



– Developed by Dented Pixel

API Docs for: LeanTween 2.20

Support
LeanTween!



APIs

Classes | Modules

Type to filter APIs

LeanTween
LeanTweenType
LTBezierPath
LTDscr
LTEvent
LTRect
LTSpline

Games Developed by Dented Pixel



Carbon Cutters – A game that
activates
real world change.



RIN – A swinging good time!



Princess Piano – Learn musical
notation
in this melodious adventure!

LeanTween Class

Show: ☒ Inherited ☐ Protected ☐ Private ☐ Deprecated

Defined in: LeanTween.cs:1554

LeanTween is an efficient tweening engine for Unity3d

[Index of All Methods](#) | [Optional Paramaters that can be passed](#)

Optional Parameters are passed at the end of every method

Example:

```
LeanTween.moveX( gameObject, 1f, 1f).setEase( LeanTweenType.easeInQuad ).setDelay(1f);
```

You can pass the optional parameters in any order, and chain on as many as you wish.
You can also pass parameters at a later time by saving a reference to what is returned.

Example:

```
LTDscr d = LeanTween.moveX(gameObject, 1f, 1f);  
...later set some parameters  
d.setOnComplete( onCompleteFunc ).setEase( LeanTweenType.easeInOutBack );
```

Index | Methods

Methods

LeanTween.addListener (caller:GameObject , eventId:int , callback:System.Action<LTEvent>)
Defined in LeanTween.cs:4400

Add a listener method to be called when the appropriate LeanTween.dispatchEvent is called

Parameters:

- **caller:GameObject** *GameObject*
the gameObject the listener is attached to
- **eventId:int** *Int*
a unique int that describes the event (best to use an enum)
- **callback:System.Action<LTEvent>** *System.Action*
the method to call when the event has been dispatched

Example:

```
LeanTween.addListener(gameObject, (int)MyEvents.JUMP, jumpUp);  
  
void jumpUp( LTEvent e ){ Debug.Log("jump!"); }
```

LeanTween.alpha (gameObject:GameObject , to:float , time:float) *LTDscr*
Defined in LeanTween.cs:2844

Fade a gameObject's material to a certain alpha value. The material's shader needs to support alpha. [Owl labs has some excellent efficient shaders.](#)

Parameters:

- **gameObject:GameObject** *GameObject*
Gameobject that you wish to fade
- **to:float** *Float*
the final alpha value (0–1)
- **time:float** *Float*
The time with which to fade the object

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

Example:

```
LeanTween.alpha(gameObject, 1f, 1f).setDelay(1f);
```

LeanTween.alpha (ltRect:LRect , to:float , time:float) [LTDscr](#)
Defined in [LeanTween.cs:2859](#)

Fade a GUI Object

Parameters:

- **ltRect:LRect** [LRect](#)
LRect that you wish to fade
- **to:float** Float
the final alpha value (0-1)
- **time:float** Float
The time with which to fade the object

Returns:

[LTDscr](#):

LTDscr an object that distinguishes the tween

Example:

LeanTween.alpha(ltRect, 1f, 1f).setEase(LeanTweenType.easeInCirc);

LeanTween.alphaVertex (gameObject:GameObject , to:float , time:float) [LTDscr](#)
Defined in [LeanTween.cs:2881](#)

This works by tweening the vertex colors directly.

Vertex-based coloring is useful because you avoid making a copy of your object's material for each instance that needs a different color.

A shader that supports vertex colors is required for it to work (for example the shaders in Mobile/Particles/)

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to alpha
- **to:float** Float
The alpha value you wish to tween to
- **time:float** Float
The time with which to delay before calling the function

Returns:

[LTDscr](#):

LTDscr an object that distinguishes the tween

LeanTween.cancel (ltRect:LRect , id:int)
Defined in [LeanTween.cs:2593](#)

Cancel a specific tween with the provided id

Parameters:

- **ltRect:LRect** [LRect](#)
LRect object whose tweens you want to cancel
 - **id:int** Float
unique id that represents that tween
-

LeanTween.cancel (gameObject:GameObject)
Defined in [LeanTween.cs:2557](#)

Cancel all tweens that are currently targeting the gameObject

Parameters:

- **gameObject:GameObject** [GameObject](#)
gameObject whose tweens you wish to cancel

Example:

