APIs

Classes | Modules

Type to filter APIs

- LeanTween
- LeanTweenType
- LTBezierPath
- LTRect

LeanTween Class

Show: Inherited Protected Private Deprecated

Defined in: LeanTween.js:451

LeanTween is an efficient tweening engine for Unity3d

Optional Parameters are passed in a hash table variable that is accepted at the end of every tweening function.
Values you can pass:
delay: time (or frames if you are using "useFrames") before the tween starts
ease: Function that desribes the easing you want to be used, you can pass your own or use many of the included tweens. ex: *{"ease":LeanTween.easeOutQuad}*
onComplete: Function to call at the end of the tween ex: *{"onComplete":functionToCallOnComplete}* or *{"onComplete":functionToCallOnComplete,"onCompleteParam":hashTableToPassToOnComplete}*
onUpdate: Function to call on every update ex: *{"onUpdate":functionToCallOnUpdate}* or *{"onUpdate":functionToCallOnUpdate,"onUpdateParam":hashTableToPassToOnUpdate}*
useEstimatedTime: This is useful if the Time.timeScale is set to zero (such as when the game is paused) or some other value and you still want the tween to move at a normal pace ex: *{"useEstimatedTime":true}*
useFrames: Instead of time passed for both the delay and time value, the amount of frames that have passed is used *ex: {"useFrames":true}*
onCompleteTarget: In C# if you are passing a String to the "onComplete" parameter, this variable allows you to define target to call the function than the game object you are tweening.
onUpdateTarget: The same as onCompleteTarget, but for the onUpdate function.
orientToPath: When moving objects along a bezier curve, this controls whether the object aligns itself with the curve or not **repeat**: If you wish the loop to repeat set this value to something other than 1 **loopType**: If the loop is repeating you can change how it repeats (clamp by default) set this value to ping-pong: *ex: {"repeat":2,"loopType":LeanTweenType.pingPong}*

Index | Methods

Methods

LeanTween.alpha (gameObject:GameObject , to:float , time:float , optional:Hashtable) Int

Defined in LeanTween.js:2204

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 rotate
- **to:float** Float
  The time with which to delay before callin the function
- **time:float** Float
  The time with which to delay before calling the function
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.alphaVertex (gameObject:GameObject , to:float , time:float , optional:Hashtable) Int

Defined in LeanTween.js:2226

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
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.cancel (gameObject:GameObject , id:int)
Defined in LeanTween.js:1241

Cancel a specific tween for a gameObject

Parameters:

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

LeanTween.cancel (ltRect:LTRect , id:int)
Defined in LeanTween.js:1256

Cancel a specific tween for a gameObject (GUI Methods)

Parameters:

- **ltRect:LTRect** LTRect
  LTRect whose tweens you want to cancel
- **id:int** Int
  Id of the tween you want to cancel ex: var id:int = LeanTween.rotate(ltRect, 180);

LeanTween.cancel (gameObject:GameObject)
Defined in LeanTween.js:1227

Cancel all tweens that are currently targeting the gameObject

Parameters:

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

LeanTween.delayedCall (gameObject:GameObject , delayTime:float , callback:String , optional:Hashtable) Int
Defined in LeanTween.js:2182

Call a function after a certain amount of time has passed

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to call the Function on
- **delayTime:float** Float
  The time with which to delay before calling the function

- **callback:String** String
  Function that is called after the certain amount of time.

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.delayedCall (gameObject:GameObject , delayTime:float , callback:String) Int
Defined in LeanTween.js:2170

Call a function after a certain amount of time has passed

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to call the Function on

- **delayTime:float** Float
  The time with which to delay before calling the function

- **callback:String** String
  Function that is called after the certain amount of time.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.delayedCall (gameObject:GameObject , delayTime:float , callback:Function , optional:Hashtable) Int
Defined in LeanTween.js:2152

Call a function after a certain amount of time has passed

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to tie this delayed function call to

- **delayTime:float** Float
  The time with which to delay before calling the function

- **callback:Function** Function
  Function that is called after the certain amount of time.

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.delayedCall (gameObject:GameObject , delayTime:float , callback:Function) Int
Defined in LeanTween.js:2139

Call a function after a certain amount of time has passed

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to tie this delayed function call to

- **delayTime:float** Float
  The time with which to delay before calling the function

- **callback:Function** Function
  Function that is called after the certain amount of time.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.delayedCall (delayTime:float , callback:Function) Int
Defined in LeanTween.js:2120

Call a function after a certain amount of time has passed

Parameters:

- **delayTime:float** Float
  The time with which to delay before calling the function
- **callback:Function** Function
  Function that is called after the certain amount of time.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.init (maxSimultaneousTweens:int)
Defined in LeanTween.js:492

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.move (gameObject:GameObject , to:Vector3 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1802

Move a GameObject to a certain location

Parameters:

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

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
LeanTween.move(gameObject, Vector3(0,-3,5), 2.0, {"ease":LeanTween.easeOutQuad});

C#:
Hashtable optional = new Hashtable();
optional.Add("ease":LeanTweenType.easeOutQuad);
LeanTween.move(gameObject, Vector3(0f,-3f,5f), 1.5f, optional);

LeanTween.move (gameObject:GameObject , [] , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1873

Move a GameObject along a set of bezier curves

Parameters:

- **gameObject:GameObject** GameObject

Gameobject that you wish to move

- **[]** Vector3 optional
  A set of points that define the curve(s) ex: Point1,Handle1,Handle2,Point2,...

