

Simple Use Case

```
myAnimObj = new YAHOO.util.Anim("myDiv", {width:
    {to: 100}, height: {to: 100}});
myAnimObj.animate();
```

Makes the HTML element whose id attribute is "myDiv" resize to a height and width of 100 pixels.

Constructor (YAHOO.util.Anim, ColorAnim, etc.)

```
YAHOO.util.Anim(str | element target, obj
    attributes[, num duration, obj easing]);
```

Arguments:

- (1) **Element id or reference:** HTML ID or element reference for the element being animated.
- (2) **Attributes object:** Defines the qualities being animated; see below.
- (3) **Duration:** Approximate, in seconds.
- (4) **Easing:** Reference to an easing effect, member of YAHOO.util.Easing.

Attributes Object

```
animAttributes = {
    animatedProperty: {
        by: 100, //start at current, change by this much
        to: 100, //start at current, go to this
        from: 100, //ignore current; start from this
        unit: 'em' //can be any legal numeric unit
    }
}
```

Note: Do not include `to` and `by` for the same animation property.

Animation Properties

Use Animation to apply gradual transitions to these properties*:

borderWidth	height
bottom	margin
fontSize	opacity
left	lineHeight
right	padding
top	width

*or to any other member of an element's style object that takes a numeric value

Dependencies

Animation requires the YAHOO Global Object, Dom Collection, and Event Utility.

Interesting Moments in Animation

Event	Fires...	Arguments
onStart	...when anim begins	
onTween	...on every frame	
onComplete	...when anim ends	[0] {frames: <i>total frames</i> , fps: <i>frames per second</i> , duration: <i>of animation in milliseconds</i> }

These are Custom Event members of YAHOO.util.Anim; use these by subscribing: `myAnimInstance.onComplete.subscribe(myOnCompleteHandler);`

Using the Motion Subclass

Use the Motion subclass to define animations to/from a specific point, using (optional) bezier control points.

```
var attributes = {
    points: {
        to: [250, 450],
        control: [[100, 800], [-100, 200], [500, 500]]};
var anim = new YAHOO.util.Motion(element,
    attributes, 1, YAHOO.util.Easing.easeIn);
```

Using the ColorAnim Subclass

Use the ColorAnim subclass to background, text or border colors.

```
var myAnim = new YAHOO.util.ColorAnim(element, {back
    groundColor: { to: '#dcdcdc' } });
myAnim.animate();
```

Using the Scroll Subclass

Use the Scroll subclass to animate horizontal or vertical scrolling of an overflowing page element.

```
var attributes = {
    scroll: { to: [220, 0] }
};
var anim = new YAHOO.util.Scroll(element,
    attributes, 1, YAHOO.util.Easing.easeOut);
```

Solutions

Subscribe to an API method:

```
myAnimObj = new YAHOO.util.Anim(element, {width:
    {to: 100}, height: {to: 100}});
myHandler = function(type, args) {
    someDiv.innerHTML = args[0].fps; //gets frames-
    per-second from the onComplete event}
myAnimObj.onComplete.subscribe(myHandler);
myAnimObj.animate();
```

YAHOO.util.Anim: Properties

- attributes** (obj)
- currentFrame** (int)
- duration** (num)
- totalFrames** (int)
- useSeconds** (b)

YAHOO.util.Anim: Methods

- animate()**
- getEl()**
- getStartTime()**
- isAnimated()**
- stop(bFinish)** if true, advances to last frame of animation

Easing Effects

Members of YAHOO.util.Easing

- backBoth**
- backIn**
- backOut**
- bounceBoth**
- bounceIn**
- bounceOut**
- easeBoth**
- easeBothStrong**
- easeIn**
- easeInStrong**
- easeNone** default; no easing
- easeOut**
- easeOutStrong**
- elasticBoth**
- elasticIn**
- elasticOut**