

# Phaser.Tween

## Phaser 2.3.0

[Game States](#)[Loader](#)[Game Scaling](#)[Signals](#)[Plugins](#)

## Game Objects

[Display](#)[Graphics](#)[Text](#)[Animation](#)[Geometry](#)**Jump to:**[Properties](#)[Methods](#)

## Constructor

**new Tween(target, game, manager)**

A Tween allows you to alter one or more properties of a target object over a defined period of time. This can be used for things such as alpha fading Sprites, scaling them or motion. Use `Tween.to` or `Tween.from` to set-up the tween values. You can create multiple tweens on the same object by calling Tween.to multiple times on the same Tween. Additional tweens specified in this way become “child” tweens and are played through in sequence. You can use Tween.timeScale and Tween.reverse to control the playback of this Tween and all of its children.

Time

Tilemaps

Math

Network

Particles

Physics

Arcade Physics

Ninja Physics

P2 Physics

Input

Gamepads

Tweens

Sound

System

Utils

Docs Search

Parameters

| Name                 | Type                | Description                                                        |
|----------------------|---------------------|--------------------------------------------------------------------|
| <code>target</code>  | object              | The target object, such as a Phaser.Sprite or Phaser.Sprite.scale. |
| <code>game</code>    | Phaser.Game         | Current game instance.                                             |
| <code>manager</code> | Phaser.TweenManager | The TweenManager responsible for looking after this Tween.         |

Source code: [tween/Tween.js](#) (Line 20)

Public Properties

`chainedTween`

`current`

`game`

`isPaused`

`isRunning`

`manager`

`onChildComplete`

`onComplete`

`onLoop`

`onRepeat`

`onStart`

`pendingDelete`

`properties`

`repeatCounter`

`reverse`

`target`

`timeline`

`timeScale`

`totalDuration`

**chainedTween** : [Phaser.Tween](#)

If this Tween is chained to another this holds a reference to it.

Source code: [tween/Tween.js](#) (Line 129)

[readonly] **current** : [number](#)

The current Tween child being run.

**Default Value**

☐ 0

Source code: [tween/Tween.js](#) (Line 119)

**game** : [Phaser.Game](#)

A reference to the currently running Game.

Source code: [tween/Tween.js](#) (Line 25)

**isPaused** : [boolean](#)

Is this Tween paused or not?

**Default Value**

☐ false

Source code: [tween/Tween.js](#) (Line 135)

**isRunning** : [boolean](#)

If the tween is running this is set to true, otherwise false. Tweens that are in a delayed state or waiting to start are considered as being running.

**Default Value**

☐ false

Source code: [tween/Tween.js](#) (Line 112)

**manager** : [Phaser.TweenManager](#)

Reference to the TweenManager responsible for updating this Tween.

Source code: [tween/Tween.js](#) (Line 35)

**onChildComplete** : [Phaser.Signal](#)

The onChildComplete event is fired when the Tween or any of its children completes. Fires every time a child completes unless a child is set to repeat forever. It will be sent 2 parameters: the target object and this tween.

Source code: [tween/Tween.js](#) (Line 99)

**onComplete** : [Phaser.Signal](#)

The onComplete event is fired when the Tween and all of its children completes. Does not fire if the Tween is set to loop or repeatAll(-1). It will be sent 2 parameters: the target object and this tween.

Source code: [tween/Tween.js](#) (Line 106)

### **onLoop** : [Phaser.Signal](#)

The onLoop event is fired if the Tween or any child tween loops. It will be sent 2 parameters: the target object and this tween.

Source code: [tween/Tween.js](#) (Line 84)

### **onRepeat** : [Phaser.Signal](#)

The onRepeat event is fired if the Tween and all of its children repeats. If this tween has no children this will never be fired. It will be sent 2 parameters: the target object and this tween.

Source code: [tween/Tween.js](#) (Line 91)

### **onStart** : [Phaser.Signal](#)

The onStart event is fired when the Tween begins. If there is a delay before the tween starts then onStart fires after the delay is finished. It will be sent 2 parameters: the target object and this tween.