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

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0, {"ease":LeanTween.easeOutQuad,"orientToPath":true});

C#:
Hashtable optional = new Hashtable();
optional.Add("ease":LeanTweenType.easeOutQuad);
optional.Add("orientToPath":true);
LeanTween.move(gameObject, new Vector3{Vector3(0f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,1f)}, 1.5f, optional);

LeanTween.move (gameObject:GameObject , [] , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1827

Move a GameObject along a set of bezier curves

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to move

- **[]** Vector3 optional
  A set of points that define the curve(s) ex: Point1,Handle1,Handle2,Point2,...

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

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0, {"ease":LeanTween.easeOutQuad,"orientToPath":true});

C#:
Hashtable optional = new Hashtable();
optional.Add("ease":LeanTweenType.easeOutQuad);
optional.Add("orientToPath":true);
LeanTween.move(gameObject, new Vector3{Vector3(0f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,1f)}, 1.5f, optional);

LeanTween.move (GUI) (ltRect:LTRect , to:Vector2 , time:float) Int
Defined in LeanTween.js:1942

Move a GUI Element to a certain location

Parameters:

- **ltRect:LTRect** LTRect
  LTRect object that you wish to move

- **to:Vector2** Vector2
  The final position with which to move to (pixel coordinates)

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

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.move (GUI) (ltRect:LTRect , to:Vector2 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1919

Move a GUI Element to a certain location

Parameters:

- **ltRect:LTRect** LTRect
  LTRect object that you wish to move

- **to:Vector2** Vector2
  The final position with which to move to (pixel coordinates)

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

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.moveLocal (gameObject:GameObject , to:Vector3 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1959

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

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to rotate

- **to:Vector3** Vector3
  The final positin with which to move to

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

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.moveX (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1741

Move a GameObject along the x-axis

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to move

- **to:float** Float
  The final position with which to move to

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

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.moveY (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1760

Move a GameObject along the y-axis

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to move
- **to:float** Float
  The final position with which to move to
- **time:float** Float
  The time to complete the move in
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.moveZ (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1779

Move a GameObject along the z-axis

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to move
- **to:float** Float
  The final position with which to move to
- **time:float** Float
  The time to complete the move in
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.pause (gameObject:GameObject , id:int)
Defined in LeanTween.js:1285

Pause a specific tween for a gameObject

Parameters:

- **gameObject:GameObject** GameObject
  GameObject whose tweens you want to pause
- **id:int** Int
  Id of the tween you want to cancel ex: var id:int = LeanTween.MoveX(gameObject, 5, 1.0);

LeanTween.pause (gameObject:GameObject)
Defined in LeanTween.js:1305

Pause a specific tween for a gameObject

Parameters:

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

LeanTween.resume (gameObject:GameObject)
Defined in LeanTween.js:1339

Pause a specific tween for a gameObject

Parameters:

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

LeanTween.resume (gameObject:GameObject , id:int)
Defined in LeanTween.js:1324

Pause a specific tween for a gameObject

Parameters:

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

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

LeanTween.rotate (gameObject:GameObject , to:Vector3 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:1545

Rotate a GameObject, to values that are in passed in degrees

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 tween in

- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
LeanTween.rotate(cube, Vector3(180,30,0), 1.5, {"ease":LeanTween.easeInOutQuad,
"onComplete":finishedTweening});

C#:
Hashtable optional = new Hashtable();
optional.Add("ease":LeanTweenType.easeInOutQuad);
optional.Add("onComplete":"finishedTweening");
optional.Add("onCompleteTarget":gameObject);
LeanTween.rotate(cube, Vector3(180f,30f,0f), 1.5f, optional);

LeanTween.rotate (gameObject:GameObject , to:Vector3 , time:float) Int
Defined in LeanTween.js:1527

Rotate a GameObject, to values are in passed in degrees

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 tween in

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
LeanTween.rotate(cube, Vector3(180,30,0), 1.5);

C#:
LeanTween.rotate(cube, Vector3(180f,30f,0f), 1.5f);

LeanTween.rotate (ltRect:LTRect , to:float , time:float , optional:Array) Int
Defined in LeanTween.js:1581

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

Parameters:

- **ltRect:LTRect** LTRect
  LTRect 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:

Int: Returns an integer id that is used to distinguish this tween

Example:

Javascript:
if(GUI.Button(buttonRect.rect, "Rotate"))
LeanTween.rotate(buttonRect4, 150.0, 1.0, ["ease",LeanTween.easeOutElastic]);
GUI.matrix = Matrix4x4.identity;

