapper translate (from 0 to 1)
<code>mySwiper.activeIndex</code>	Index number of currently active slide Note, that in loop mode active index value will be always shifted on a number of looped/duplicated slides
<code>mySwiper.previousIndex</code>	Index number of previously active slide
<code>mySwiper.isBeginning</code>	<code>true</code> if slider on most "left"/"top" position
<code>mySwiper.isEnd</code>	<code>true</code> if slider on most "right"/"bottom" position

<code>mySwiper.autoplaying</code>	<code>true</code> if autoplay is enabled
<code>mySwiper.animating</code>	<code>true</code> if swiper is in transition
<code>mySwiper.touches</code>	Object with the following touch event properties: <ul style="list-style-type: none"> <code>mySwiper.touches.startX</code> <code>mySwiper.touches.startY</code> <code>mySwiper.touches.currentX</code> <code>mySwiper.touches.currentY</code> <code>mySwiper.touches.diff</code>
<code>mySwiper.clickedIndex</code>	Index number of last clicked slide
<code>mySwiper.clickedSlide</code>	Link to last clicked slide (<code>HTMLElement</code>)
Methods	
<code>mySwiper.slideNext(runCallbacks, speed);</code>	Run transition to next slide <ul style="list-style-type: none"> <code>runCallbacks</code> - <code>boolean</code> - Set it to <code>false</code> (by default it is <code>true</code>) and transition will not produce <code>onSlideChange</code> callback functions. <i>Optional</i> <code>speed</code> - <code>number</code> - transition duration (in ms). <i>Optional</i>
<code>mySwiper.slidePrev(runCallbacks, speed);</code>	Run transition to previous slide <ul style="list-style-type: none"> <code>runCallbacks</code> - <code>boolean</code> - Set it to <code>false</code> (by default it is <code>true</code>) and transition will not produce <code>onSlideChange</code> callback functions. <i>Optional</i> <code>speed</code> - <code>number</code> - transition duration (in ms). <i>Optional</i>
<code>mySwiper.slideTo(index, speed, runCallbacks);</code>	Run transition to the slide with index number equal to 'index' parameter for the duration equal to 'speed' parameter. <ul style="list-style-type: none"> <code>index</code> - <code>number</code> - index number of slide <code>speed</code> - <code>number</code> - transition duration (in ms). <i>Optional</i> <code>runCallbacks</code> - <code>boolean</code> - Set it to <code>false</code> (by default it is <code>true</code>) and transition will not produce <code>onSlideChange</code> callback functions. <i>Optional</i>
<code>mySwiper.update(updateTranslate);</code>	This method includes <code>updateContainerSize</code> , <code>updateSlidesSize</code> , <code>updateProgress</code> , <code>updatePagination</code> and <code>updateClasses</code> methods You should call it after you add/remove slides manually, or after you hide/show it, or do any custom DOM modifications with Swiper <ul style="list-style-type: none"> <code>updateTranslate</code> - <code>boolean</code> - Set it to <code>true</code> (by default it is <code>false</code>) to hard set/reset/update Swiper wrapper translate. It is useful if you use not default effect or scrollbar. <i>Optional</i> <p>This method also includes subcall of the following methods which you can use separately:</p> <ul style="list-style-type: none"> <code>mySwiper.updateContainerSize()</code> - recalculate size of swiper container <code>mySwiper.updateSlidesSize()</code> - recalculate number of slides and their offsets. Useful after you add/remove slides with JavaScript <code>mySwiper.updateProgress()</code> - recalculate swiper progress <code>mySwiper.updatePagination()</code> - updates pagination layout and re-render bullets <code>mySwiper.updateClasses()</code> - update active/prev/next classes on slides and bullets
<code>mySwiper.onResize();</code>	Swiper executes this method when you resize browser. It is almost the same as <code>.update()</code> but a bit softer, without hard setting translate
<code>mySwiper.detachEvents();</code>	Detach all events listeners
<code>mySwiper.attachEvents();</code>	Attach all events listeners again
<code>mySwiper.startAutoplay();</code>	start auto play. It may be useful for custom "Play" and "Pause" buttons
<code>mySwiper.stopAutoplay();</code>	stop auto play. It may be useful for custom "Play" and "Pause" buttons
<code>mySwiper.destroy(deleteInstance, cleanupStyles);</code>	Destroy slider instance and detach all events listeners, where

	<ul style="list-style-type: none"> • <code>deleteInstance</code> - <code>boolean</code> - Set it to <code>false</code> (by default it is <code>true</code>) to not to delete Swiper instance • <code>cleanupStyles</code> - <code>boolean</code> - Set it to <code>true</code> (by default it is <code>false</code>) and all custom styles will be removed from slides, wrapper and container. Useful if you need to destroy Swiper and to init again with new options or in different direction
<code>mySwiper.appendSlide(slides);</code>	Add new slides to the end. <code>slides</code> could be <code>HTMLElement</code> or <code>HTML string</code> with new slide or array with such slides, for example: <pre>mySwiper.appendSlide('<div class="swiper-slide">Slide 10</div>') mySwiper.appendSlide(['<div class="swiper-slide">Slide 10</div>', '<div class="swiper-slide">Slide 11</div>']);</pre>
<code>mySwiper.prependSlide(slides);</code>	Add new slides to the beginning. <code>slides</code> could be <code>HTMLElement</code> or <code>HTML string</code> with new slide or array with such slides, for example: <pre>mySwiper.prependSlide('<div class="swiper-slide">Slide 0</div>') mySwiper.prependSlide(['<div class="swiper-slide">Slide 1</div>', '<div class="swiper-slide">Slide 2</div>']);</pre>
<code>mySwiper.removeSlide(slideIndex);</code>	Remove selected slides. <code>slideIndex</code> could be a number with slide index to remove or array with indexes, for example: <pre>mySwiper.removeSlide(0); //remove first slide mySwiper.removeSlide([0, 1]); //remove first and second slides</pre>
<code>mySwiper.removeAllSlides();</code>	Remove all slides
<code>mySwiper.setWrapperTranslate(translate);</code>	Set custom css3 transform's translate value for swiper wrapper
<code>mySwiper.getWrapperTranslate();</code>	Get current value of swiper wrapper css3 transform translate
<code>mySwiper.on(callback, handler)</code>	Add callback/event handler
<code>mySwiper.once(callback, handler)</code>	Add event/callback that will be executed only once
<code>mySwiper.off(callback)</code>	Remove all handlers for specified callback/event
<code>mySwiper.lockSwipeToNext()</code> <code>mySwiper.unlockSwipeToNext()</code>	Disable (lock) / enable (unlock) ability to slide to the next slides
<code>mySwiper.lockSwipeToPrev()</code> <code>mySwiper.unlockSwipeToPrev()</code>	Disable (lock) / enable (unlock) ability to slide to the previous slides
<code>mySwiper.lockSwipes()</code> <code>mySwiper.unlockSwipes()</code>	Disable (lock) / enable (unlock) ability to change slides
<code>mySwiper.disableMousewheelControl();</code>	Disable mousewheel control
<code>mySwiper.enableMousewheelControl();</code>	Enable mousewheel control
<code>mySwiper.disableKeyboardControl();</code>	Disable keyboard control
<code>mySwiper.enableKeyboardControl();</code>	Enable keyboard control

Hash Navigation

Hash navigation is intended to have a link to specific slide that allows to load page with specific slide opened.

To make it work, you need to enable it by passing `hashnav:true` parameter and adding slides hashes in `data-hash` attribute:

```
<div class="swiper-container">
```

```
<div class="swiper-wrapper">
  <div class="swiper-slide" data-hash="slide1">Slide 1</div>
  <div class="swiper-slide" data-hash="slide2">Slide 2</div>
  <div class="swiper-slide" data-hash="slide3">Slide 3</div>
  <div class="swiper-slide" data-hash="slide4">Slide 4</div>
  <div class="swiper-slide" data-hash="slide5">Slide 5</div>
  ...
</div>
</div>
```

```
var swiper = new Swiper('.swiper-container', {
  //enable hash navigation
  hashnav: true
})
```

Parallax

Since version 3.0.3 Swiper supports parallax transition effects for swiper/slides nested elements. There are two types of parallax elements supported:

- Direct child elements of `swiper-container`. Parallax effect for such elements will depend on total slider progress. Useful for parallax backgrounds
- Slides child elements. Parallax effect for such elements will depend on slide progress

