API

Auto-Initialization

Cycle2 slideshows can be automatically initialized simply by adding the classname `cycle-slideshow` to your slideshow container element.

```
<div class="cycle-slideshow" ...
```

Cycle2 will automatically find and initialize a slideshow for any element that contains this classname. If you do not want this behavior then do not add the `cycle-slideshow` class to your slideshow and instead initalize the slideshow programmatically by invoking the `cycle` method on the slideshow container element:

```
$( '.mySlideshows' ).cycle();
```

Auto-initialization is not supported for slideshows that are added to the DOM after jQuery's ready event has fired. In this case you will need to programatically initialize your slideshow by invoking the `cycle` method as shown above. You do not need to qualify your selector to the part of the DOM that has been updated as Cycle2 will not re-initialize a running slideshow if you invoke `cycle` on it more than once. So it is perfectly safe to run the code above multiple times without having to worry about slideshows that are already running.

Options

Options are set on the slideshow element using `data-cycle-option-name` attributes. The option name is always lowercase and prefixed with `data-cycle-`. For example, to set the speed and manual-speed options you would do this:

```
<div class="cycle-slideshow" data-cycle-speed="600" data-cycle-manual-speed="100"> ...
```

This is the preferred method of setting slideshow options as it greatly reduces (or eliminates) the need for custom initialize script.

Options can also be set programatically if you prefer, by passing an options hash to the `cycle` method. When setting options programmically the option name does not contain the `data-cycle-` prefix and the string is camelCased on hyphens:

```
$('.cycle-slideshow').cycle({
    speed: 600,
    manualSpeed: 100
});
```

All options can be declared with `data-cycle-*` attributes on the container element and will be inherited by all slides.

Options in the table below that have a ● symbol can be overridden on individual slide elements in order to have specific properties for specific slides.

Name		Type	Default Value	Description
allow-wrap		boolean	true	This option determines whether or not a slideshow can advance from the last slide to the first (or vice versa). If set to `false` then once a slideshow reaches its last slide it can no longer be advanced forward, it can only advance backward. Likewise, when a slideshow is displaying the first slide it can not advance backward. By default a slideshow will advance seamlessly from the last slide to the first.
auto-height		integer or ratio	0	This option determines whether or not Cycle2 will provide height management for the slideshow which can be very useful in fluid or responsive designs. There are three ways this option can be used: • an integer can be provided to identify the zero-based slide index for a *sentinel* slide • the string "calc" can be provided to instruct Cycle2 to calculate the tallest slide and use it as the sentinel • a ratio string can be provided which identifies the width:height aspect ratio for the container By default, the slideshow's first slide is used as the *sentinel* slide and its demensions will control the height of the slideshow container if the container does not have a specific height set via CSS. To force the slideshow container to a specific aspect ratio, for example to hold a set of images that are 600x400, use a ratio string like this: `data-cycle-auto-height="600:400"` To disable height management, set this option's value to `-1` or `false`. See Auto Height demo for more info.
autoSelector (programmatic use only)		jQuery selector	'.cycle-slideshow[data-cycle-auto-init!=false]'	A jQuery selector which identifies elements that should be initialized automatically by Cycle2. The default value is `.cycle-slideshow`. Add the `cycle-slideshow` class to your slideshow container and Cycle2 will automatically find and initialize it when the DOM ready event fires. This value can only be changed programmatically and it can **not** be changed after the DOM ready event has fired. To change, include script like this ***after*** Cycle2 has loaded and ***before*** the *ready* event fires: `$.fn.cycle.defaults.autoSelector = '.mySlideshows';`
caption		jQuery	'> .cycle-	A selector which identifies the element that should be used