Source code: [tween/Tween.js](#) (Line 77)

### [\[readonly\]](#) **pendingDelete** : boolean

True if this Tween is ready to be deleted by the TweenManager.

### Default Value

☐ false

Source code: [tween/Tween.js](#) (Line 70)

### properties : Object

Target property cache used when building the child data values.

Source code: [tween/Tween.js](#) (Line 124)

### repeatCounter : number

If the Tween and any child tweens are set to repeat this contains the current repeat count.

Source code: [tween/Tween.js](#) (Line 63)

### reverse : boolean

If set to `true` the current tween will play in reverse. If the tween hasn't yet started this has no effect. If there are child tweens then all child tweens will play in reverse from the current point.

### Default Value

☐ false

Source code: [tween/Tween.js](#) (Line 49)

### target : Object

The target object, such as a Phaser.Sprite or property like Phaser.Sprite.scale.

Source code: [tween/Tween.js](#) (Line 30)

### **timeline** : Array

An Array of TweenData objects that comprise the different parts of this Tween.

Source code: [tween/Tween.js](#) (Line 40)

### **timeScale** : number

The speed at which the tweens will run. A value of 1 means it will match the game frame rate. 0.5 will run at half the frame rate. 2 at double the frame rate, etc. If a tweens duration is 1 second but timeScale is 0.5 then it will take 2 seconds to complete.

#### **Default Value**

■ 1

Source code: [tween/Tween.js](#) (Line 58)

### **totalDuration** : Phaser.TweenData

Gets the total duration of this Tween, including all child tweens, in milliseconds.

Source code: [tween/Tween.js](#) (Line 863)

## Public Methods

chain delay easing from from generateData interpolation loop  
onUpdateCallback pause repeat repeat repeatDelay resume start  
stop to to update updateTweenData yoyo yoyoDelay

### `chain(tweens) → {Phaser.Tween}`

This method allows you to chain tweens together. Any tween chained to this tween will have its `Tween.start` method called as soon as this tween completes. If this tween never completes (i.e. `repeatAll` or `loop` is set) then the chain will never progress. Note that `Tween.onComplete` will fire when *this* tween completes, not when the whole chain completes. For that you should listen to `onComplete` on the final tween in your chain.

If you pass multiple tweens to this method they will be joined into a single long chain. For example if this is Tween A and you pass in B, C and D then B will be chained to A, C will be chained to B and D will be chained to C. Any previously chained tweens that may have been set will be overwritten.

### Parameters

| Name                | Type                      | Argument     | Description                                          |
|---------------------|---------------------------|--------------|------------------------------------------------------|
| <code>tweens</code> | <code>Phaser.Tween</code> | <repeatable> | One or more tweens that will be chained to this one. |



## Returns

Phaser.Tween -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 546)

---

## delay(duration, index) → {Phaser.Tween}

Sets the delay in milliseconds before this tween will start. If there are child tweens it sets the delay before the first child starts. The delay is invoked as soon as you call `Tween.start`. If the tween is already running this method doesn't do anything for the current active tween. If you have not yet called `Tween.to` or `Tween.from` at least once then this method will do nothing, as there are no tweens to delay. If you have child tweens and pass -1 as the index value it sets the delay across all of them.

## Parameters

| Name                  | Type   | Argument | Default | Description                                                                                                                         |
|-----------------------|--------|----------|---------|-------------------------------------------------------------------------------------------------------------------------------------|
| <code>duration</code> | number |          |         | The amount of time in ms that the Tween should wait until it begins once started is called. Set to zero to remove any active delay. |
|                       |        |          |         |                                                                                                                                     |

|                    |        |            |   |                                                                                                                                           |
|--------------------|--------|------------|---|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>index</code> | number | <optional> | o | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the delay on all the children. |
|--------------------|--------|------------|---|-------------------------------------------------------------------------------------------------------------------------------------------|

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 387)

## `easing(ease, index) → {Phaser.Tween}`

Set easing function this tween will use, i.e. `Phaser.Easing.Linear.None`. The ease function allows you define the rate of change. You can pass either a function such as `Phaser.Easing.Circular.Out` or a string such as "Circ". ".easeIn", ".easeOut" and "easeInOut" variants are all supported for all ease types. If you have child tweens and pass -1 as the index value it sets the easing function defined here across all of them.

