**GreenSock Animation Platform**



* What is **GreenSock** (GSAP).
* What are **TweenLite**, **TimelineLite**, **TimelineMax** and **TweenMax** useful for.
* How to **create simple animations using TweenLite**.
* How to use **.to()**, **.from()**, **.fromTo()** methods.
* How to apply **easings to your GreenSock tweens**.
* How to use **callback functions with TweenLite**.

**TweenLite**.

TweenLite is an extremely fast, lightweight, and flexible animation tool that serves as the foundation of the GreenSock Animation Platform (GSAP), available in AS2, AS3, and JavaScript. A TweenLite instance handles tweening one or more properties of **any object** (or array of objects) over time. TweenLite can be used on its own to accomplish most animation chores with minimal file size or it can be use in conjuction with advanced sequencing tools like TimelineLite or TimelineMax to make complex tasks much simpler. With scores of other animation frameworks to choose from, why consider the GreenSock Animation Platform?:

* **SPEED**- The platform has been highly optimized for maximum performance. See some speed comparisons yourself at <http://www.greensock.com/tweening-speed-test/>
* **Freakishly robust feature set**- In addition to tweening any numeric property of any object, TweenLite has plugins that give it the ability to tween hex colors, beziers, arrays, filters, plus **LOTS** more. It can round values, use relative values, smoothly reverse() on the fly, automatically detect and accommodate getter/setter functions, employ virtually any easing equation, pause()/resume() anytime, and intelligently manage conflicting tweens of the same object with various overwrite modes. TweenMax extends TweenLite and adds even more capabilities like repeat, yoyo, repeatDelay, on-the-fly destination value updates and more.
* **Sequencing, grouping, and management features**- TimelineLite and TimelineMax make it surprisingly simple to create complex sequences or groups of tweens that you can control as a whole. play(), pause(), restart(), or reverse(). You can even tween a timeline's time or progress to fastforward or rewind the entire timeline. Add labels, change the timeline's timeScale, nest timelines within timelines, and much more. This can revolutionize your animation workflow, making it more modular and concise.
* **AS3, AS2, and JavaScript**- Most other engines are only developed for one language, but the GreenSock Animation Platform allows you to use a consistent API across all your Flash and HTML5 projects.
* **Ease of use**- Designers and Developers alike rave about how intuitive the platform is.
* **Support and reliability**- With frequent updates, [dedicated forums](http://forums.greensock.com/), committed authorship, a solid track record, a proven funding mechansim, and a thriving community of users, the platform is a safe long-term bet (unlike many open source projects).
* **Expandability**- With its plugin architecture, you can activate as many (or as few) extra features as your project requires. Write your own plugin to handle particular special properties in custom ways. Minimize bloat and maximize performance.

# Create Your First TweenLite Animation

## Add a simple script to the html document

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <script>  window.onload = function(){      var logo = document.getElementById("logo");      TweenLite.to(logo, 1, {left:"632px"});  }  </script> |

Example

Add Easing

Easing allows you to change the speed of your animation over time

|  |  |
| --- | --- |
| 1 | TweenLite.to(logo,1, {left:"440px", ease:Linear.easeNone}); |

# Delay

## Specify how long a tween should wait before playing

|  |  |
| --- | --- |
| 1  2 | TweenLite.to(bar, 2, {width:"692px"});  TweenLite.to(logo, 2, {left:"632px", delay:1}); |

Animate Multiple Properties With One Tween

Below is an example of four properties being animated with one tween

|  |  |
| --- | --- |
| 1  2  3  4 | TweenLite.to(logo, 2, {left:"542px",                         backgroundColor:"black",                         borderBottomColor:"#90e500",                         color:"white"}); |

From Tweens

Animate a property from a given value to its current value

|  |  |
| --- | --- |
| 1 | TweenLite.from(logo, 2, {opacity:0, left:"300px"}); |

Tween to Relative Values

Set end values in relationship to existing values

|  |  |
| --- | --- |
| 1 | TweenLite.to(logo, 0.5, {left:"+=100px"}); |

Animate Multiple Targets With One Tween

TweenLite can accept an Array of objects as the tween's target. Cool!

|  |  |
| --- | --- |
| 1  2  3  4 | var red = document.getElementById("red");  var yellow = document.getElementById("yellow");  var green = document.getElementById("green");  TweenLite.to([red, yellow, green], 1, {scale:0.2, opacity:0.3}); |

