

---

# **classjs Documentation**

*Release 1.0*

**Angelo Dini**

March 21, 2015



<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Plugins</b>	<b>5</b>
2.1	Cl.Accordion . . . . .	5
2.2	Cl.Autocomplete . . . . .	7
2.3	Cl.Carousel . . . . .	8
2.4	Cl.Debug . . . . .	10
2.5	Cl.Gallery . . . . .	11
2.6	Cl.Lightbox . . . . .	13
2.7	Cl.Mobilemenu . . . . .	16
2.8	Cl.Uniform . . . . .	18



These plugins are aimed for professionals with extensive JavaScript knowledge. Each plugin follows a simplistic and a modular approach. All plugins have demos, tests and documentation regarding their options, methods, events and callbacks.

The plugins are compatible with Firefox, Chrome, Safari and IE7+. We also support iOS, Android and Windows Phone to insure correct responsive behaviour with full accessible support according to the WCAG 2.0 standards. Each plugin requires at least class.js 1.0+ and jQuery 1.7+.

The latest stable versions are available on github - <https://github.com/finalangel/classjs-plugins>



---

## Requirements

---

For additional information on how to run documentation or the examples locally, consult the docs/README.rst or the demo page's FAQ section: <http://finalangel.github.io/classjs-plugins/>



## 2.1 Cl.Accordion

This is a list of all relevant options, methods, events and callbacks.

### 2.1.1 Options

Options are set on initializing the accordion:

```
new Cl.Accordion({
  // your options
  'index': 0,
  'event': 'mouseenter'
});
```

Option	De- fault	Description
index	null	shows selected element[index] starting form zero.
expanded	false	initial visibility of all items inside the selector.
event	'click'	the jquery element of which the accordion will listen to.
easing	'swing'	the jquery easing method for all animations.
duration	300	the jquery duration speed for all animations.
grouping	true	enables all elements to be grouped together, only one will be visible at any given time.
forceClose	false	if enabled, allows open element to be closed again.
disableAn- chors	true	disables descending anchors from triggering. For example if the trigger is an <code>li</code> wrapping an anchor.
autoHeight	false	sets the height for each element on initialization.
cls	object	the available css class getters and setters.
lang	object	the available language configurations.

Option `cls` setters and getters:

Option	Default	Description
expanded	'expanded'	will be added on the trigger when expanded
collapsed	'collapsed'	will be added on the trigger when collapsed
trigger	'trigger'	the element which triggers <code>toggle</code>
container	'container'	the container to be expanded and collapsed
text	'text'	will be used to switch language text

Option `lang` strings:

Option	Default
expanded	'Expanded '
collapsed	'Collapsed '

## 2.1.2 Methods

Methods can be called using an instance of the class:

```
var accordion = new Cl.Accordion();  
// trigger the method  
accordion.toggle();
```

All Methods have appropriate events and callbacks.

`accordion.toggle(index)`

**Description** toggles a specified accordion entry.

**Arguments**

- **index** (*number*) – required the index of the element to be toggled.

**Returns** toggle callback.

`accordion.show([index])`

**Description** opens an accordion entry.

**Arguments**

- **index** (*number*) – optional index of the element to be shown, if empty shows all.

**Returns** open callback.

`accordion.hide([index])`

**Description** hides an accordion entry.

**Arguments**

- **index** (*number*) – optional index of the element to be shown, if empty hides all.

**Returns** close callback.

## 2.1.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var accordion = new Cl.Accordion();  
// register callback  
accordion.callbacks.toggle = function () {  
    console.log('item has toggled');  
};
```

*Available keywords:*

**toggle**

is called when triggering method ``toggle``.

**open**

is called when triggering method ``open``.

**close**

is called when triggering method ```close```.

## 2.2 Cl.Autocomplete

This is a list of all relevant options, methods, events and callbacks.

### 2.2.1 Options

Options are set on initializing autocomplete:

```
new Cl.Autocomplete({
  // your options
  'minLength': 1,
  'fx': 'toggle'
});
```

Option	Default	Description
url	false	sd
minLength	3	sd
easing	'swing'	the jquery easing method for all animations.
duration	300	the jquery duration speed for all animations.
delay	300	the jquery duration speed for all animations.
fx	'slide'	the jquery duration speed for all animations.
cls	object	the available css class getters and setters.
lang	object	the available language strings.

Option `cls` setters and getters:

### 2.2.2 Methods

Methods can be called using an instance of the class:

```
var mobilemenu = new Cl.Mobilemenu();
// trigger the method
mobilemenu.toggle();
```

All Methods have appropriate events and callbacks.

`mobilemenu.toggle(index)`

**Description** toggles a specified mobilemenu entry.

**Arguments**

- **index** (*number*) – required the index of the element to be toggled.

**Returns** toggle callback.

`mobilemenu.show([index])`

**Description** opens an mobilemenu entry.

**Arguments**

- **index** (*number*) – optional index of the element to be shown, if empty shows all.

**Returns** open callback.

mobilemenu.**hide** (*[index]*)

**Description** hides an mobilemenu entry.

**Arguments**

- **index** (*number*) – optional index of the element to be shown, if empty hides all.

**Returns** close callback.

## 2.2.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var mobilemenu = new Cl.Mobilemenu();
// register callback
mobilemenu.callbacks.toggle = function () {
    console.log('item has toggled');
};
```

Available keywords:

**toggle**

is called when triggering method ``toggle``.

**open**

is called when triggering method ``open``.

**close**

is called when triggering method ``close``.

## 2.3 Cl.Carousel

This is a list of all relevant options, methods, events and callbacks.

### 2.3.1 Options

Options are set on initializing the carousel:

```
new Cl.Carousel({
    // your options
    'index': 0,
    'move': 'single'
});
```

Option	Default	Description
index	null	shows selected element[index]/group[index] starting form zero.
timeout	null	timeout in ms for autoplay, if 0 or null autoplay is ignored.
autoplay	false	continues timeout even after manual cancellation.
easing	'linear'	the jquery easing method for all animations.
duration	300	the jquery duration speed for all animations.
move	'auto'	either <code>single</code> to move one element at a time or <code>auto</code> to move all visible elements.
momentum	true	enables <code>next</code> and <code>previous</code> to jump to the end or beginning when reaching their bounds.
cls	object	the available css class getters and setters.

Option `cls` setters and getters:

Option	Default	Description
active	'active'	class will be added to the active item.
disabled	'disabled'	class will be added to the triggers when bound is reached and momentum is false.
wrapper	'wrapper'	the outer relative positioned wrapper where the height will be added.
viewport	'viewport'	the inner absolute positioned wrapper that will be moved.
elements	'item'	the element / individual items.
next	'trigger-next a'	element which triggers the method <code>next</code> .
previous	'trigger-previous a'	element which triggers the method <code>previous</code> .
naviga- tion	'nav a'	navigation elements which trigger the method <code>move</code> with their corresponding index.

### 2.3.2 Methods

Methods can be called using an instance of the class:

```
var carousel = new Cl.Carousel();
// trigger the instance
carousel.toggle();
```

All Methods have appropriate events and callbacks.

```
carousel.next()
```

**Description** goes to the next carousel slide.

**Returns** next callback.

```
carousel.previous()
```

**Description** goes to the previous carousel slide.

**Returns** previous callback.

```
carousel.move(index)
```

**Description** moves to a carousel entry.

**Arguments**

- **index** (*number*) – index of the element to be moved to.

**Returns** move callback.

```
carousel.play()
```

**Description** starts the carousel timeout.

**Returns** play callback.

```
carousel.stop()
```

**Description** stops the carousel timeout.

**Returns** stop callback.

```
carousel.update()
```

**Description** updates the carousels states.

**Returns** update callback.

```
carousel.destroy()
```

**Description** removes all states and events.

**Returns** destroy callback.

### 2.3.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var carousel = new Cl.Carousel();
// register callback
carousel.callbacks.move = function (scope) {
  console.log('caourse1 is moving to position ' + scope.index);
};
```

*Available keywords:*

**next**

is called when triggering method ``next``.

**previous**

is called when triggering method ``previous``.

**move**

is called when triggering method ``move``.

**play**

is called when triggering method ``play``.

**stop**

is called when triggering method ``stop``.

**update**

is called when triggering method ``update``.

**destroy**

is called when triggering method ``destroy``.

## 2.4 Cl.Debug

This is a list of all relevant options, methods, events and callbacks.

### 2.4.1 Options

Options are set on initializing debug:

```
new Cl.Debug({
  // your options
  'close': true
});
```

Option	Default	Description
closed	false	defines if the button is visible or not.
collapsed	true	defines if the menu is expanded or collapsed.

## 2.4.2 Methods

append `#show` to url for enabling the button when closed.

## 2.4.3 Callbacks

There are no callbacks so far.

## 2.5 Cl.Gallery

This is a list of all relevant options, methods, events and callbacks.

### 2.5.1 Options

Options are set on initializing the gallery:

```
new Cl.Gallery({
  // your options
  'index': 0,
  'engine': 'slide'
});
```

Option	Default	Description
index	null	shows selected element[index]/group[index] starting form zero.
timeout	5000	timeout in ms for autoplay, if 0 or null autoplay is ignored.
autoplay	false	continues timeout even after manual cancellation.
easing	'swing'	the jquery easing method for all animations.
duration	300	the jquery duration speed for all animations.
autoHeight	true	sets the height on the warpper to the heighest element.
autoResize	true	sets the height on the warpper on window.resize.
engine	'fade'	this is the engine to be loaded when animating. Build-in are <code>fade</code> and <code>slide</code> .
cls	object	the available css class getters and setters.

Option `cls` setters and getters:

Option	Default	Description
active	'active'	class will be added to the active item.
wrapper	'wrapper'	the outer relative positioned wrapper where the height will be added.
viewport	'viewport'	the inner absolute positioned wrapper that will be moved.
elements	'item'	the element / individual items.
next	'trigger-next a'	element which triggers the method <code>next</code> .
previous	'trigger-previous a'	element which triggers the method <code>previous</code> .
naviga- tion	'nav a'	navigation elements which trigger the method <code>move</code> whith their corresponding index.

### 2.5.2 Methods

Methods can be called using an instance of the class:

```
var gallery = new Cl.Gallery();
// trigger the instance
gallery.toggle();
```

All Methods have appropriate events and callbacks.

gallery.**next**()

**Description** goes to the next gallery slide.

**Returns** next callback.

gallery.**previous**()

**Description** goes to the previous gallery slide.

**Returns** previous callback.

gallery.**move**(*index*)

**Description** moves to a gallery entry.

**Arguments**

- **index** (*number*) – index of the element to be moved to.

**Returns** move callback.

gallery.**play**()

**Description** starts the gallery timeout.

**Returns** play callback.

gallery.**stop**()

**Description** stops the gallery timeout.

**Returns** stop callback.

gallery.**update**()

**Description** updates gallery to current index.

**Returns** update callback.

### 2.5.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var gallery = new Cl.Gallery();
// register callback
gallery.callbacks.move = function (scope) {
  console.log('gallery is moving to element ' + scope.index);
};
```

*Available keywords:*

**next**

is called when triggering method ``next``.

**previous**

is called when triggering method ``previous``.

**move**

is called when triggering method ``move``.

**play**

is called when triggering method ``play``.

**stop**

is called when triggering method ``stop``.

**update**

is called when triggering method ``update``.

## 2.6 Cl.Lightbox

This is a list of all relevant options, methods, events and callbacks.

### 2.6.1 Options

Options are set on initializing the lightbox:

```
new Cl.Lightbox({
  'group': false,
  'responsive': false
});
```

Option	De- fault	Description
prefix	'cl'	the prefix is attached to all internal events and css classes for example cl-lightbox or cl-lightbox.
group	true	allows grouping of matching elements and enables status and navigation.
cycle	true	requires group allows previous and next methods to cycle through matching elements.
modal	true	enables the dimmer functionality.
modalClick- able	true	requires modal allows the dimmer to be closed onclick.
modal- Closable	true	requires modalClickable disables all close events, lightbox can be only closed using the API.
forceLoad	false	if enabled, insures that iframes are fully loaded before display.
easing	'swing'	jquery easing effect for all animations.
duration	300	duration until the lightbox is fully expanded.
speed	300	speed for all regular animations.
fixed	true	sets the lightbox always inside the viewport of the user even when scrolling.
responsive	true	enables responsive behaviour of the lightbox.
ajax	false	ajax loads the provided url and tries to inject the html into the lightbox. It does not create an iframe in order to maintain the css style.
controls	true	requires group enables controllable elements when a collection is active.
cls	''	adds an additional class to the top level element.
opacity	0.8	“requires modal“transparent level for dimmer.
styles	Ob- ject	adds jquery style css object to gallery content element.
dimen- sions	Ob- ject	available dimensions are initialWidth, initialHeight, offset, width and height.
keys	true	enables control for the lightbox using the keyboard.
keyCodes	Ob- ject	requires keys enables key control for close, next and previous.
lang	Ob- ject	the available language configurations.

Option lang strings:

Option	Default
close	'Close lightbox'
errorMessage	'<p><strong>The requested element...</p>'
next	'Next'
previous	'Previous'
status	'Slide {current} of {total}.'

## 2.6.2 Methods

Methods can be called using an instance of the class:

```
var lightbox = new Cl.Lightbox();
// trigger the instance
lightbox.open();
```

All Methods have appropriate events and callbacks.

instance.**open**(url)

**Description** opens the lightbox with the provided url or jQuery element.

**Arguments**

- **type** (*jquery*) – either url or jQuery element.