## Parameters

| Name              | Type                 | Argument | Default | Description                                                                            |
|-------------------|----------------------|----------|---------|----------------------------------------------------------------------------------------|
| <code>ease</code> | function<br>  string |          |         | The easing function this tween will use, i.e. <code>Phaser.Easing.Linear.None</code> . |

|                    |        |            |   |                                                                                                                                                 |
|--------------------|--------|------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>index</code> | number | <optional> | o | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the easing function on all children. |
|--------------------|--------|------------|---|-------------------------------------------------------------------------------------------------------------------------------------------------|

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 483)

## `from(properties, duration, ease, delay, repeat, yoyo) → {Phaser.TweenData}`

Sets this tween to be a `from` tween on the properties given. A `from` tween sets the target to the destination value and tweens to its current value. For example a Sprite with an `x` coordinate of 100 tweened from `x` 500 would be set to `x` 500 and then tweened to `x` 100 by giving a properties object of `{ x: 500 }`.

## Parameters

| Name                    | Type   | Argument | Default | Description                                                        |
|-------------------------|--------|----------|---------|--------------------------------------------------------------------|
| <code>properties</code> | object |          |         | The properties you want to tween, such as <code>Sprite.x</code> or |

|  |                       |          |            |       |                                                                                                                                                                                       |
|--|-----------------------|----------|------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                       |          |            |       | <code>Sound.volume</code> . Given as a JavaScript object.                                                                                                                             |
|  | <code>duration</code> | number   | <optional> | 1000  | Duration of this tween in ms.                                                                                                                                                         |
|  | <code>ease</code>     | function | <optional> | null  | Easing function. If not set it will default to <code>Phaser.Easing.Default</code> , which is <code>Phaser.Easing.Linear.None</code> by default but can be over-ridden at will.        |
|  | <code>delay</code>    | number   | <optional> | 0     | Delay before this tween will start, defaults to 0 (no delay). Value given is in ms.                                                                                                   |
|  | <code>repeat</code>   | number   | <optional> | 0     | Should the tween automatically restart once complete? If you want it to run forever set as -1. This ignores any chained tweens.                                                       |
|  | <code>yoyo</code>     | boolean  | <optional> | false | A tween that yoyos will reverse itself and play backwards automatically. A yoyo'd tween doesn't fire the <code>Tween.onComplete</code> event, so listen for <code>Tween.onLoop</code> |

instead.

## Returns

`Phaser.TweenData` -

This Tween object.

Source code: [tween/TweenData.js](#) (Line 205)

## `from(properties, duration, ease, autoStart, delay, repeat, yoyo) → {Phaser.Tween}`

Sets this tween to be a `from` tween on the properties given. A `from` tween sets the target to the destination value and tweens to its current value. For example a Sprite with an `x` coordinate of 100 tweened from `x` 500 would be set to `x` 500 and then tweened to `x` 100 by giving a properties object of `{ x: 500 }`. The ease function allows you define the rate of change. You can pass either a function such as `Phaser.Easing.Circular.Out` or a string such as `"Circ"`. `".easeIn"`, `".easeOut"` and `".easeInOut"` variants are all supported for all ease types.

## Parameters

| Name                    | Type   | Argument | Default | Description                                             |
|-------------------------|--------|----------|---------|---------------------------------------------------------|
| <code>properties</code> | object |          |         | An object containing the properties you want to tween., |