Use jQuery with GSAP JS

For convenience use jQuery to select single or multiple elements to animate.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!-- HTML -->  <div id="red" class="box"></div>  <div id="yellow" class="box"></div>  <div id="green" class="box"></div>  // Use jQuery to select all elements that use the .box class  TweenLite.to($(".box"), 1, {scale:0.2, opacity:0.3}); |

Event Callbacks

Robust callback system allows you to respond to many animation events

|  |  |
| --- | --- |
| 1  2  3 | TweenLite.to(logo, 2, {left:"300px", onUpdate:updateHandler,  onComplete:completeHandler,  onCompleteParams:["animation complete!"]}); |

Reference A Tween

Access a tween's methods by referencing it with a variable.

|  |  |
| --- | --- |
| 1  2  3  4  5 | var tween = TweenLite.to(logo, 1, {left:"632px"});  restartButton.onclick = function(){      //call the tween's restart() method      tween.restart();  } |

Control Playback

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | var tween = TweenLite.to(logo, 6, {left:"632px",                           ease:Linear.easeNone});  tween.reverse(1); //reverse first second of animation  //timeScale of 2 will make tween play at double-speed (faster)  //tween will take 3 seconds to complete  tween.timeScale(2);  Macintosh HD:Users:apple:Desktop:Screen Shot 2016-03-29 at 6.24.52 PM.png  Macintosh HD:Users:apple:Desktop:Screen Shot 2016-03-29 at 6.26.16 PM.png  Repeat a Tween any Number of Times  TweenMax allows you to set repeat, repeatDelay and yoyo properties   |  |  | | --- | --- | | 1  2 | //notice that the repeat value is outside the css {} object  TweenMax.to(logo, 1, {left:"300px", repeat:1}); |   Stagger Multiple Animations  TweenMax's staggerTo() and staggerFrom() methods generate multiple animations with staggered start times.   |  |  | | --- | --- | | 1  2 | TweenMax.staggerTo([red, yellow, green, blue, pink, purple], 1,                     {scale:0.2, opacity:0.3}, 0.25); |   Macintosh HD:Users:apple:Desktop:Screen Shot 2016-03-29 at 6.28.21 PM.png  TimelineLite Basic Syntax  Clean code that packs a powerful punch   |  |  | | --- | --- | | 1  2  3  4 | var tl = new TimelineLite({paused:true});  tl.from(logo, 0.5, {left:"-=60px", ease:Back.easeOut})    .from(timelineLite, 0.5, {width:"0px", alpha:0}, "-=0.2")    .staggerFrom(tagline, 0.5, {top:"-=30px", rotation:"-40deg", alpha:0, scale:1.8,  ease:Back.easeOut}, 0.2); |  TimelineLite ControlTimelineLite uses the same familiar control methods as TweenLite //resume playback in current direction  tl.resume();  //timeScale set to default speed  tl.timeScale(1);  Macintosh HD:Users:apple:Desktop:Screen Shot 2016-03-29 at 6.30.39 PM.png  TimelineLite Labels  Add labels to a timeline for easy and intuitive navigation   |  |  | | --- | --- | | 1  2  3  4 | tl.add("skew") // adds a new label    .add(getSkewAnimation()) // method returns a TimelineLite instance that gets nested at the end    .add(getStaggerAnimation(), "stagger") //creates new label and adds animation there    .add(getParticlesAnimation(), "particles") |   Macintosh HD:Users:apple:Desktop:Screen Shot 2016-03-29 at 6.33.37 PM.png  TimelineMax in Action   |  |  | | --- | --- | | 1  2  3  4  5 | var tl = new TimelineMax({onUpdate:updateUI, repeat:2, repeatDelay:1, yoyo:true});  tl.from(logo, 0.5, {left:'-=60px', ease:Back.easeOut})    .staggerFrom(txt, 0.1, {alpha:0}, 0.02, "textEffect")    .staggerFrom(txt, 0.8, {rotationY:"-270deg", top:"100px", transformOrigin: "50% 50% -80", ease:Back.easeOut}, 0.02, "textEffect")    .staggerTo(txt, 0.6, {rotationX:"+=360deg", transformOrigin:"50% 50% 10", color:"#90e500"}, 0.02); | |