

COMMON PARAMETERS FOR THE EFFECTS

http://www.tutorialspoint.com/script.aculo.us/scriptaculous_common_parameters.htm Copyright © tutorialspoint.com

The following common options which can be set for all the core effects –

Option	Description
duration	Duration of the effect in seconds, given as a float. Defaults to 1.0.
fps	Target these many frames per second. Defaults to 25. Can't be higher than 100.
transition	Sets a function that modifies the current point of the animation, which is between 0 and 1. Following transitions are supplied – <ul style="list-style-type: none">• Effect.Transitions.sinoidal <i>default</i>• Effect.Transitions.linear• Effect.Transitions.reverse• Effect.Transitions.wobble• Effect.Transitions.flicker
from	Sets the starting point of the transition, a float between 0.0 and 1.0. Defaults to 0.0.
to	Sets the end point of the transition, a float between 0.0 and 1.0. Defaults to 1.0.
sync	Sets whether the effect should render new frames automatically <i>which it does by default</i> . If true, you can render frames manually by calling the render instance method of an effect. This is used by Effect.Parallel.
queue	Sets queuing options. When used with a string, it can be <i>front</i> or <i>end</i> to queue the effect in the global effects queue at the beginning or end, or a queue parameter object that can have {position: <i>front/end</i> , scope: <i>scope</i> , limit: 1}.
delay	Sets the number of seconds to wait before the effect actually starts. Defaults to 0.0.
direction	Sets the direction of the transition. Values can be either 'top-left', 'top-right', 'bottom-left', 'bottom-right' or 'center' <i>Default</i> . Applicable only on Grow and Shrink effects.

Here is an example to apply one or more of the above-mentioned parameters. All the parameters are put in a {} and they are separated by comma , .

```
<html>
  <head>
    <title>script.aculo.us examples</title>
    <script type="text/javascript" src="/javascript/prototype.js"></script>
    <script type="text/javascript" src="/javascript/scriptaculous.js?load=effects"
  ></script>
  </head>

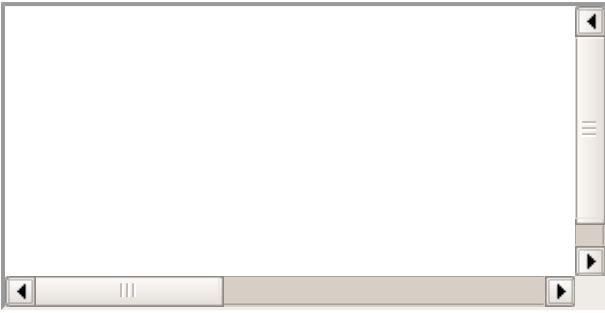
  <body>

    <p>Try by giving different parameters</p>

    <div onclick="new Effect.Opacity(this, {duration: 2.0, transition:
Effect.Transitions.linear, from: 1.0, to: 0.5 });" >
      Click here to see the result:
    </div>
```

```
</body>
</html>
```

This will produce following result –



Callback Methods

You can apply any of the above-mentioned parameters to any element at various events while the effect is running. This is done by writing a callback method in JavaScript for that element.

To use a callback method, you have additional four parameters as listed below:

Callback	Description
beforeStart	Called before the main effects rendering loop is started.
beforeUpdate	Called on each iteration of the effects rendering loop, before the redraw takes places.
afterUpdate	Called on each iteration of the effects rendering loop, after the redraw takes places.
afterFinish	Called after the last redraw of the effect was made.

Within the effect object, there are several useful variables you can access and use them in your callback functions:

Variable	Description
effect.element	The element the effect is applied to.
effect.options	Holds the options you gave to the effect.
effect.currentFrame	The number of the last frame rendered.
effect.startOn	The times <i>inms</i> when the effect was started.
effect.finishOn	The times <i>inms</i> when the effect will be finished after starting an effect.
effect.effects[]	On an Effect.Parallel effect, there's an effects[] array containing the individual effects the parallel effect is composed of.

Example

The following example shows how to use a callback:

```
<html>
  <head>
    <title>script.aculo.us examples</title>
    <script type="text/javascript" src="/javascript/prototype.js"></script>
```

```

<script type="text/javascript" src="/javascript/scriptaculous.js?
load=effects"></script>

<script type="text/javascript">

    function OnFinish(obj){
        alert("Finishing - I'm element :" + obj.element.id);
    }

    function OnStart(obj){
        alert("Starting - I'm element :" + obj.element.id);
    }

    function myEffect(myObject){
        new Effect.Highlight(myObject,
            { startcolor:'#ffffff', endcolor:'#ffffcc', duration: 0.5, afterFinish:
OnFinish, beforeStart: OnStart });
    }
</script>
</head>
<body>

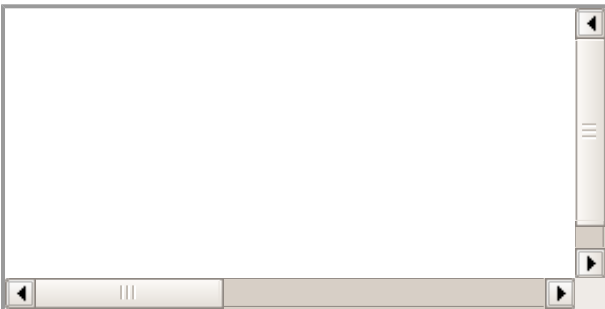
    <p>Click following line to see the result:</p>

    <div onclick="myEffect(this)" >
        Click me to see the result!
    </div>

</body>
</html>

```

This will produce following result –



NOTE – Here *startcolor* and *endcolor* are effect specific parameters. We will discuss these parameters in *Effect Highlight*

Loading [MathJax]/jax/output/HTML-CSS/jax.js