|  |                        |                      |            |       |                                                                                                                                                                        |
|--|------------------------|----------------------|------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                        |                      |            |       | such as <code>Sprite.x</code> or <code>Sound.volume</code> . Given as a JavaScript object.                                                                             |
|  | <code>duration</code>  | number               | <optional> | 1000  | Duration of this tween in ms.                                                                                                                                          |
|  | <code>ease</code>      | function<br>  string | <optional> | null  | Easing function. If not set it will default to <code>Phaser.Easing.Default</code> , which is <code>Phaser.Easing.Linear.None</code> by default but can be over-ridden. |
|  | <code>autoStart</code> | boolean              | <optional> | false | Set to <code>true</code> to allow this tween to start automatically. Otherwise call <code>Tween.start()</code> .                                                       |
|  | <code>delay</code>     | number               | <optional> | 0     | Delay before this tween will start in milliseconds. Defaults to 0, no delay.                                                                                           |
|  | <code>repeat</code>    | number               | <optional> | 0     | Should the tween automatically restart once complete? If you want it to run forever set as -1. This only effects this individual tween, not any chained tweens.        |
|  | <code>yoyo</code>      | boolean              | <optional> | false | A tween that yoyos will reverse                                                                                                                                        |

itself and play backwards automatically. A yoyo'd tween doesn't fire the Tween.onComplete event, so listen for Tween.onLoop instead.

## Returns

Phaser.Tween -

This Tween object.

Source code: [tween/Tween.js](#) (Line 215)

---

## generateData(frameRate, data) → {array}

This will generate an array populated with the tweened object values from start to end. It works by running the tween simulation at the given frame rate based on the values set-up in Tween.to and Tween.from. It ignores delay and repeat counts and any chained tweens, but does include child tweens. Just one play through of the tween data is returned, including yoyo if set.

## Parameters

| Name                   | Type   | Argument   | Default | Description                    |
|------------------------|--------|------------|---------|--------------------------------|
| <code>frameRate</code> | number | <optional> | 60      | The speed in frames per second |

|                   |       |            |  |                                                                                                     |
|-------------------|-------|------------|--|-----------------------------------------------------------------------------------------------------|
|                   |       |            |  | that the data should be generated at. The higher the value, the larger the array it creates.        |
| <code>data</code> | array | <optional> |  | If given the generated data will be appended to this array, otherwise a new array will be returned. |

## Returns

array -

An array of tweened values.

Source code: [tween/Tween.js](#) (Line 811)

---

## **interpolation(interpolation, context, index) → {Phaser.Tween}**

Sets the interpolation function the tween will use. By default it uses `Phaser.Math.linearInterpolation`. Also available: `Phaser.Math.bezierInterpolation` and `Phaser.Math.catmullRomInterpolation`. The interpolation function is only used if the target properties is an array. If you have child tweens and pass -1 as the index value and it will set the interpolation function across all of them.

## Parameters



| Name                       | Type     | Argument   | Default | Description                                                                                                                                            |
|----------------------------|----------|------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>interpolation</code> | function |            |         | The interpolation function to use<br>(Phaser.Math.linearInterpolation by default)                                                                      |
| <code>context</code>       | object   | <optional> |         | The context under which the interpolation function will be run.                                                                                        |
| <code>index</code>         | number   | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the interpolation function on all children. |

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 505)

---

**loop(value) → {`Phaser.Tween`}**

Enables the looping of this tween and all child tweens. If this tween has no children this setting has no effect. If `value` is `true` then this is the same as setting `Tween.repeatAll(-1)`. If `value` is `false` it is the same as setting `Tween.repeatAll(0)` and will reset the `repeatCounter` to zero.

Usage: `game.add.tween(p).to({ x: 700 }, 1000, Phaser.Easing.Linear.None, true).to({ y: 300 }, 1000, Phaser.Easing.Linear.None).to({ x: 0 }, 1000, Phaser.Easing.Linear.None).to({ y: 0 }, 1000, Phaser.Easing.Linear.None).loop();`

## Parameters

| Name               | Type    | Argument   | Default | Description                                                                                                                                 |
|--------------------|---------|------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>value</code> | boolean | <optional> | true    | If <code>true</code> this tween and any child tweens will loop once they reach the end. Set to <code>false</code> to remove an active loop. |

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 580)

---

**onUpdateCallback(callback, callbackContext) → {`Phaser.Tween`}**

Sets a callback to be fired each time this tween updates.