LeanTween.move(gameObject, new Vector3(0f,1f,2f), 1f);
LeanTween.cancel(gameObject);

LeanTween.cancel (gameObject:GameObject , id:int)
Defined in [LeanTween.cs:2575](#)

Cancel a specific tween with the provided id

Parameters:

- **gameObject:GameObject** GameObject
gameObject whose tweens you want to cancel
- **id:int** Float
unique id that represents that tween

LeanTween.cancelAll (callComplete:bool)
Defined in [LeanTween.cs:2537](#)

Cancels all tweens

Parameters:

- **callComplete:bool** CallComplete
if true, then the onComplete event will be fired if it exists

Example:

LeanTween.cancelAll(true);

LeanTween.color (gameObject:GameObject , to:Color , time:float) [LTDescr](#)
Defined in [LeanTween.cs:2900](#)

Change a gameobject's material to a certain color value. The material's shader needs to support color tinting. [Owl labs has some excellent efficient shaders.](#)

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to change the color
- **to:Color** Color
the final color value ex: Color.Red, new Color(1.0f,1.0f,0.0f,0.8f)
- **time:float** Float
The time with which to fade the object

Returns:

[LTDescr](#):

LTDescr an object that distinguishes the tween

Example:

LeanTween.color(gameObject, Color.yellow, 1f).setDelay(1f);

LeanTween.dispatchEvent (eventId:int)
Defined in [LeanTween.cs:4476](#)

Tell the added listeners that you are dispatching the event

Parameters:

- **eventId:int** Int
a unique int that describes the event (best to use an enum)

Example:

LeanTween.dispatchEvent((int)MyEvents.JUMP);

LeanTween.dispatchEvent (eventId:int , data:object)
Defined in [LeanTween.cs:4487](#)

Tell the added listeners that you are dispatching the event

Parameters:

- **eventId:int** Int
a unique int that describes the event (best to use an enum)
- **data:object** Object
Pass data to the listener, access it from the listener with *.data on the LTEvent object

Example:

```
LeanTween.dispatchEvent( (int)MyEvents.JUMP, transform );
```

```
void jumpUp( LEvent e ){  
    Transform tran = (Transform)e.data;  
}
```

LeanTween.init (maxSimultaneousTweens:int)
Defined in [LeanTween.cs:1605](#)

This line is optional. Here you can specify the maximum number of tweens you will use (the default is 400). This must be called before any use of LeanTween is made for it to be effective.

Parameters:

- **maxSimultaneousTweens:int** Integer
The maximum number of tweens you will use, make sure you don't go over this limit, otherwise the code will throw an error

Example:

```
LeanTween.init( 800 );
```

LeanTween.isTweening (ltRect:LRect)
Defined in [LeanTween.cs:2757](#)

Test whether or not a tween is active on a LRect

Parameters:

- **ltRect:LRect** [LRect](#)
LRect that you want to test if it is tweening

LeanTween.isTweening (id:int)
Defined in [LeanTween.cs:2736](#)

Test whether or not a tween is active or not

Parameters:

- **id:int** [GameObject](#)
id of the tween that you want to test if it is tweening *Example:*
int id = LeanTween.moveX(gameObject, 1f, 3f).id;
if(LeanTween.isTweening(id))
 Debug.Log("I am tweening!");

LeanTween.isTweening (gameObject:GameObject)
Defined in [LeanTween.cs:2721](#)

Test whether or not a tween is active on a GameObject

Parameters:

- **gameObject:GameObject** [GameObject](#)
GameObject that you want to test if it is tweening

LeanTween.move (gameObject:GameObject , path:Vector3[] , time:float) [LTDescr](#)
Defined in [LeanTween.cs:2963](#)

Move a GameObject along a set of bezier curves

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to move
- **path:Vector3[]** [Vector3](#)
A set of points that define the curve(s) ex: Point1,Handle2,Handle1,Point2,...
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDescr](#):

LTDescr an object that distinguishes the tween

Example:

Javascript:

```
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0)
.setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

C#:

```
LeanTween.move(gameObject, new Vector3[]{new Vector3(0f,0f,0f),new Vector3(1f,0f,0f),new
Vector3(1f,0f,0f),new Vector3(1f,0f,1f)}, 1.5f).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);;
```

LeanTween.move ([GameObject](#) , [vec:Vector3](#) , [time:float](#)) [LTDscr](#)
Defined in [LeanTween.cs:2945](#)

Move a GameObject to a certain location

Parameters:

- **GameObject** [GameObject](#)
gameObject Gameobject that you wish to move
- **vec:Vector3** [Vector3](#)
to The final positin with which to move to
- **time:float** [Float](#)
time The time to complete the tween in

Returns:

[LTDscr](#):

LTDscr an object that distinguishes the tween

Example:

```
LeanTween.move(gameObject, new Vector3(0f,-3f,5f), 2.0f).setEase(LeanTweenType.easeOutQuad);
```

LeanTween.move (GUI) ([ltRect:LRect](#) , [vec:Vector2](#) , [time:float](#)) [LTDscr](#)
Defined in [LeanTween.cs:3029](#)

Move a GUI Element to a certain location

Parameters:

- **ltRect:LRect** [LRect](#)
ltRect LRect object that you wish to move
- **vec:Vector2** [Vector2](#)
to The final position with which to move to (pixel coordinates)
- **time:float** [Float](#)
time The time to complete the tween in

Returns:

[LTDscr](#):

LTDscr an object that distinguishes the tween

LeanTween.move (RectTransform) ([rectTrans:RectTransform](#) , [to:Vector3](#) , [time:float](#)) [LTDscr](#)
Defined in [LeanTween.cs:3505](#)

Move a RectTransform object (used in Unity GUI in 4.6+, for Buttons, Panel, Scrollbar, etc...)

Parameters:

- **rectTrans:RectTransform** [RectTransform](#)
RectTransform that you wish to attach the tween to
- **to:Vector3** [Vector3](#)
The final Vector3 with which to tween to
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDscr](#):

LTDscr an object that distinguishes the tween

Example:

```
LeanTween.move(button, new Vector3(200f,-100f,0f), 1f).setDelay(1f);
```

LeanTween.moveLocal (GameObject , Vector3 , float , Hashtable) [LTDScr](#)
Defined in [LeanTween.cs:3085](#)

Move a GameObject to a certain location relative to the parent transform.

Parameters:

- **GameObject** [GameObject](#)
gameObject Gameobject that you wish to rotate
- **Vector3** [Vector3](#)
to The final positin with which to move to
- **float** [Float](#)
time The time to complete the tween in
- **Hashtable** [Hashtable](#)
optional Hashtable where you can pass [optional items](#).

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

LeanTween.moveLocal (gameObject:GameObject , path:Vector3[] , time:float) [LTDScr](#)
Defined in [LeanTween.cs:3099](#)

Move a GameObject along a set of bezier curves, in local space

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to move
- **path:Vector3[]** [Vector3](#)
A set of points that define the curve(s) ex: Point1,Handle1,Handle2,Point2,...
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

Javascript:

```
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

C#:

```
LeanTween.move(gameObject, new Vector3[]  
{Vector3(0f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,1f)}).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

LeanTween.moveSpline (gameObject:GameObject , path:Vector3[] , time:float) [LTDScr](#)
Defined in [LeanTween.cs:2987](#)

Move a GameObject through a set of points

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to move
- **path:Vector3[]** [Vector3](#)
A set of points that define the curve(s) ex: ControlStart,Pt1,Pt2,Pt3,... ..ControlEnd
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

Javascript:

```
LeanTween.moveSpline(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0)  
.setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

C#:

```
LeanTween.moveSpline(gameObject, new Vector3[]{new Vector3(0f,0f,0f),new Vector3(1f,0f,0f),new  
Vector3(1f,0f,0f),new Vector3(1f,0f,1f)}, 2.0f).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

```
vector3(11,01,01),new vector3(11,01,11)), 1.5f).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

LeanTween.moveSplineLocal (gameObject:GameObject , path:Vector3[] , time:float) [LTD descr](#)
Defined in [LeanTween.cs:3008](#)

Move a GameObject through a set of points, in local space

Parameters:

- **gameObject:GameObject** [GameObject](#)
GameObject that you wish to move
- **path:Vector3[]** [Vector3](#)
A set of points that define the curve(s) ex: ControlStart,Pt1,Pt2,Pt3,...ControlEnd
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTD descr](#):

LTD descr an object that distinguishes the tween

Example:

Javascript:

```
LeanTween.moveSpline(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0)  
.setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

C#:

```
LeanTween.moveSpline(gameObject, new Vector3[]{new Vector3(0f,0f,0f),new Vector3(1f,0f,0f),new  
Vector3(1f,0f,0f),new Vector3(1f,0f,1f)}, 1.5f).setEase(LeanTweenType.easeOutQuad).setOrientToPath(true);
```

LeanTween.moveX (gameObject:GameObject , to:float , time:float) [LTD descr](#)
Defined in [LeanTween.cs:3046](#)

Move a GameObject along the x-axis

Parameters:

- **gameObject:GameObject** [GameObject](#)
gameObject GameObject that you wish to move
- **to:float** [Float](#)
to The final position with which to move to
- **time:float** [Float](#)
time The time to complete the move in

Returns:

[LTD descr](#):

LTD descr an object that distinguishes the tween

LeanTween.moveY (GameObject , float , float) [LTD descr](#)
Defined in [LeanTween.cs:3059](#)

Move a GameObject along the y-axis

Parameters:

- **GameObject** [GameObject](#)
gameObject GameObject that you wish to move
- **float** [Float](#)
to The final position with which to move to
- **float** [Float](#)
time The time to complete the move in

Returns:

[LTD descr](#):

LTD descr an object that distinguishes the tween

LeanTween.moveZ (GameObject , float , float) [LTD descr](#)
Defined in [LeanTween.cs:3072](#)

Move a GameObject along the z-axis

Parameters:

- **GameObject** GameObject
gameObject GameObject that you wish to move
- **float** Float
to The final position with which to move to
- **float** Float
time The time to complete the move in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.pause (gameObject:GameObject)

Defined in [LeanTween.cs:2649](#)

Pause all tweens for a GameObject

Parameters:

- **gameObject:GameObject** GameObject
GameObject whose tweens you want to pause
-

LeanTween.pauseAll ()

Defined in [LeanTween.cs:2664](#)

Pause all active tweens

LeanTween.removeListener (caller:GameObject , eventId:int , callback:System.Action<LTEvent>)

Defined in [LeanTween.cs:4449](#)

Remove an event listener you have added

Parameters:

- **caller:GameObject** GameObject
the gameObject the listener is attached to
- **eventId:int** Int
a unique int that describes the event (best to use an enum)
- **callback:System.Action<LTEvent>** System.Action
the method that was specified to call when the event has been dispatched

Example:

```
LeanTween.removeListener(gameObject, (int)MyEvents.JUMP, jumpUp);
```

```
void jumpUp( LTEvent e ){ }
```

LeanTween.resume (gameObject:GameObject)

Defined in [LeanTween.cs:2707](#)

Resume all the tweens on a GameObject

Parameters:

- **gameObject:GameObject** GameObject
GameObject whose tweens you want to resume
-

LeanTween.resume (id:int)

Defined in [LeanTween.cs:2693](#)

Resume a specific tween

Parameters:

- **id:int** Int
Id of the tween you want to resume ex: int id = LeanTween.MoveX(gameObject, 5, 1.0).id;
-

LeanTween.resumeAll ()

Defined in [LeanTween.cs:2676](#)

Resume all active tweens

LeanTween.rotate ([ltRect:LRect](#) , [to:float](#) , [time:float](#) , [optional:Array](#)) [LTDScr](#)
Defined in [LeanTween.cs:3150](#)

Rotate a GUI element (using an LRect object), to a value that is in degrees

Parameters:

- **ltRect:LRect** [LRect](#)
LRect that you wish to rotate
- **to:float** [Float](#)
The final rotation with which to rotate to
- **time:float** [Float](#)
The time to complete the tween in
- **optional:Array** [Array](#)
Object Array where you can pass [optional items](#).

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

```
if(GUI.Button(buttonRect.rect, "Rotate"))  
LeanTween.rotate( buttonRect4, 150.0f, 1.0f).setEase(LeanTweenType.easeOutElastic);  
GUI.matrix = Matrix4x4.identity;
```

LeanTween.rotate ([GameObject](#) , [Vector3](#) , [float](#)) [LTDScr](#)
Defined in [LeanTween.cs:3135](#)

Rotate a GameObject, to values are in passed in degrees

Parameters:

- **GameObject** [GameObject](#)
gameObject Gameobject that you wish to rotate
- **Vector3** [Vector3](#)
to The final rotation with which to rotate to
- **float** [Float](#)
time The time to complete the tween in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