		selector	caption'	for the slideshow caption. By default, Cycle2 looks for an element with the class `.cycle-caption` that exists ***within*** the slideshow container.
caption-template	●	string	'{{slideNum}} / {{slideCount}}'	A template string which defines how the slideshow caption should be formatted. The template can reference real-time state information from the slideshow, such as the current slide index, etc. Cycle2 uses simple Mustache-style templates by default.
continueAuto (programatic use only)		boolean	undefined	Option which can be set dynamically to instruct C2 to stop transitioning on a timeout. This is useful when you want to start with an automatic slideshow but later switch to a manual slideshow. This option can also be a function which returns a boolean value.
delay		integer	0	The number of milliseconds to add onto, or substract from, the time before the first slide transition occurs.
disabled-class		string	disabled	The classname to assign to prev/next links when they cannot be activated (due to `data-cycle-allow-wrap="false"`. The value of this option should ***not*** include a preceding "dot".
easing	●	string	null	Name of the easing function to use for animations.
fx	●	string	'fade'	The name of the slideshow transition to use. The following transition names are available by default and more can be added with plugins: `fade`, `fadeout`, `none`, and `scrollHorz`.
hide-non-active		boolean	true	Determines whether or not Cycle2 hides the inactive slides.
loader		true | false | "wait"	false	The *loader* option sets the image loader behavior for the slideshow. Example. • `false` disabled loader functionality • `true` load slides as images arrive • `"wait"` wait for all images to arrive before staring slideshow
log		boolean	true	Set to `false` to disable console logging.
loop		integer	0	The number of times an auto-advancing slideshow should loop before terminating. If the value is less than 1 then the slideshow will loop continuously. Set to 1 to loop once, etc. Setting the `allow-wrap` option to false will override looping.
manual-fx		string	undefined	The transition effect to use for manually triggered transitions (not timer-based transitions).
manual-speed	●	integer	undefined	The speed (in milliseconds) for transitions that are manually initiated, such as those caused by clicking a "next" button or a pager link. By default, manual transitions occur at the same speed as automatic (timer-based) transitions.
manual-trump		boolean	true	Determines whether or not transitions are interrupted to

				begin new ones if the new ones are the result of a user action (not timer)
next		jQuery selector	'> .cycle-next'	A selector string which identifies the element (or elements) to use as a trigger to advance the slideshow forward. By default, Cycle2 looks for an element with the class `.cycle-next` that exists ***within*** the slideshow container.
next-event		jQuery selector	'click.cycle'	The event to bind to elements identified with the `next` option. By default, Cycle2 binds click events.
overlay		jQuery selector	'> .cycle-overlay'	A selector string which identifies the element to use as the overlay element. A slideshow overlay typically provides information about the current slide. By default, Cycle2 looks for an element with the class `.cycle-overlay` that exists ***within*** the slideshow container.
overlay-template	●	string	'<div>{{title}}</div><div>{{desc}}</div>'	A template string which defines how the overlay should be formatted. The template can reference real-time state information from the slideshow, such as the current slide index, etc. Cycle2 uses simple Mustache-style templates by default.
pager		jQuery selector	'> .cycle-pager'	A selector string which identifies the element to use as the container for pager links. By default, Cycle2 looks for an element with the class `.cycle-pager` that exists ***within*** the slideshow container.
pager-active-class		string (css classname)	'cycle-pager-active'	The classname to assign to pager links when a particular link references the currently visible slide. The value of this option should ***not*** include a preceding "dot".
pager-event		string (event name)	'click.cycle'	The type of event that is bound on the pager links. By default, Cycle2 binds click events.
pager-event-bubble		boolean	false	Set to `true` to allow pager events to bubble up the DOM. This is useful if you have an anchor inside your pager element and want the anchor to be followed when it is clicked.
pager-template	●	string	'<span>&bull;</span>'	A template string which defines how the pager links should be formatted. The template can reference real-time state information from the slideshow as each slide is added. The default pager link is simply a bullet. Cycle2 uses simple Mustache-style templates by default.
pause-on-hover		boolean or string	false	If `true` an auto-running slideshow will be paused while the mouse is over the slideshow. You may also specify a jQuery selector string for the value in order to specify the element(s) to which the hover events should be bound.
paused		boolean	false	If `true` the slideshow will begin in a *paused* state.
prev		jQuery selector	'> .cycle-prev'	A selector string which identifies the element (or elements) to use as a trigger to advance the slideshow backward. By

				default, Cycle2 looks for an element with the class `.cycle-prev` that exists ***within*** the slideshow container.
prev-event		string (event name)	'click.cycle'	The type of event that is bound on the prev and next links. By default, Cycle2 binds click events.
progressive		selector string	undefined	Identifies an element in the DOM which holds a JSON array representing the slides to be progressively loaded into the slideshow. Example.
random		boolean	false	If `true` the order of the slides will be randomized. This only effects slides that are initially in the markup, not slides added via the *add* command or via Cycle2's image loader functionality.
reverse		boolean	false	If `true` the slideshow will proceed in reverse order and transitions that support this option will run a reverse animation.
slide-active-class		string	cycle-slide-active	The classname to assign to the active slide. The value of this option should ***not*** include a preceding "dot".
slide-css	●	object hash	{}	An object which defines css properties that should be applied to each slide as it is initialized (once).
slide-class		string	"cycle-slide"	Name of the *class* to add to each slide.
slides		jQuery selector	'> img'	A selector string which identifies the elements within the slideshow container that should become slides. By default, Cycle2 finds all image elements that are immediate children of the slideshow container.
speed	●	integer	500	The speed of the transition effect in milliseconds.
starting-slide		integer	0	The zero-based index of the slide that should be initially displayed.
swipe		boolean	false	Set to `true` to enable swipe gesture support for advancing the slideshow forward or back.
swipe-fx		string	undefined	The transition effect to use for swipe-triggered transitions. If not provided the transition declared in the data-manual-fx or data-fx attribute will be used.
sync		boolean	true	If `true` then animation of the incoming and outgoing slides will be synchronized. If `false` then the animation for the incoming slide will not start until the animation for the outgoing slide completes.
timeout	●	integer	4000	The time between slide transitions in milliseconds.
tmpl-regex	●	string	{{((.)?.*?)}}	The default regular expression used for template tokenization.
update-view		number	0	Determines when the `updateView` method is invoked (and event is triggered). If the value is -1 then updateView is only invoked immediately after the slide transition.

			If the value is 0 then updateView is invoked during the slide transition. If the value is 1 then updateView is invoked immediately upon the beginning of a slide transition and again immediately after the transition.

Commands

Commands are issued to slideshows by invoking `cycle` on the container element and passing it a string argument, which is the command name.

It is not valid to issue a command to an element that has not previously been initialized as a Cycle2 slideshow.

Command	Description	Argument	Example Usage
add	Adds one or more slides to the slideshow.	Slide markup or jQuery object	`var newSlide = '<img src="pic.jpg">';` `$('.cycle-slideshow').cycle('add', newSlide);`
destroy	Restores slideshow DOM to it's original state (and unbinds events).	none	`$('.cycle-slideshow').cycle('destroy');`
goto	Transitions the slideshow to the slide index provided.	zero-based slide index	`// goto 3rd slide` `$('.cycle-slideshow').cycle('goto', 2);`
next	Transitions the slideshow to the next slide.	none	`$('.cycle-slideshow').cycle('next');`
pause	Pauses an auto-running slideshow.	none	`$('.cycle-slideshow').cycle('pause');`
prev	Transitions the slideshow to the previous slide.	none	`$('.cycle-slideshow').cycle('prev');`
reinit	Reinitializes a slideshow. This is equivalent to issuing the 'destroy' command and then invoking `cycle()` again.	none	`$('.cycle-slideshow').cycle('reinit');`
remove	Removes a slide from a running slideshow	zero-based slide index	`// remove 2nd slide` `$('.cycle-slideshow').cycle('remove', 1);`
resume	Resumes a paused slideshow.	none	`$('.cycle-slideshow').cycle('resume');`
stop	Stops an auto-running slideshow.	none	`$('.cycle-slideshow').cycle('stop');`

Declarative Commands

It is possible to issue commands declaratively by using the `data-cycle-cmd` attribute. You can use this attribute on any element and Cycle2 will use event delegation to issue a cycle command when the element is clicked. For example, to have a random button element pause your slideshow you can do this:

```
<button data-cycle-cmd="pause">Pause</button>
```

When the above button is clicked, Cycle2 will auto-generate and run this code for you:

```
$('.cycle-slideshow').cycle('pause');
```

To specify a specific slideshow as the target of the command, use the `data-cycle-context` attribute and set its value to a selector string that identifies the desired slideshow:

```
<button data-cycle-cmd="pause" data-cycle-context="#mySlideshow">Pause</button>
```

When the above button is clicked, Cycle2 will auto-generate and run this code for you:

```
$('#mySlideshow').cycle('pause');
```

And finally, you can pass an argument to the cycle command using the `data-cycle-arg` attribute. For example, to remove the first slide in a slideshow:

```
<button data-cycle-cmd="remove" data-cycle-context="#mySlideshow" data-cycle-arg="0">Remove Slide</button>
```

Events

Cycle2 emits numerous events as a slideshow runs and you can bind to these events in order to further customize or augment the slideshow's behavior.

All cycle events are triggered on the slideshow element and so can be bound using code like:

```
$( '#mySlideshow' ).on( 'cycle-eventname', function( event, opts ) {
    // your event handler code here
    // argument opts is the slideshow's option hash
});
```

Event Name	Description / Callback Signature
cycle-after	Triggered after the slideshow has completed transitioning to the next slide. `function(event, optionHash, outgoingSlideEl, incomingSlideEl, forwardFlag)`
cycle-before	Triggered just prior to a transition to a new slide. `function(event, optionHash, outgoingSlideEl, incomingSlideEl, forwardFlag)`
cycle-bootstrap	This is the first event triggered by a slideshow and it provides an opportunity to override the slideshow options and API methods. The *API* arg is an object hash of this slideshow instance's API methods. `function(event, optionHash, API)`
cycle-destroyed	Triggered after the slideshow has been destroyed by the 'destroy' command. `function(event, optionHash)`

cycle-finished	Triggered after the slideshow has terminated due to the 'loop' option. `function(event, optionHash)`
cycle-initialized	Triggered after a slideshow has been fully initalized. `function(event, optionHash)`
cycle-next	Triggered after the slideshow has started advancing due to the 'next' command. `function(event, optionHash)`
cycle-pager-activated	Triggered after the slideshow has started transitioning due to a pager link event. `function(event, optionHash)`
cycle-paused	Triggered after the slideshow has been paused as a result of either the 'pause' command or the 'pause-on-hover' option. `function(event, optionHash)`
cycle-post-initialize	Triggered immediately after running the full initialation logic on the slideshow. `function(event, optionHash)`
cycle-pre-initialize	Triggered prior to running the full initialation logic on the slideshow (but after the initial slide set has been added). `function(event, optionHash)`
cycle-prev	Triggered after the slideshow has started advancing due to the 'prev' command. `function(event, optionHash)`
cycle-resumed	Triggered after the slideshow has been paused as a result of either the 'resume' command or the 'pause-on-hover' option. `function(event, optionHash)`
cycle-slide-added	Triggered after a slide has been added to the slideshow `function(event, jQueryWrappedSlideEl)`
cycle-slide-removed	Triggered after a slide has been removed from the slideshow due to the 'remove' command. `function(event, indexOfSlideRemoved, removedSlideEl)`
cycle-stopped	Triggered after the slideshow has been stopped by the 'stop' command. `function(event, optionHash)`
cycle-transition-stopped	Triggered after animation of one or more slides has been interrupted due to manual slide advancement or due to either the 'stop' or 'destroy' command being issued. `function(event, optionHash)`

cycle-update-view	Triggered after initialization, after a slide transition, and after a slide has been added or removed. `function(event, optionHash, slideOptionsHash, currentSlideEl)`

Miscellaneous Bits

State

The Cycle2 options hash contains all of the slideshow's options detailed above, along with the following additional pieces of run-time state.

State data is read-only.

Name	Description
busy	`true` when a slide transition is active.
currSlide	Zero-based index of the current slide.
nextSlide	Zero-based index of the next slide to be displayed.
paused	`true` if the slideshow is paused.
slideNum	Useful in template strings for displaying a slide's one-based index.
slideCount	Total number of slides in the slideshow.

Testing the Paused State

Cycle2 assignes the class `cycle-paused` to the slideshow container when the slideshow is paused. Scripts can determine if the slideshow is currently paused using code like this:

```
var paused = $('#mySlideshow').is( '.cycle-paused' );
```

hash

Each slide in your slideshow can have an attribute named `data-cycle-hash`. If you assign a value to this attribute then the location hash of the current page will be updated to reflect that value when the slide is active. Likewise, when the page is loaded if the location hash is equal to the `data-cycle-hash` of one of the slides then that slide will be made active. See the Bookmarkable demo for more information.