## Parameters

| Name                         | Type     | Description                                                                                                            |
|------------------------------|----------|------------------------------------------------------------------------------------------------------------------------|
| <code>callback</code>        | function | The callback to invoke each time this tween is updated. Set to <code>null</code> to remove an already active callback. |
| <code>callbackContext</code> | object   | The context in which to call the onUpdate callback.                                                                    |

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 612)

---

## `pause()`

Pauses the tween. Resume playback with `Tween.resume`.

Source code: [tween/Tween.js](#) (Line 629)

---

## **repeat(total) → {Phaser.Tween}**

Set how many times this tween and all of its children will repeat. A tween (A) with 3 children (B,C,D) with a `repeatAll` value of 2 would play as: ABCDABCD before completing. When all child tweens have completed Tween.onLoop will be dispatched.

### **Parameters**

| Name               | Type   | Description                                                                                                                                      |
|--------------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>total</code> | number | How many times this tween and all children should repeat before completing. Set to zero to remove an active repeat. Set to -1 to repeat forever. |

### **Returns**

Phaser.Tween -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 527)

---

## **repeat(total, repeat, index) → {Phaser.Tween}**

Sets the number of times this tween will repeat. If you have not yet called `Tween.to` or `Tween.from` at least once then this method will do nothing, as there are no tweens to repeat. If you have child tweens and pass -1 as the index value it sets the number of times they'll repeat across

all of them. If you wish to define how many times this Tween and all children will repeat see Tween.repeatAll.

## Parameters

| Name                | Type   | Argument   | Default | Description                                                                                                                                      |
|---------------------|--------|------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>total</code>  | number |            |         | How many times a tween should repeat before completing. Set to zero to remove an active repeat. Set to -1 to repeat forever.                     |
| <code>repeat</code> | number | <optional> | 0       | This is the amount of time to pause (in ms) before the repeat will start.                                                                        |
| <code>index</code>  | number | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the repeat value on all the children. |

## Returns

[Phaser.Tween](#) -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 404)

## repeatDelay(duration, index) → {Phaser.Tween}

Sets the delay in milliseconds before this tween will repeat itself. The repeatDelay is invoked as soon as you call `Tween.start`. If the tween is already running this method doesn't do anything for the current active tween. If you have not yet called `Tween.to` or `Tween.from` at least once then this method will do nothing, as there are no tweens to set repeatDelay on. If you have child tweens and pass -1 as the index value it sets the repeatDelay across all of them.

### Parameters

| Name                  | Type   | Argument   | Default | Description                                                                                                                                       |
|-----------------------|--------|------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>duration</code> | number |            |         | The amount of time in ms that the Tween should wait until it repeats or yoyos once start is called. Set to zero to remove any active repeatDelay. |
| <code>index</code>    | number | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the repeatDelay on all the children.   |

### Returns

Phaser.Tween -

This tween. Useful for method chaining.

Source code: [tween/Tween.js \(Line 426\)](#)

---

## resume()

Resumes a paused tween.

Source code: [tween/Tween.js \(Line 661\)](#)

---

## start(index) → {Phaser.Tween}

Starts the tween running. Can also be called by the autoStart parameter of `Tween.to` or `Tween.from`. This sets the `Tween.isRunning` property to `true` and dispatches a `Tween.onStart` signal. If the Tween has a delay set then nothing will start tweening until the delay has expired.

## Parameters

| Name               | Type   | Argument   | Default | Description                                                                                                                     |
|--------------------|--------|------------|---------|---------------------------------------------------------------------------------------------------------------------------------|
| <code>index</code> | number | <optional> | 0       | If this Tween contains child tweens you can specify which one to start from. The default is zero, i.e. the first tween created. |

## Returns

Phaser.Tween -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 262)

---

## stop(complete) → {Phaser.Tween}

Stops the tween if running and flags it for deletion from the TweenManager. If called directly the `Tween.onComplete` signal is not dispatched and no chained tweens are started unless the complete parameter is set to `true`. If you just wish to pause a tween then use Tween.pause instead.