```
LeanTween.rotate(cube, new Vector3(180f,30f,0f), 1.5f);
```

LeanTween.rotate (RectTransform) ([rectTrans:RectTransform](#) , [to:float](#) , [time:float](#)) [LTDScr](#)
Defined in [LeanTween.cs:3519](#)

Rotate a RectTransform object (used in Unity GUI in 4.6+, for Buttons, Panel, Scrollbar, etc...)

Parameters:

- **rectTrans:RectTransform** [RectTransform](#)
RectTransform that you wish to attach the tween to
- **to:float** [Float](#)
The degree with which to rotate the RectTransform
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

```
LeanTween.rotate(button, 90f, 1f).setDelay(1f);
```

LeanTween.rotateAround (gameObject:GameObject , vec:Vector3 , degrees:float , time:float) [LTDScr](#)
Defined in [LeanTween.cs:3220](#)

Rotate a GameObject around a certain Axis (the best method to use when you want to rotate beyond 180 degrees)

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to rotate
- **vec:Vector3** [Vector3](#)
axis in which to rotate around ex: Vector3.up
- **degrees:float** [Float](#)
the degrees in which to rotate
- **time:float** [Float](#)
time The time to complete the rotation in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

Example:

LeanTween.rotateAround (gameObject, Vector3.left, 90f, 1f);

LeanTween.rotateAround (RectTransform) (rectTrans:RectTransform , axis:Vector3 , to:float , time:float) [LTDScr](#)
Defined in [LeanTween.cs:3533](#)

Rotate a RectTransform object (used in Unity GUI in 4.6+, for Buttons, Panel, Scrollbar, etc...)

Parameters:

- **rectTrans:RectTransform** [RectTransform](#)
RectTransform that you wish to attach the tween to
- **axis:Vector3** [Vector3](#)
The axis in which to rotate the RectTransform (Vector3.forward is most commonly used)
- **to:float** [Float](#)
The degree with which to rotate the RectTransform
- **time:float** [Float](#)
The time to complete the tween in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

LeanTween.rotateAround(button, Vector3.forward, 90f, 1f).setDelay(1f);

LeanTween.rotateAroundLocal (gameObject:GameObject , vec:Vector3 , degrees:float , time:float) [LTDScr](#)
Defined in [LeanTween.cs:3237](#)

Rotate a GameObject around a certain Axis in Local Space (the best method to use when you want to rotate beyond 180 degrees)

Parameters:

- **gameObject:GameObject** [GameObject](#)
Gameobject that you wish to rotate
- **vec:Vector3** [Vector3](#)
axis in which to rotate around ex: Vector3.up
- **degrees:float** [Float](#)
the degrees in which to rotate
- **time:float** [Float](#)
time The time to complete the rotation in

Returns:

[LTDScr](#):

LTDScr an object that distinguishes the tween

Example:

Example:

```
LeanTween.rotateAround ( gameObject, Vector3.left, 90f, 1f );
```

LeanTween.rotateLocal (gameObject:GameObject , to:Vector3 , time:float) LTD descr
Defined in [LeanTween.cs:3168](#)

Rotate a GameObject in the objects local space (on the transforms localEulerAngles object)

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to rotate
- **to:Vector3** Vector3
The final rotation with which to rotate to
- **time:float** Float
The time to complete the rotation in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

LeanTween.rotateX (GameObject , to:float , time:float) LTD descr
Defined in [LeanTween.cs:3181](#)

Rotate a GameObject only on the X axis

Parameters:

- **GameObject** GameObject
Gameobject that you wish to rotate
- **to:float** Float
The final x-axis rotation with which to rotate
- **time:float** Float
The time to complete the rotation in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

LeanTween.rotateY (GameObject , to:float , time:float) LTD descr
Defined in [LeanTween.cs:3194](#)

Rotate a GameObject only on the Y axis

Parameters:

- **GameObject** GameObject
Gameobject that you wish to rotate
- **to:float** Float
The final y-axis rotation with which to rotate
- **time:float** Float
The time to complete the rotation in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

LeanTween.rotateZ (GameObject , to:float , time:float) LTD descr
Defined in [LeanTween.cs:3207](#)

Rotate a GameObject only on the Z axis

Parameters:

- **GameObject** GameObject
Gameobject that you wish to rotate
- **to:float** Float

The final z-axis rotation with which to rotate

- **time:float** Float
The time to complete the rotation in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.scale (gameObject:GameObject , vec:Vector3 , time:float) **LTDscr**
Defined in [LeanTween.cs:3254](#)

Scale a GameObject to a certain size

Parameters:

- **gameObject:GameObject** GameObject
gameObject GameObject that you wish to scale
- **vec:Vector3** Vector3
to The size with which to tween to
- **time:float** Float
time The time to complete the tween in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.scale (GUI) (LRect , Vector2 , float) **LTDscr**
Defined in [LeanTween.cs:3267](#)

Scale a GUI Element to a certain width and height

Parameters:

- **LRect** **LRect**
ItRect LRect object that you wish to move
- **Vector2** Vector2
to The final width and height to scale to (pixel based)
- **float** Float
time The time to complete the tween in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

Example:

Example Javascript:

```
var bRect:LRect = new LRect( 0, 0, 100, 50 );
LeanTween.scale( bRect, Vector2(bRect.rect.width, bRect.rect.height) * 1.3, 0.25
).setEase(LeanTweenType.easeOutBounce);
function OnGUI(){
    if(GUI.Button(bRect.rect, "Scale")){ }
}
```

Example C#:

```
LRect bRect = new LRect( 0f, 0f, 100f, 50f );
LeanTween.scale( bRect, new Vector2(150f,75f), 0.25f ).setEase(LeanTweenType.easeOutBounce);
void OnGUI(){
    if(GUI.Button(bRect.rect, "Scale")){ }
}
```

LeanTween.scale (RectTransform) (rectTrans:RectTransform , to:float , time:float) **LTDscr**
Defined in [LeanTween.cs:3548](#)

Rotate a RectTransform object (used in Unity GUI in 4.6+, for Buttons, Panel, Scrollbar, etc...)

Parameters:

- **rectTrans:RectTransform** RectTransform
RectTransform that you wish to attach the tween to
- **to:float** Float
The final Vector3 with which to tween to (localScale)

- **time:float** Float
The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

LeanTween.scale(button, button.localScale*2f, 1f).setDelay(1f);

LeanTween.scaleX (gameObject:GameObject , scaleTo:float , time:float) LTDescr
Defined in [LeanTween.cs:3294](#)

Scale a GameObject to a certain size along the x-axis only

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to scale
- **scaleTo:float** Float
the size with which to scale to
- **time:float** Float
the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.scaleY (gameObject:GameObject , scaleTo:float , time:float) LTDescr
Defined in [LeanTween.cs:3307](#)

Scale a GameObject to a certain size along the y-axis only

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to scale
- **scaleTo:float** Float
the size with which to scale to
- **time:float** Float
the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.scaleZ (gameObject:GameObject , scaleTo:float , time:float) LTDescr
Defined in [LeanTween.cs:3320](#)

Scale a GameObject to a certain size along the z-axis only

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to scale
- **scaleTo:float** Float
the size with which to scale to
- **time:float** Float
the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.value (Color) (gameObject:GameObject , from:Color , to:Color , time:float) LTDescr
Defined in [LeanTween.cs:3477](#)

Tween any particular value (Color)

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to attach the tween to
- **from:Color** Color
The original value to start the tween from
- **to:Color** Color
The final Color with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.value (Color) (GameObject , callOnUpdate:Action<Color> , Color , Color , Color) LTDescr
Defined in [LeanTween.cs:3361](#)

Tween from one color to another

Parameters:

- **GameObject** GameObject
gameObject GameObject with which to tie the tweening with. This is only used when you need to cancel this tween, it does not actually perform any operations on this gameObject
- **callOnUpdate:Action<Color>** Action
The function that is called on every Update frame, this function needs to accept a color value ex: function `updateValue(Color val){ }`
- **Color** Color
from The original value to start the tween from
- **Color** Color
to The value to end the tween on
- **Color** Color
time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

Example Javascript:

```
LeanTween.value( gameObject, updateValueExampleCallback, Color.red, Color.green, 1f).setEase(LeanTweenType.easeOutElastic);
function updateValueExampleCallback( val:Color ){
    Debug.Log("tweened color:"+val+" set this to whatever variable you are tweening...");
}
```

Example C#:

```
LeanTween.value( gameObject, updateValueExampleCallback, Color.red, Color.green, 1f).setEase(LeanTweenType.easeOutElastic);
void updateValueExampleCallback( Color val ){
    Debug.Log("tweened color:"+val+" set this to whatever variable you are tweening...");
}
```

LeanTween.value (float) (GameObject , callOnUpdate:Action<float> , float , float , float) LTDescr
Defined in [LeanTween.cs:3333](#)

Tween any particular value, it does not need to be tied to any particular type or GameObject

Parameters:

- **GameObject** GameObject
gameObject GameObject with which to tie the tweening with. This is only used when you need to cancel this tween, it does not actually perform any operations on this gameObject
- **callOnUpdate:Action<float>** Action
The function that is called on every Update frame, this function needs to accept a float value ex: function `updateValue(float val){ }`
- **float** Float
from The original value to start the tween from
- **float** Float
to The value to end the tween on
- **float** Float
time The time to complete the tween in

to the value to end the tween on

- **float** Float
time The time to complete the tween in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

Example:

Example Javascript:

```
LeanTween.value( gameObject, updateValueExampleCallback, 180f, 270f,
1f).setEase(LeanTweenType.easeOutElastic);
function updateValueExampleCallback( val:float ){
    Debug.Log("tweened value:"+val+" set this to whatever variable you are tweening...");
}
```

Example C#:

```
LeanTween.value( gameObject, updateValueExampleCallback, 180f, 270f,
1f).setEase(LeanTweenType.easeOutElastic);
void updateValueExampleCallback( float val ){
    Debug.Log("tweened value:"+val+" set this to whatever variable you are tweening...");
}
```

LeanTween.value (float) (gameObject:GameObject , from:float , to:float , time:float) LTD descr
Defined in [LeanTween.cs:3435](#)

Tween any particular value (float)

Parameters:

- **gameObject:GameObject** GameObject
GameObject that you wish to attach the tween to
- **from:float** Float
The original value to start the tween from
- **to:float** Vector3
The final float with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

LeanTween.value (float,object) (gameObject:GameObject , callOnUpdate:Action<float,object> , from:float , to:float , time:float) LTD descr
Defined in [LeanTween.cs:3420](#)

Tween any particular value (float)

Parameters:

- **gameObject:GameObject** GameObject
GameObject that you wish to attach the tween to
- **callOnUpdate:Action<float,object>** Action
The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(Vector3 val, object obj){ }
- **from:float** Float
The original value to start the tween from
- **to:float** Vector3
The final Vector3 with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTD descr:

LTD descr an object that distinguishes the tween

LeanTween.value (Vector2) (gameObject:GameObject , from:Vector2 , to:Vector2 , time:float) LTD descr
Defined in [LeanTween.cs:3449](#)

Tween any particular value (Vector2)

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to attach the tween to
- **from:Vector2** Vector2
The original value to start the tween from
- **to:Vector2** Vector3
The final Vector2 with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.value (Vector2) (gameObject:GameObject , callOnUpdate:Action<Vector2> , from:Vector2 , to:Vector2 , time:float) LTDscr
Defined in [LeanTween.cs:3390](#)

Tween any particular value (Vector2), this could be used to tween an arbitrary value like a material color

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to attach the tween to
- **callOnUpdate:Action<Vector2>** Action
The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(Vector3 val){ }
- **from:Vector2** Float
The original value to start the tween from
- **to:Vector2** Vector2
The final Vector3 with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.value (Vector3) (gameObject:GameObject , callOnUpdate:Action<Vector3> , from:Vector3 , to:Vector3 , time:float) LTDscr
Defined in [LeanTween.cs:3405](#)

Tween any particular value (Vector3), this could be used to tween an arbitrary value like a material color

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to attach the tween to
- **callOnUpdate:Action<Vector3>** Action
The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(Vector3 val){ }
- **from:Vector3** Float
The original value to start the tween from
- **to:Vector3** Vector3
The final Vector3 with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTDscr:

LTDscr an object that distinguishes the tween

LeanTween.value (Vector3) (gameObject:GameObject , from:Vector3 , to:Vector3 , time:float) LTDscr
Defined in [LeanTween.cs:3463](#)

Tween any particular value (Vector3)

Parameters:

- **gameObject:GameObject** GameObject
Gameobject that you wish to attach the tween to
- **from:Vector3** Vector3
The original value to start the tween from
- **to:Vector3** Vector3
The final Vector3 with which to tween to
- **time:float** Float
The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween
