# Scriptaculous

## effects

Andrea Ferracani
andreaferracani@gmail.com

Scriptaculous è una libreria ajax che permette una serie di efetti dinamici sugli elementi del DOM.

E' costruita su Prototype, quindi prima bisogna importare anche questa libreria.

Link: http://script.aculo.us/downloads

Nella cartella src ci sono i file della libreria, mentre in lib una versione compatibile di Prototype

I file devono essere lasciati tutti nella stessa cartella altrimenti non funziona

Il modo più semplice per importare tutti i files è questo:

```
<script type="text/javascript"
src="scripts/scriptaculous.js"></script>
```

Ma il loading può essere anche selettivo attraverso un parametro:

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

# I tipi di effetti della libreria sono principi palmente due: core effects e combina tion effects

| | Effect | Description |
|---|---|---|
| **Core effects** | Opacity | Adjusts the opacity of the target element. |
| | Highlight | Adjusts the background color of the target element. |
| | Scale | Adjusts the size of the target element. |
| | MoveBy | Adjusts the position of the target element. |
| | Parallel | Allows multiple effects to be executed smoothly in parallel (see section 5.9). Although listed as an effect, Parallel is actually a wrapper. |
| **ombination effects** | Appear | Adjusts the opacity of the target element after revealing it. |
| | Fade | Adjusts the opacity of the target element and then hides it. |
| | Puff | Expands the target element while adjusting its opacity. |
| | DropOut | Moves the target element down the page while fading its opacity. |
| | Shake | Moves the target element left and right a few times. |
| | SwitchOff | Flickers the opacity of the target element, and then collapses and hides it (emulating switching off a television). |
| | BlindUp | Progressively hides the target element from the bottom up. |
| | BlindDown | Progressively reveals the element from top to bottom. |
| | SlideUp | Slides the target element up until it is hidden. |
| | SlideDown | Slides the target element down until it is fully revealed. |
| | Pulsate | Adjusts the opacity of the target element multiple times through a range. |
| | Squish | Reduces the target element in size (toward the top left) until it is hidden. |

| | Effect | Description |
|---|---|---|
| **Combination effects** ***(continued)*** | Fold | Reduces the target element in size from top to bottom, then right to left, until it is hidden. |
| | Grow | Reveals the target by scaling it from zero to full size in a specified direction. |
| | Shrink | Hides the target element by scaling it from full size to zero in a specified direction. |

I tipi di effetti della libreria sono princi palmente due: core
effects e combina tion effects

# Effects laboratory - see code labs

Gli effetti di una libreria javascript sono dei metodi di oggetti che adempiono certi tasks dunque ed in quanto metodi richiedono quasi sempre dei parametri.

In Scriptaculous ci sono alcuni parametri di base che ricorrono molto spesso in quasi tutti i metodi:

# I più comuni:

| Option | Description |
|---|---|
| `duration` | Specifies the length of time that an effect is to take. The default is usually 1 second. |
| `from` | Defines the starting point of an effect. Its exact semantics depend on the effect to which it is applied. The default is usually 0.0. |
| `to` | Defines the ending point of an effect. Its exact semantics depend on the effect to which it is applied. The default is usually 1.0. |
| `transition` | Specifies a callback function that controls the progression of the effect. The default function provides a smooth progression. |
| `fps` | Specifies the frames-per-second value. The default is 25. |
| `sync` | Synchronizes effects when applied in parallel. |
| `queue` | Sets the queuing position for effect queues. |

Duration: shake e switchOff non ne tengono conto

From - to: in alcuni effetti tipo opacità è necessario settarli.

Fps: di default è 25, se viene settata a meno, l'animazione può apparire a scatti

# Parametri particolari per i core effects

## Highlight:

| Option | Description |
|---|---|
| `startcolor` | Sets the starting color of the element's background. If omitted, a light yellow color is used. |
| `endcolor` | Sets the ending color of the element's background. If omitted, the original background color of the element is used if it can be determined. The default is white. |
| `restorecolor` | Sets the final color of the background after the effect has completed. |

# Scale:

| Option | Description |
|---|---|
| `scaleX` | Specifies whether scaling in the horizontal direction should occur. Defaults to `true`. If `false`, only vertical scaling takes place. |
| `scaleY` | Specifies whether scaling in the vertical direction should occur. Defaults to `true`. If `false`, only horizontal scaling takes place. |
| `scaleFrom` | Specifies the starting percentage for the effect; defaults to 100. When `from` and `scaleFrom` are both specified, the clipping applied by the `from` value is applied after the `scaleFrom` value has been accounted for. |
| `scaleContent` | Specifies whether the em value of the contents is scaled along with the container. Defaults to `true`. If `false`, the contents are not scaled.<br>Content scaling occurs by modifying the base size of the em measurement. For best cross-browser compatibility, it's best to specify font sizes in em units rather than points or pixels.<br>Images contained in the target element will not be scaled unless they too have had their dimensions assigned using em units. This is rather an odd and painful way to size an image, but if you want it to scale, you'll have to deal with it. Note that if the target itself is an image, it is scaled as you would expect. The "em restriction" only applies to target content, not the target itself. |

| Option | Description |
|---|---|
| `scaleFromCenter` | When `true`, specifies that the target element's center is to remain in its fixed position and that the element is to grow (or shrink, as the case may be) around that point. Defaults to `true`. |
| `scaleMode` | Allows you to indicate the size of the element that should be used as "box" (the default) for the visible size of the element on the page or "content" to take the full size of the element into consideration including any scrollable content that may not be actually visible. Assigning a hash object containing `originalHeight` and `originalWidth` properties can specify a precise size. |

# Combination effects

Sono una combinazione dei core effects.