## Parameters

| Name                  | Type    | Argument   | Default | Description                                                       |
|-----------------------|---------|------------|---------|-------------------------------------------------------------------|
| <code>complete</code> | boolean | <optional> | false   | Set to <code>true</code> to dispatch the Tween.onComplete signal. |

## Returns

Phaser.Tween -

This tween. Useful for method chaining.



Source code: [tween/Tween.js](#) (Line 320)

## **to(properties, duration, ease, autoStart, delay, repeat, yoyo) → {Phaser.Tween}**

Sets this tween to be a `to` tween on the properties given. A `to` tween starts at the current value and tweens to the destination value given. For example a Sprite with an `x` coordinate of 100 could be tweened to `x` 200 by giving a properties object of `{ x: 200 }`. The ease function allows you define the rate of change. You can pass either a function such as `Phaser.Easing.Circular.Out` or a string such as `"Circ"`. `".easeIn"`, `".easeOut"` and `".easeInOut"` variants are all supported for all ease types.

### **Parameters**

| Name                    | Type                 | Argument   | Default | Description                                                                                                                                       |
|-------------------------|----------------------|------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>properties</code> | object               |            |         | An object containing the properties you want to tween, such as <code>Sprite.x</code> or <code>Sound.volume</code> . Given as a JavaScript object. |
| <code>duration</code>   | number               | <optional> | 1000    | Duration of this tween in ms.                                                                                                                     |
| <code>ease</code>       | function<br>  string | <optional> | null    | Easing function. If not set it will default to <code>Phaser.Easing.Default</code> , which is <code>Phaser.Easing.Linear.None</code> by            |

|  |                        |         |            |       |                                                                                                                                                                                                |
|--|------------------------|---------|------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                        |         |            |       | default but can be over-ridden.                                                                                                                                                                |
|  | <code>autoStart</code> | boolean | <optional> | false | Set to <code>true</code> to allow this tween to start automatically. Otherwise call <code>Tween.start()</code> .                                                                               |
|  | <code>delay</code>     | number  | <optional> | 0     | Delay before this tween will start in milliseconds. Defaults to 0, no delay.                                                                                                                   |
|  | <code>repeat</code>    | number  | <optional> | 0     | Should the tween automatically restart once complete? If you want it to run forever set as -1. This only effects this individual tween, not any chained tweens.                                |
|  | <code>yoyo</code>      | boolean | <optional> | false | A tween that yoyos will reverse itself and play backwards automatically. A yoyo'd tween doesn't fire the <code>Tween.onComplete</code> event, so listen for <code>Tween.onLoop</code> instead. |

## Returns

Phaser.Tween -

This Tween object.

Source code: [tween/Tween.js](#) (Line 168)

---

## **to(properties, duration, ease, delay, repeat, yoyo) → {Phaser.TweenData}**

Sets this tween to be a `to` tween on the properties given. A `to` tween starts at the current value and tweens to the destination value given. For example a Sprite with an `x` coordinate of 100 could be tweened to `x` 200 by giving a properties object of `{ x: 200 }`.

### **Parameters**

| Name                    | Type     | Argument   | Default | Description                                                                                                                                                           |
|-------------------------|----------|------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>properties</code> | object   |            |         | The properties you want to tween, such as <code>Sprite.x</code> or <code>Sound.volume</code> . Given as a JavaScript object.                                          |
| <code>duration</code>   | number   | <optional> | 1000    | Duration of this tween in ms.                                                                                                                                         |
| <code>ease</code>       | function | <optional> | null    | Easing function. If not set it will default to <code>Phaser.Easing.Default</code> , which is <code>Phaser.Easing.Linear.None</code> by default but can be over-ridden |

|                     |         |            |       |                                                                                                                                                                      |
|---------------------|---------|------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |         |            |       | at will.                                                                                                                                                             |
| <code>delay</code>  | number  | <optional> | 0     | Delay before this tween will start, defaults to 0 (no delay). Value given is in ms.                                                                                  |
| <code>repeat</code> | number  | <optional> | 0     | Should the tween automatically restart once complete? If you want it to run forever set as -1. This ignores any chained tweens.                                      |
| <code>yoyo</code>   | boolean | <optional> | false | A tween that yoyos will reverse itself and play backwards automatically. A yoyo'd tween doesn't fire the Tween.onComplete event, so listen for Tween.onLoop instead. |

## Returns

`Phaser.TweenData` -

This Tween object.