**Returns** open callback`instance.close()`**Description** closes the lightbox.**Returns** close callback`instance.resize(width, height)`**Description** resizes the lightbox to the specified dimensions.**Arguments**

- **width** (*number*) – the width the lightbox should be resized to.
- **height** (*number*) – the height the lightbox should be resized to.

**Returns** resize callback`instance.destroy()`**Description** removes the lightbox from the dom.**Returns** destroy callback`instance.next()`**Description** movies to the next element.**Returns** next callback`instance.previous()`**Description** movies to the previous element.**Returns** previous callback`instance.getElement()`**Returns** the current visible element`instance.getCollection()`**Returns** all current elements in the collection

## 2.6.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var lightbox = new Cl.Lightbox();
// register callback
lightbox.callbacks.close = function () {
  console.log('lightbox is closing');
};
```

*Available keywords:*

**open**

is called when triggering method ``open``.

**close**

is called when triggering method ``close``.

**resize**

is called when triggering method ``resize``.

**destroy**

is called when triggering method ``destroy``.

**next**

is called when triggering method ``next``.

**previous**

is called when triggering method ``previous``.

*Private keywords:*

**load**

is called when triggering private method ``load``.

**complete**

is called when triggering private method ``complete``.

**unload**

is called when triggering private method ``unload``.

## 2.7 Cl.Mobilemenu

This is a list of all relevant options, methods, events and callbacks.

### 2.7.1 Options

Options are set on initializing the mobilemenu:

```
new Cl.Mobilemenu({
  // your options
  'duration': 0,
  'ratio': 80 / 100
});
```

Option	Default	Description
easing	'swing'	the jquery easing method for all animations.
duration	300	the jquery duration speed for all animations.
bound	539	the bound used when the mobilemenu should show up.
ratio	70 / 100	the ratio used for how much the menu is taking from the viewport.
fixedRatio	null	same as ratio just in pixel instead of multiplicator.
offset	object	offsets to left or top of the menu.
cls	object	the available css class getters and setters.
overlay	'	the generated overlay.

Option `cls` setters and getters:

Option	Default	Description
menu	'mainnav'	will be added on the trigger when expanded
inner	'> ul'	will be added on the trigger when collapsed
knob	'mainnav-knob'	the element which triggers <code>toggle</code>

## 2.7.2 Methods

Methods can be called using an instance of the class:

```
var mobilemenu = new Cl.Mobilemenu();
// trigger the method
mobilemenu.toggle();
```

All Methods have appropriate events and callbacks.

`mobilemenu.toggle(index)`

**Description** toggles a specified mobilemenu entry.

**Arguments**

- **index (number)** – required the index of the element to be toggled.

**Returns** toggle callback.

`mobilemenu.show([index])`

**Description** opens an mobilemenu entry.

**Arguments**

- **index (number)** – optional index of the element to be shown, if empty shows all.

**Returns** open callback.

`mobilemenu.hide([index])`

**Description** hides an mobilemenu entry.

**Arguments**

- **index (number)** – optional index of the element to be shown, if empty hides all.

**Returns** close callback.

## 2.7.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var mobilemenu = new Cl.Mobilemenu();
// register callback
mobilemenu.callbacks.toggle = function () {
  console.log('item has toggled');
};
```

*Available keywords:*

**toggle**

is called when triggering method ``toggle``.

**show**

is called when triggering method ``show``.

**hide**

is called when triggering method ``hide``.

## 2.8 Cl.Uniform

This is a list of all relevant options, methods, events and callbacks.

### 2.8.1 Options

Options are set on initializing the carousel:

```
new Cl.Uniform();
```

Option	Default	Description
offset	-9999	the offset of the input related to the surrounding container.
cls	object	the available css class strings.
lang	object	the available language configurations.
tpl	object	the html template which will be used to replace the forms.

Option `cls` strings:

Option	Default	Description
prefix	'uniform'	will be added before all following classes.
radio	'radio'	will be used as the wrapper for radio elements.
checkbox	'checkbox'	will be used as the wrapper for checkbox elements.
file	'file'	will be used as the wrapper for file elements.
select	'select'	will be used as the wrapper for select elements.
disabled	'disabled'	will be added whenever the attribute <code>disabled="disabled"</code> is defined.
focus	'focus'	will be added whenever <code>focus</code> is triggered on the element.
ready	'ready'	will be added when the uniform is ready on this particular field.
checked	'checked'	will be added when checkbox/radio is checked/selected

Option `lang` strings:

Option	Default
fileBtn	'Upload'
fileStatus	'Please select a file...'