Fade e appear: elimina l'elemento da flusso della pagina al contrario di opacity. Per una sequenza fade-in fade-out si useranno fade e appear

BlindUp e BlindDown: effetto a serranda, stessi parametri di scale

SlideUp e SlideDown: stesso effetto del precedente, ma si applica anche al contenuto.

C'è però bisogno di un piccolo hack: il testo deve essere inserito all'interno di un div con proprietà display:"block".

Shrink e Grow: effetti di scale, stessi parametri ma più specifici per uno scale dal centro

Puff: effetto a nuvoletta di fumo, combina opacità e scale e rimuove l'elemento dal flusso della pagina

DropOut : Combina opacità e moveBy per rimuovere un elemento

SwitchOff: effetto televisione prima dell'avvento dell'LCD quando si spengeva, combina scale ed opacità

Squish: rimuove l'elemento con uno scale relativo all'angolo top-left

Fold: simile a squish solo a cartella prima su un asse e poi sull'altro

Shake: non rimuove l'elemento. Serve ad attirare l'attenzione, non tiene conto della duration

Pulsate: dà un effetto di opacità da visible to invisible per 5 volte.

# Effect.toggle() method

E' un metodo molto utile per applicare in coppia i combination effects che abbiamo anlizzato.

Effect.toggle( element, effectType, options );

I type sono:

- appear (usa la coppia fade, appear)
- slide (slideUp, slideDown)
- blind (blindUp, blindDown)

uso: Effect.toggle('testSubject','appear',{duration: 1.5});

Vedi esempio d'uso in un widget personalizzato che usa un fieldset per far apparire o sparire dei contenuti premendo un pulsante: è utile per esempio se volessimo costruire una form di input dove alcuni campi sono opzionali.

# Transizioni

Sono formul matematiche che si applicano ad un valore di input in ogni frame. Dunque il risultato dipende anche dal valore di fps. Scriptaculous ha alcune transizioni built-in.

```
new Effect.Opacity(testSubject, {duration: 2,from: 0, to: 1,
transition: Effect.Transitions.pulse});
```

# built-in

| Option | Description |
| --- | --- |
| `Effect.Transitions.sinoidal` | Apparently a misspelling of *sinusoidal*, this transition applies a transform that maps the input values onto a trigonometric curve. This is the default value, as it provides a more visually pleasing progression of the effect from start to finish than if the values were mapped linearly. |
| `Effect.Transitions.linear` | The linear transition passes the input value directly as its result. Thus the effect progresses in a completely linear fashion from the start value to its end. |

# built-in

| Option | Description |
|---|---|
| `Effect.Transitions.flicker` | As its name implies, this transition applies a transform to its input values that presents a "flickery" (for want of a better word) progression from start to finish. |
| `Effect.Transitions.pulse` | This transition applies a transform that causes the applied effect to pulsate as it is being applied. |
| `Effect.Transitions.wobble` | This transition is a variation on the pulse transition; in this case, the pulsating takes place more rapidly toward the end of the effect. |
| `Effect.Transitions.reverse` | This transition applies a transform that inverts its input value so that the effect is applied linearly in the reverse of the specified `from` and `to` option values. |
| `Effect.Transitions.full` | This transition ignores its input value and always returns the value 1.0. This has the result of causing the effect to be applied as if both the `from` and `to` values were specified as 1.0. |

# Callbacks

Gli effetti permettono anche di specificare delle routines che possono essere eseguite tramite dei gestori d'evento che sono definiti come proprietà dell'array associativo

```
new Effect.Opacity( 'testSubject', {duration: 5, from: 0, to: 1,
beforeStart: function(o) { report('beforeStart',o); },
beforeUpdate: function(o) { report('beforeUpdate',o); },
afterUpdate: function(o) { report('afterUpdate',o); },
afterFinish: function(o) { report('afterFinish',o); }
}); //sample general callbacks and cancel
```

# Effetti multipli

Possono essere simultnei o in coda

Parallel:

```
new Effect.MoveBy(targetElement, 100, 200);
new Effect.Scale(targetElement, 200);
```

un effetto in parallelo definito così potrebbe anche funzionare. Ma i browser si comportano diversamente

Si uano Parallel and sync:

```
new Effect.Parallel(
[
new Effect.MoveBy(targetElement, 100, 200, {
sync: true }),
new Effect.Scale(targetElement, 200, { sync: true
})
],
{}
);
```

vedi labs/parallel.html

Serial effects (in coda):

Si potrebbe usare il gestore di evento afterFinish su ogni effetto ma sarebbe hard-coded. Scriptaculous mette a disposizione un meccanismo:

- global queue
- queue personalizzata

GLOBAL:

```
new Effect.MoveBy( ... , { ... , queue: 'front'});
new Effect.Scale( ... , { ... , queue: 'front'});
new Effect.Opacity( ... , { ... , queue: 'front'});
```

In order opacity, scale, moveBy.
Se il valore fosse stato 'end' sarebbe stato il contrario.

Se però abbiamo una serie di threads di effetti da far eseguire simultaneamente, non potremmo usare la global queue

NAMED queues:

```
new Effect.MoveBy( ... , { ... ,
queue: {position:'end',scope:'queue1'});
new Effect.Scale( ... , { ... ,
queue: {position:'end',scope:'queue1'});
new Effect.Opacity( ... , { ... ,
queue: {position:'end',scope:'queue2'});
new Effect.MoveBy( ... , { ... ,
queue: {position:'end',scope:'queue2'});
```

Le due sequenze verranno eseguite simultanemente. Vedi lab-queues.html