To enable parallax effects you need to init Swiper with passed `parallax:true` parameter and add `data-swiper-parallax` attribute on required elements

```
<div class="swiper-container">
  <!-- Parallax background element -->
  <div
    class="parallax-bg"
    style="background-image:url(http://lorempixel.com/900/600/nightlife/2/)"
    data-swiper-parallax="-23%">
  </div>
  <div class="swiper-wrapper">
    <div class="swiper-slide">
      <!-- Each slide has parallax title -->
      <div class="title" data-swiper-parallax="-100">Slide 1</div>
      <!-- Parallax subtitle -->
      <div class="subtitle" data-swiper-parallax="-200">Subtitle</div>
      <!-- And parallax text with custom transition duration -->
      <div class="text" data-swiper-parallax="-300" data-swiper-parallax-duration="600">
        <p>Lorem ipsum dolor sit amet, ...</p>
      </div>
    </div>
    ...
  </div>
</div>
```

All elements with specified `data-swiper-parallax` attribute will have parallax transition. This attribute may accept:

- **number** - value in px (as for title, subtitle in example above) to move element depending on progress. In this case such element will be moved on \pm this value in px depending on slide position (next or previous)
- **percentage** - (as for "parallax-bg") to move element depending on progress and on its size. In this case such element will be moved on \pm this percentage of its size (width in horizontal direction, and height in vertical direction) depending on slide position (next or previous). So if element has 400px width and you specified `data-swiper-parallax="50%"` then it will be moved on $\pm 200px$

It is possible to overwrite parallax direction by using `data-swiper-parallax-x` and `data-swiper-parallax-y` attributes with same rules

You can also pass custom transition duration for parallax elements by adding `data-swiper-parallax-duration` attribute.

Lazy Loading

Since version 3.0.4 Swiper supports images lazy loading.

To enable lazy loading, first of all we need special layout for images in slides:

```
<div class="swiper-container">
  <div class="swiper-wrapper">
    <div class="swiper-slide">
      
      <div class="swiper-lazy-preloader"></div>
    </div>
    <div class="swiper-slide">
      
      <div class="swiper-lazy-preloader"></div>
    </div>
    <div class="swiper-slide">
      
      <div class="swiper-lazy-preloader"></div>
    </div>
  </div>
</div>
```

As you see:

- Each lazy loaded image should have additional "swiper-lazy" class
- Image source should be specified in "data-src" attribute instead of "src"

You may also add animated preloader spinner to slide which will be removed automatically after image loaded:

```
<div class="swiper-lazy-preloader"></div>
```

Or white-one for dark layout:

```
<div class="swiper-lazy-preloader swiper-lazy-preloader-white"></div>
```

After that we need to enable lazy loading on Swiper initialization:

```
var swiper = new Swiper('.swiper-container', {
  // Disable preloading of all images
  preloadImages: false,
  // Enable Lazy Loading
  lazyLoading: true
});
```

If you use slidesPerView "auto" or slidesPerView > 1, then you should also enable **watchSlidesVisibility** and Swiper will load images in currently visible slides

Emitter API / Events

Emitter API allows to you add events/callbacks to Swiper like usual events, even after Swiper initialization:

```
// Init Swiper
var mySwiper = new Swiper('.swiper-container');

// Later add callback
mySwiper.on('slideChangeStart', function () {
  console.log('slide change start');
});

// Add one more handler for this event
mySwiper.on('slideChangeStart', function () {
  console.log('slide change start 2');
});
```

```
// Add handler that will be executed only once
mySwiper.once('sliderMove', function () {
    console.log('slider moved');
});

// Somewhen Later, remove all slideChangeStart handlers
mySwiper.off('slideChangeStart');
```

Note, that callback/event name in this case is the same as usual callback name but without "on" and with lowercase first character, so "onTouchMove" becomes just "touchMove"

Access to Swiper's Instance

If you initialize Swiper using HTML it is still possible to access to Swiper's instance. It is `swiper` property of Swiper's HTML container element:

```
var mySwiper = $('.swiper-container')[0].swiper;

// Now you can use all slider methods like
mySwiper.slideNext();
```

Like 248

Tweet 376

Follow @idangerous

783 followers

g+1

420

9.6K

 Flattr this!

Star 5,223

Fork 2,067

[Swiper](#) [Get Started](#) [API](#) [Demos](#) [Forum](#)

2015 © Swiper by iDangerous