Option `tpl` strings:

Option	Default
radio	dom element (string)
checkbox	dom element (string)
file	dom element (string)
select	dom element (string)

### 2.8.2 Methods

Methods can be called using an instance of the class:

```
var uniform = new Cl.Uniform();  
// trigger the instance  
uniform.update();
```

All Methods have appropriate events and callbacks.

`uniform.update()`

**Description** checks all form elements and updates their states accordingly.

**Returns** update callback.

`uniform.destroy()`

**Description** removes uniform from form elements.

**Returns** destroy callback.

### 2.8.3 Callbacks

**Callbacks** are always triggered **after** the method is executed. You can interact with callbacks as follows:

```
var uniform = new Cl.Uniform();  
// register callback  
uniform.callbacks.update = function (scope) {  
    console.log('uniform is executing update');  
};
```

*Available keywords:*

**update**

is called when triggering method ``update``.

**destroy**

is called when triggering method ``destroy``.



**A**

accordion.hide() (accordion method), 6  
 accordion.show() (accordion method), 6  
 accordion.toggle() (accordion method), 6

**C**

carousel.destroy() (carousel method), 9  
 carousel.move() (carousel method), 9  
 carousel.next() (carousel method), 9  
 carousel.play() (carousel method), 9  
 carousel.previous() (carousel method), 9  
 carousel.stop() (carousel method), 9  
 carousel.update() (carousel method), 9  
 close (global variable or constant), 6, 8  
 close (None attribute), 15  
 complete (None attribute), 16

**D**

destroy (global variable or constant), 10, 19  
 destroy (None attribute), 16

**G**

gallery.move() (gallery method), 12  
 gallery.next() (gallery method), 12  
 gallery.play() (gallery method), 12  
 gallery.previous() (gallery method), 12  
 gallery.stop() (gallery method), 12  
 gallery.update() (gallery method), 12

**H**

hide (global variable or constant), 17

**I**

instance.close() (instance method), 15  
 instance.destroy() (instance method), 15  
 instance.getCollection() (instance method), 15  
 instance.getElement() (instance method), 15  
 instance.next() (instance method), 15  
 instance.open() (instance method), 14  
 instance.previous() (instance method), 15

instance.resize() (instance method), 15  
 is called when triggering method “close“. (global variable or constant), 6, 8  
 is called when triggering method “close“. (is called when triggering method “close“ attribute), 15  
 is called when triggering method “destroy“. (global variable or constant), 10, 19  
 is called when triggering method “destroy“. (is called when triggering method “destroy“ attribute), 16  
 is called when triggering method “hide“. (global variable or constant), 17  
 is called when triggering method “move“. (global variable or constant), 10, 12  
 is called when triggering method “next“. (global variable or constant), 10, 12  
 is called when triggering method “next“. (is called when triggering method “next“ attribute), 16  
 is called when triggering method “open“. (global variable or constant), 6, 8  
 is called when triggering method “open“. (is called when triggering method “open“ attribute), 15  
 is called when triggering method “play“. (global variable or constant), 10, 13  
 is called when triggering method “previous“. (global variable or constant), 10, 12  
 is called when triggering method “previous“. (is called when triggering method “previous“ attribute), 16  
 is called when triggering method “resize“. (is called when triggering method “resize“ attribute), 15  
 is called when triggering method “show“. (global variable or constant), 17  
 is called when triggering method “stop“. (global variable or constant), 10, 13  
 is called when triggering method “toggle“. (global variable or constant), 6, 8, 17  
 is called when triggering method “update“. (global variable or constant), 10, 13, 19  
 is called when triggering private method “complete“. (is called when triggering private method “com-

plete“ attribute), 16  
is called when triggering private method “load“. (is called when triggering private method “load“ attribute), 16

is called when triggering private method “unload“. (is called when triggering private method “unload“ attribute), 16

## L

load (None attribute), 16

## M

mobilemenu.hide() (mobilemenu method), 7, 17

mobilemenu.show() (mobilemenu method), 7, 17

mobilemenu.toggle() (mobilemenu method), 7, 17

move (global variable or constant), 10, 12

## N

next (global variable or constant), 10, 12

next (None attribute), 16

## O

open (global variable or constant), 6, 8

open (None attribute), 15

## P

play (global variable or constant), 10, 13

previous (global variable or constant), 10, 12

previous (None attribute), 16

## R

resize (None attribute), 15

## S

show (global variable or constant), 17

stop (global variable or constant), 10, 13

## T

toggle (global variable or constant), 6, 8, 17

## U

uniform.destroy() (uniform method), 19

uniform.update() (uniform method), 18

unload (None attribute), 16

update (global variable or constant), 10, 13, 19