[ScrollMagic](http://docs.google.com/index.html)

* [Classes](http://docs.google.com/classes.list.html)
  + [Controller](http://docs.google.com/ScrollMagic.Controller.html)
  + [Scene](http://docs.google.com/ScrollMagic.Scene.html)
* [Events](http://docs.google.com/events.list.html)
  + [add](http://docs.google.com/ScrollMagic.Scene.html#event:add)
  + [change](http://docs.google.com/ScrollMagic.Scene.html#event:change)
  + [destroy](http://docs.google.com/ScrollMagic.Scene.html#event:destroy)
  + [end](http://docs.google.com/ScrollMagic.Scene.html#event:end)
  + [enter](http://docs.google.com/ScrollMagic.Scene.html#event:enter)
  + [leave](http://docs.google.com/ScrollMagic.Scene.html#event:leave)
  + [progress](http://docs.google.com/ScrollMagic.Scene.html#event:progress)
  + [remove](http://docs.google.com/ScrollMagic.Scene.html#event:remove)
  + [shift](http://docs.google.com/ScrollMagic.Scene.html#event:shift)
  + [start](http://docs.google.com/ScrollMagic.Scene.html#event:start)
  + [update](http://docs.google.com/ScrollMagic.Scene.html#event:update)
* [Plugins](http://docs.google.com/mixins.list.html)
  + [GSAP](http://docs.google.com/animation.GSAP.html)
  + [Velocity](http://docs.google.com/animation.Velocity.html)
  + [addIndicators](http://docs.google.com/debug.addIndicators.html)
  + [jQuery](http://docs.google.com/framework.jQuery.html)

Classes

### Classes

### [Controller](http://docs.google.com/ScrollMagic.Controller.html) [Scene](http://docs.google.com/ScrollMagic.Scene.html)

Namespaces

### [ScrollMagic](http://docs.google.com/ScrollMagic.html)

Events

#### add

Scene add event.

Fires when the scene is added to a controller.

This is mostly used by plugins to know that change might be due.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | controller | boolean | The controller object the scene was added to. | |

Since:

* 2.0.0

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 192](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-192)

##### Example

scene.on("add", function (event) {  
 console.log('Scene was added to a new controller.');  
});

#### change

Scene change event.

Fires whenvever a property of the scene is changed.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | what | string | Indicates what value has been changed | | newval | mixed | The new value of the changed property | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 130](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-130)

##### Example

scene.on("change", function (event) {  
 console.log("Scene Property \"" + event.what + "\" changed to " + event.newval);  
});

#### destroy

Scene destroy event.

Fires whenvever the scene is destroyed.

This can be used to tidy up custom behaviour used in events.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | reset | boolean | Indicates if the destroy method was called with reset true or false. | |

Since:

* 1.1.0

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 167](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-167)

##### Example

scene.on("enter", function (event) {  
 // add custom action  
 $("#my-elem").left("200");  
 })  
 .on("destroy", function (event) {  
 // reset my element to start position  
 if (event.reset) {  
 $("#my-elem").left("0");  
 }  
 });

#### end

Scene end event.

Fires whenever the scroll position its the ending point of the scene.

It will also fire when scrolling back up from after the scene and going over its end position. If you want something to happen only when scrolling down/right, use the scrollDirection parameter passed to the callback.

For details on this event and the order in which it is fired, please review the Scene.progress method.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | progress | number | Reflects the current progress of the scene | | state | string | The current state of the scene "DURING" or "AFTER" | | scrollDirection | string | Indicates which way we are scrolling "PAUSED", "FORWARD" or "REVERSE" | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 29](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-29)

##### Example

scene.on("end", function (event) {  
 console.log("Hit end point of scene.");  
});

#### enter

Scene enter event.

Fires whenever the scene enters the "DURING" state.

Keep in mind that it doesn't matter if the scene plays forward or backward: This event always fires when the scene enters its active scroll timeframe, regardless of the scroll-direction.

For details on this event and the order in which it is fired, please review the Scene.progress method.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | progress | number | Reflects the current progress of the scene | | state | string | The current state of the scene - always "DURING" | | scrollDirection | string | Indicates which way we are scrolling "PAUSED", "FORWARD" or "REVERSE" | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 50](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-50)

##### Example

scene.on("enter", function (event) {  
 console.log("Scene entered.");  
});

#### leave

Scene leave event.

Fires whenever the scene's state goes from "DURING" to either "BEFORE" or "AFTER".

Keep in mind that it doesn't matter if the scene plays forward or backward: This event always fires when the scene leaves its active scroll timeframe, regardless of the scroll-direction.

For details on this event and the order in which it is fired, please review the Scene.progress method.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | progress | number | Reflects the current progress of the scene | | state | string | The current state of the scene "BEFORE" or "AFTER" | | scrollDirection | string | Indicates which way we are scrolling "PAUSED", "FORWARD" or "REVERSE" | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 71](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-71)

##### Example

scene.on("leave", function (event) {  
 console.log("Scene left.");  
});

#### progress

Scene progress event.

Fires whenever the progress of the scene changes.

For details on this event and the order in which it is fired, please review the Scene.progress method.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | progress | number | Reflects the current progress of the scene | | state | string | The current state of the scene "BEFORE", "DURING" or "AFTER" | | scrollDirection | string | Indicates which way we are scrolling "PAUSED", "FORWARD" or "REVERSE" | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 110](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-110)

##### Example

scene.on("progress", function (event) {  
 console.log("Scene progress changed to " + event.progress);  
});

#### remove

Scene remove event.

Fires when the scene is removed from a controller.

This is mostly used by plugins to know that change might be due.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | |

Since:

* 2.0.0

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 210](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-210)

##### Example

scene.on("remove", function (event) {  
 console.log('Scene was removed from its controller.');  
});

#### shift

Scene shift event.

Fires whenvever the start or end **scroll offset** of the scene change.

This happens explicitely, when one of these values change: offset, duration or triggerHook.

It will fire implicitly when the triggerElement changes, if the new element has a different position (most cases).

It will also fire implicitly when the size of the container changes and the triggerHook is anything other than onLeave.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | reason | string | Indicates why the scene has shifted | |

Since:

* 1.1.0

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 147](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-147)

##### Example

scene.on("shift", function (event) {  
 console.log("Scene moved, because the " + event.reason + " has changed.)");  
});

#### start

Scene start event.

Fires whenever the scroll position its the starting point of the scene.

It will also fire when scrolling back up going over the start position of the scene. If you want something to happen only when scrolling down/right, use the scrollDirection parameter passed to the callback.

For details on this event and the order in which it is fired, please review the Scene.progress method.

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | progress | number | Reflects the current progress of the scene | | state | string | The current state of the scene "BEFORE" or "DURING" | | scrollDirection | string | Indicates which way we are scrolling "PAUSED", "FORWARD" or "REVERSE" | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 8](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-8)

##### Example

scene.on("start", function (event) {  
 console.log("Hit start point of scene.");  
});

#### update

Scene update event.

Fires whenever the scene is updated (but not necessarily changes the progress).

##### Properties:

| Name | Type | Description |
| --- | --- | --- |
| event | object | The event Object passed to each callback Properties  | Name | Type | Description | | --- | --- | --- | | type | string | The name of the event | | target | Scene | The Scene object that triggered this event | | startPos | number | The starting position of the scene (in relation to the conainer) | | endPos | number | The ending position of the scene (in relation to the conainer) | | scrollPos | number | The current scroll position of the container | |

Source:

* [ScrollMagic/Scene/event-management.js](http://docs.google.com/ScrollMagic_Scene_event-management.js.html), [line 92](http://docs.google.com/ScrollMagic_Scene_event-management.js.html#sunlight-1-line-92)

##### Example

scene.on("update", function (event) {  
 console.log("Scene updated.");  
});

© Jan Paepke 2015

Documentation generated by [JSDoc 3.5.5](https://github.com/jsdoc3/jsdoc) using a customized version of the [DocStrap template](https://github.com/terryweiss/docstrap).