Source code: [tween/TweenData.js](#) (Line 177)

## **update(time) → {boolean}**

Core tween update function called by the TweenManager. Does not need to be invoked directly.

### **Parameters**

| Name              | Type   | Description                                |
|-------------------|--------|--------------------------------------------|
| <code>time</code> | number | A timestamp passed in by the TweenManager. |

### **Returns**

boolean -

false if the tween and all child tweens have completed and should be deleted from the manager, otherwise true (still active).

Source code: [tween/Tween.js](#) (Line 703)

---

## **updateTweenData(property, value, index) → {Phaser.Tween}**

Updates either a single TweenData or all TweenData objects properties to the given value. Used internally by methods like Tween.delay, Tween.yoyo, etc. but can also be called directly if you know which property you want to tweak. The property is not checked, so if you pass an invalid one you'll generate a run-time error.

## Parameters

| Name                  | Type                    | Argument   | Default | Description                                                                                                                               |
|-----------------------|-------------------------|------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>property</code> | string                  |            |         | The property to update.                                                                                                                   |
| <code>value</code>    | number<br> <br>function |            |         | The value to set the property to.                                                                                                         |
| <code>index</code>    | number                  | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the delay on all the children. |

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 354)

---

## **yoyo(enable, yoyoDelay, index) → {`Phaser.Tween`}**

A Tween that has yoyo set to true will run through from its starting values to its end values and then play back in reverse from end to start. Used in combination with repeat you can create endless loops.

If you have not yet called `Tween.to` or `Tween.from` at least once then this method will do nothing, as there are no tweens to yoyo. If you have child tweens and pass -1 as the index value it sets the yoyo property across all of them. If you wish to yoyo this Tween and all of its children then see `Tween.yoyoAll`.

## Parameters

| Name                   | Type    | Argument   | Default | Description                                                                                                                          |
|------------------------|---------|------------|---------|--------------------------------------------------------------------------------------------------------------------------------------|
| <code>enable</code>    | boolean |            |         | Set to true to yoyo this tween, or false to disable an already active yoyo.                                                          |
| <code>yoyoDelay</code> | number  | <optional> | 0       | This is the amount of time to pause (in ms) before the yoyo will start.                                                              |
| <code>index</code>     | number  | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set yoyo on all the children. |

## Returns

`Phaser.Tween` -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 443)

## **yoyoDelay(duration, index) → {Phaser.Tween}**

Sets the delay in milliseconds before this tween will run a yoyo (only applies if yoyo is enabled). The repeatDelay is invoked as soon as you call `Tween.start`. If the tween is already running this method doesn't do anything for the current active tween. If you have not yet called `Tween.to` or `Tween.from` at least once then this method will do nothing, as there are no tweens to set repeatDelay on. If you have child tweens and pass -1 as the index value it sets the repeatDelay across all of them.

### **Parameters**

| Name                  | Type   | Argument   | Default | Description                                                                                                                                     |
|-----------------------|--------|------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>duration</code> | number |            |         | The amount of time in ms that the Tween should wait until it repeats or yoyos once start is called. Set to zero to remove any active yoyoDelay. |
| <code>index</code>    | number | <optional> | 0       | If this tween has more than one child this allows you to target a specific child. If set to -1 it will set the yoyoDelay on all the children.   |



## Returns

[Phaser.Tween](#) -

This tween. Useful for method chaining.

Source code: [tween/Tween.js](#) (Line 466)



## Comments

[comments powered by Disqus](#)



**Photon Storm**

© 2015 Photon Storm Ltd.  
All rights reserved.

Get Phaser  
2.3.0  
API Docs  
HTML5 Game Dev



**Phaser  
Forums**



**@photonstorm**



**Phaser on Github**