C#:
if(GUI.Button(buttonRect.rect, "Rotate"))
LeanTween.rotate(buttonRect4, 150.0, 1.0, new object[]{"ease",LeanTween.easeOutElastic});
GUI.matrix = Matrix4x4.identity;

LeanTween.rotateAround (gameObject:GameObject , axis:Vector3 , add:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1708

Rotate a GameObject in the objects around an axis

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to rotate
- **axis:Vector3** Vector3
  The final rotation with which to rotate to
- **add:float** Float
  Rotate in x degrees
- **time:float** Float
  The time to complete the rotation in
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.rotateLocal (gameObject:GameObject , to:Vector3 , time:float , optional:Hashtable)
Defined in LeanTween.js:1687

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
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.rotateX (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1615

Rotate a GameObject only on the X axis

Parameters:

- **gameObject: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
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.rotateX (gameObject:GameObject , to:float , time:float)
Defined in LeanTween.js:1604

Rotate a GameObject only on the X axis

Parameters:

- **gameObject: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

LeanTween.rotateY (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1643

Rotate a GameObject only on the Y axis

Parameters:

- **gameObject: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
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.rotateY (gameObject:GameObject , to:float , time:float)
Defined in LeanTween.js:1632

Rotate a GameObject only on the Y axis

Parameters:

- **gameObject: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

LeanTween.rotateZ (gameObject:GameObject , to:float , time:float)
Defined in LeanTween.js:1660

Rotate a GameObject only on the Z axis

Parameters:

- **gameObject: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

LeanTween.rotateZ (gameObject:GameObject , to:float , time:float , optional:Hashtable)
Defined in LeanTween.js:1671

Rotate a GameObject only on the Z axis

Parameters:

- **gameObject: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
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

LeanTween.scale (gameObject:GameObject , to:Vector3 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:2007

Scale a GameObject to a certain size

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to rotate
- **to:Vector3** Vector3
  The size with which to tween to
- **time:float** Float
  The time to complete the tween in
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.scale (GUI) (ltRect:LTRect , to:Vector2 , time:float , optional:Hashtable) Int
Defined in LeanTween.js:2027

Scale a GUI Element to a certain width and height

Parameters:

- **ltRect:LTRect** LTRect
  LTRect object that you wish to move
- **to:Vector2** Vector2
  The final width and height to scale to (pixel based)
- **time:float** Float
  The time to complete the tween in
- **optional:Hashtable** Hashtable
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.scale (GUI) (ltRect:LTRect , to:Vector2 , time:float) Int
Defined in LeanTween.js:2050

Scale a GUI Element to a certain width and height

Parameters:

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

Returns:

Int: Returns an integer id that is used to distinguish this tween

Example:

Example Javascript:

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

Example C#:

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

LeanTween.value *(gameObject:GameObject , callOnUpdate:Function , from:float , to:float , time:float , optional:Hashtable) Int*
Defined in LeanTween.js:1430

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:Function*** *Function*

The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(val:float){ }

- ***from:float*** *Float*
  The original value to start the tween from

- ***to:float*** *Float*
  The value to end the tween on

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

- ***optional:Hashtable*** *Hashtable*
  The time to complete the tween in

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.value `( gameObject:GameObject , callOnUpdate:Function , from:Vector3 , to:Vector3 , time:float , optional:Hashtable )` *Int*
Defined in LeanTween.js:1465

Tween any particular value (Vector3), it does not need to be tied to any particular type or GameObject

Parameters:

- ***gameObject:GameObject*** *GameObject*
  Gameobject that you wish to attach the tween to

- ***callOnUpdate:Function*** *Function*
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(val:Vector3){ }

- ***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

- ***optional:Hashtable*** *Hashtable*
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.value `( callOnUpdate:Function , from:float , to:float , time:float )` *Int*
Defined in LeanTween.js:1385

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

Parameters:

- ***callOnUpdate:Function*** *Function*
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(val:float){ }

- ***from:float*** *Float*
  The original value to start the tween from

- ***to:float*** *Float*
  The value to end the tween on

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

Returns:

Int: Returns an integer id that is used to distinguish this tween

---

LeanTween.value `( gameObject:GameObject , callOnUpdate:String , from:Vector3 , to:Vector3`

LeanTween.value (*gameObject:GameObject , callOnUpdate:String , from:Vector3 , to:Vector3 , time:float , optional:Hashtable*) *Int*
Defined in LeanTween.js:1502

Tween any particular value (Vector3), it does not need to be tied to any particular type or GameObject

Parameters:

- ***gameObject:GameObject*** *GameObject*
  Gameobject that you wish to attach the tween to
- ***callOnUpdate:String*** *String*
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(val:Vector3){ }
- ***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
- ***optional:Hashtable*** *Hashtable*
  Hashtable where you can pass optional items.

Returns:

Int: Returns an integer id that is used to distinguish this tween