



8. Extending jQuery jQueryUI

Client-Side Web Programming

José Socuéllamos

Index

1. Introduction
2. Basic Structure
3. jQueryUI Interactions
4. jQueryUI Widgets
5. jQueryUI Effects



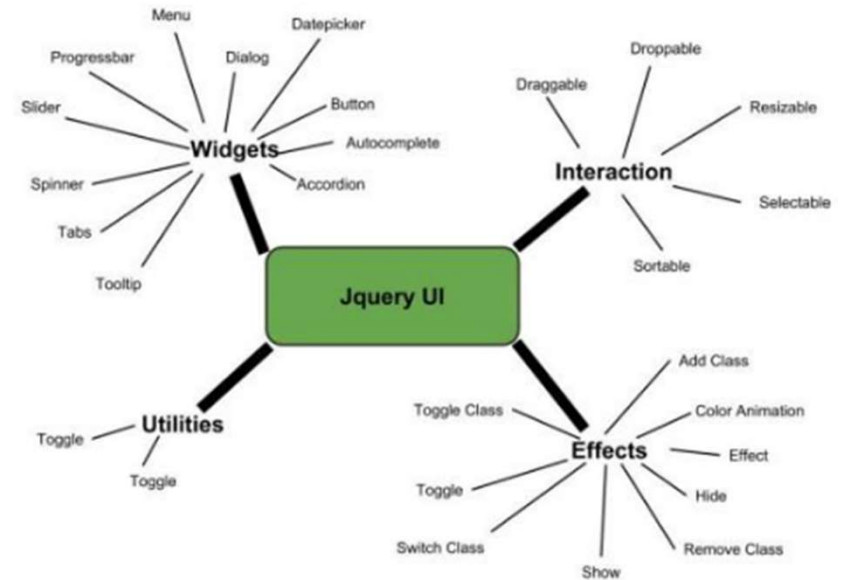
1.1.- jQueryUI

- jQueryUI is the most popular front-end frameworks nowadays.
- It is smooth, intuitive, and powerful mobile first front-end framework for faster and easier web development.
- It uses HTML, CSS and JavaScript.
- It contains a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.
- We can use it to build interactive web applications or just to add stylish elements to forms



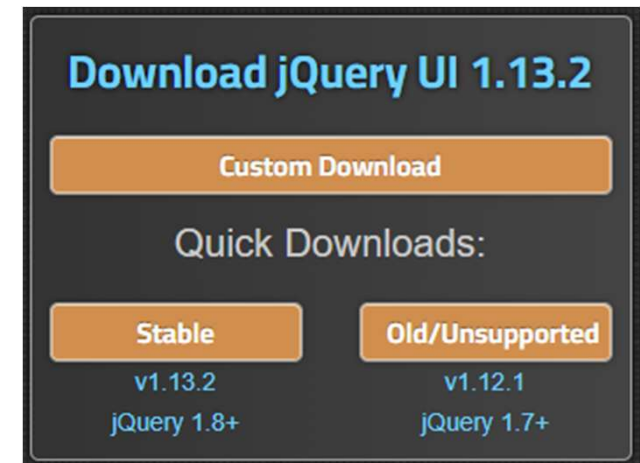
1.2.- jQueryUI Features

- jQueryUI is categorized into four groups:
 - Interactions: drag, drop, resize and more. Gives us the ability to interact with DOM elements.
 - Widgets: with them we can create user interface elements like accordion, datepicker, etc.
 - Effects: They contain a full suite of custom animations and transitions for DOM elements
 - Utilities: modular tools the JQueryUI library uses internally



2.1.- Downloading jQueryUI

- The JQueryUI library can be used in two ways in our web page:
 - Download UI Library from the Official Website:
 - Custom Download – customized version of jQueryUI.
 - Stable – stable and latest version of jQueryUI.
 - Legacy – previous major releases of jQueryUI.
 - Link JQueryUI from CDNs:
 - A CDN (Content Delivery Network) is a network of servers designed to serve files to users.
 - Because a CDN does not require us to host our own version of jQuery UI, it's perfect for testing and learning.



2.2.- jQueryUI “Hello World”

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
<script src="https://code.jquery.com/ui/1.13.2/jquery-ui.min.js"></script>
<link href="https://code.jquery.com/ui/1.13.2/themes/ui-lightness/jquery-ui.css" rel="stylesheet">
```

- First line, adds the jQuery library, as jQuery UI is built on top of jQuery library.
- Second line line, adds the jQuery UI library. This enables jQuery UI in our page.
- Third line, adds jQuery UI theme via CSS. This CSS will make our UI stylish.

```
<script type="text/javascript">
    $(function () {
        $('#dialogMsg').dialog();
    });
</script>
```

- jQueryUI code: it will show a dialog.

```
<form id="form1" runat="server">
    <div id="dialogMsg" title="Hello World">
        Hello! jQueryUI is great!
    </div>
</form>
```

- HTML code: a *form* with a *div* inside. Its **id** is a reserved word that jQuery understands.



3.1.- Draggable Interaction

- jQueryUI provides `draggable()` method to make any DOM element draggable.
- Once the element is draggable, you can move that element by clicking on it with the mouse and dragging it anywhere within the viewport.
- The `draggable()` method has the following syntax:
 - `$(selector, context).draggable(options)`



3.1.- Draggable Interaction

- The draggable method declares that an HTML element can be moved in the HTML page.
- The options parameter is an object that specifies the behavior of the elements involved.
 - `$(selector, context).draggable({option1: value1, option2: value2... })`
- Besides options, the Draggable interaction also supports methods and events. All of them can be checked at: api.jqueryui.com/draggable



3.1.- Draggable Interaction - Example

```
<style>
  #draggable {
    width: 150px;
    height: 150px;
    background: lightgray;
  }
</style>
```

```
<script>
  $("#draggable").draggable();
</script>
```

```
<div id="other"></div>
<div id="container">
  <div id="draggable">
    <p>Drag me!!!</p>
  </div>
</div>
```

- CSS Style of the draggable element
- Draggable method applied on *#draggable*
- *other* and *container* div (we'll use them later)
- *draggable* div element



3.1.- Draggable Interaction - Options

```
$("#draggable").draggable({ distance: 50 });

$("#draggable").draggable({ containment: "parent" });

$("#draggable").draggable({ containment: "#other" });

$("#draggable").draggable({ axis: "x" });

$("#draggable").draggable({
  cursor: "move",
  cursorAt: { top: 56, left: 56 }
});

$( "#draggable" ).draggable({ revert: true });

$( "#draggable" ).draggable({ grid: [ 20, 20 ] });
```

- Moves after dragged 50 pixels
- Dragged only within parent element
- Dragged only within *#other* element
- Dragged only along the horizontal axis
- When dragging, the cursor changes into an arrowed cross and places itself on coordinates (56,56) of *#draggable*
- After being dragged returns to origin
- The movement snaps to a 20x20 grid



3.1.- Draggable Interaction - Methods

```
$("#draggable").draggable("destroy");
```

```
$("#draggable").draggable("disable");
```

```
$("#draggable").draggable("enable");
```

- The #draggable element won't be draggable anymore
- The #draggable element is temporarily disabled
- The #draggable element is reenabled



3.1.- Draggable Interaction - Events

- The Draggable method has some related events.

```
$("#draggable").draggable({  
  drag: function (event, ui) { }  
});  
  
$("#draggable").on("drag", function (event, ui) { });
```

- They can be managed with the event callback specified or binding an event listener to the element.

- The Draggable events are:
 - create: triggered when the draggable is created
 - drag: triggered while the mouse is moved during the dragging
 - start: triggered when dragging starts.
 - stop: triggered when dragging stops.



3.2.- Droppable Interaction

- jQueryUI provides `droppable()` method to make any DOM element droppable at a specified target (a target for draggable elements).
- Once the element is droppable, you can drop a draggable element on it.
- The `droppable()` method has the following syntax:
 - `$(selector, context).droppable(options)`



3.2.- Droppable Interaction

- The droppable method declares that an HTML element can be used as an element on which you can drop other elements.
- The options parameter is an object that specifies the behavior of the elements involved.
 - `$(selector, context).droppable({option1: value1, option2: value2... })`
- Besides options, the Droppable interaction also supports methods and events. All of them can be checked at:
api.jqueryui.com/droppable



3.2.- Droppable Interaction - Example

```
#draggable {  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}  
  
#droppable {  
  width: 150px;  
  height: 150px;  
  margin: 20px;  
}
```

CSS Style of the
draggable and
droppable
elements

```
<script>  
  $(function () {  
    $("#draggable").draggable();  
    $("#droppable").droppable({  
      drop: function (event, ui) {  
        $(this)  
          .addClass("ui-state-highlight")  
          .find("p")  
          .html("Dropped!");  
      }  
    });  
  });  
</script>
```

Draggable method on #draggable element

Droppable method on #droppable element
with a drop event that is triggered when the
dragged element is dropped, changing
some options

```
<div id="draggable" class="ui-widget-content">  
  <p>Drag me to my target</p>  
</div>  
<div id="droppable" class="ui-widget-header">  
  <p>Drop here</p>  
</div>
```

draggable div element

droppable div element

Both of them have a jQueryUI theme
class applied



3.2.- Droppable Interaction - Options

```
$("#draggable").draggable();
$("#droppable").droppable({
  disabled : "true",
  tolerance: 'touch',
  tolerance: 'fit',
  accept: "#draggable",
  accept: "#another",
  activeClass: "active",
  hoverClass: "hover",
  drop: function (event, ui) {
    $(this)
      .addClass("ui-state-highlight")
      .find("p")
      .html("Dropped!");
  }
});
```

- disabled: #droppable won't accept drops
- tolerance: when we consider a drop happens
 - touch: if #draggable is simply touching #droppable
 - fit: if #draggable fits inside #droppable
- accept: #droppable only accepts drops of the #draggable element. If we accept #another, we won't be able to drop #draggable
- activeClass: CSS when #droppable is active (expecting to receive a drop)
- hoverClass: CSS when #droppable is hovered (more than half #draggable is over it)



3.2.- Droppable Interaction - Methods

```
$(function () {  
    $("#draggable").draggable({ revert: true });  
    $("#droppable").droppable({  
        drop: function (event, ui) {  
            $(this).html(ui.draggable.remove().html());  
            $(this).droppable('destroy');  
            $(this)  
                .addClass("ui-state-highlight")  
                .find("p")  
                .html("No more drops accepted!");  
        }  
    });  
});
```

- The #draggable will return to its place if it's not dropped at #droppable
- When #draggable is dropped at #droppable:
 - #draggable is removed
 - #droppable won't accept more drops anymore
 - #droppable will change its look and text
- Besides *drop*, Droppable widget supports more methods like *destroy*, *enable* or *disable*.



3.2.- Droppable Interaction - Events

- The Droppable method has some related events too.
- They can be managed with the event callback specified or binding an event listener to the element.
- The Droppable events are:
 - activate: triggered when an accepted draggable starts dragging
 - create: triggered when the draggable is created
 - deactivate: triggered when an accepted draggable stops dragging
 - drop: triggered when an accepted draggable is dropped on the droppable
 - out: triggered when an accepted draggable is dragged out of the droppable
 - over: triggered when an accepted draggable is dragged over the droppable



3.3.- Resizable Interaction

- The `resizable()` method is able to resize any DOM element.
- It displays an icon in the bottom right of the item to resize.
- The `resizable()` method has the following syntax:
 - `$(selector, context).resizable({option1: value1, option2: value2... })`

```
$("#resizable").resizable({  
  animate: true,  
  ghost: true,  
  containment: "#container",  
  minHeight: 70,  
  minWidth: 100,  
  alsoResize: "resizable2"  
})
```

- `animate`: resize will be animated
- `ghost`: `#resizable` will leave a trail when stretched
- `containment`: `#resizable` won't grow outside `#container`
- `minHeight`: `#resizable` will always this minimum height
- `minWidth`: `#resizable` will always this minimum width
- `alsoResize`: `#resizable2` will be resized with `#resizable`

api.jqueryui.com/resizable



3.4.- Selectable Interaction

- The `selectable()` method selects DOM elements individually or in a group.
- Elements can be selected by dragging a box over them or by clicking them.
- The `selectable()` method has the following syntax:
 - `$(selector, context).selectable({option1: value1, option2: value2... })`

```
$("#selectable").selectable();  
  
<ol id = "selectable">  
  <li>Product 1</li>  
  <li>Product 2</li>  
  <li>Product 3</li>  
  <li>Product 4</li>  
  <li>Product 5</li>  
  <li>Product 6</li>  
  <li>Product 7</li>  
</ol>
```

- By making an ordered list (ol) a selectable element, it's the list items (li) which become selectable
- The selection can be made...
 - by clicking a single element
 - by clicking multiple elements while holding Ctrl key
 - by dragging a box (lasso) over the contiguous items

api.jqueryui.com/selectable



3.5.- Sortable Interaction

- The `sortable()` method reorders elements in list or grid using the mouse.
- `sortable()` makes an HTML element contain interchangeable elements.
- The `sortable()` method has the following syntax:
 - `$(selector, context).sortable({option1: value1, option2: value2... })`

```
$("#sortable").sortable({  
  placeholder: "highlight",  
  connectWith: "#sortable2",  
});
```

```
<ul id = "sortable">  
  <li>Product 1</li>  
  <li>Product 2</li>  
  <li>Product 3</li>  
  <li>Product 4</li>  
  <li>Product 5</li>  
  <li>Product 6</li>  
  <li>Product 7</li>  
</ul>
```

- By making an unordered list (ul) a sortable element, it's the list items (li) which become sortable
- We can rearrange the items by dragging them to a new position
- placeholder: it will show up on an available place by highlighting it
- connectWith: li elements in the #sortable element can be dragged to a second sortable element called #sortable2

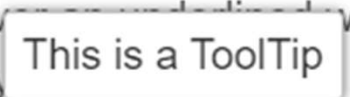
api.jqueryui.com/sortable



4.1.- jQueryUI Widgets

- Widgets are elements with a nice and functional interface that we can add to our pages easily.
- Some are simple but useful, while others are more complex and striking
- Using widgets, we can apply behaviors to the elements.
- Besides, they are fully themeable using a consolidated and coordinated theme mechanism.

the **Tooltips** . They are the
ov
wy

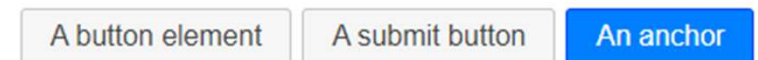
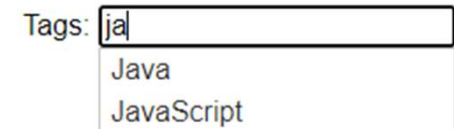
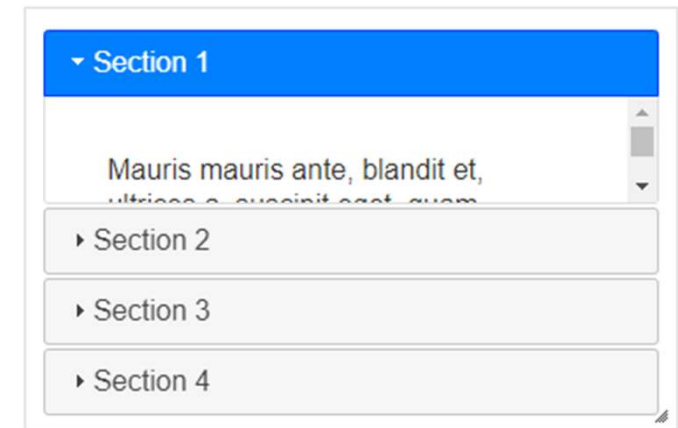


January 2021						
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						



4.1.- jQueryUI Widgets

- Accordion – Accordion containers
- Autocomplete – Auto-complete boxes based on what the user types
- Button – Enhanced button appearance, turn radio buttons and checkboxes into pushbuttons



4.1.- jQueryUI Widgets

- **Checkboxradio** - Enhanced checkbox and radio input elements with styles.
- **Controlgroup** - Groups multiple buttons and other widgets into one visual set.
- **Datepicker** – Advanced date-picker

Select a Location:

☐ New York ☐ Paris ☒ London

Hotel Ratings:

☐ 2 Star ☒ 3 Star ☒ 4 Star ☐ 5 Star

Rental Car

Compact car ☐ Standard ☐ Automatic

☐ Insurance # of cars

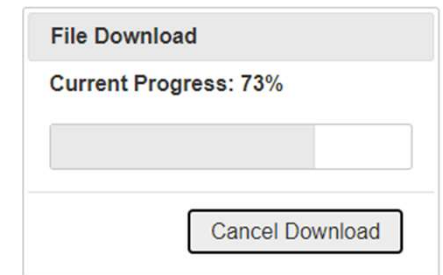
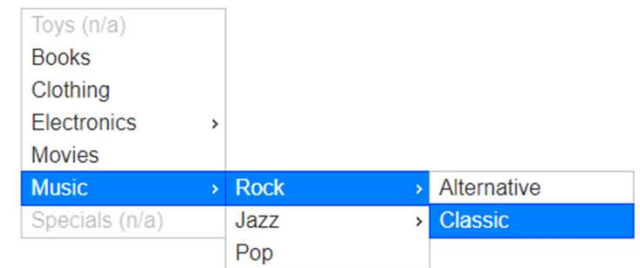
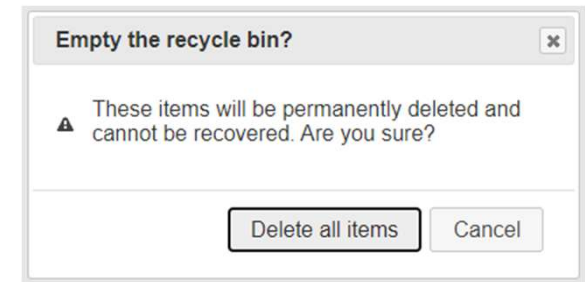
Date:

January 2021						
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						



4.1.- jQueryUI Widgets

- Dialog – Show dialog boxes on top of other content, easily and robustly
- Menu – Themeable menu with mouse and keyboard interactions
- Progressbar – Progress bars, both animated and not



4.1.- jQueryUI Widgets

- Selectmenu – Duplicates and extends the functionality of a native HTML select element to overcome its limitations
- Slider – Fully customizable sliders
- Spinner – Enhanced text inputs for numeric values, with up/down buttons

Select a Person:

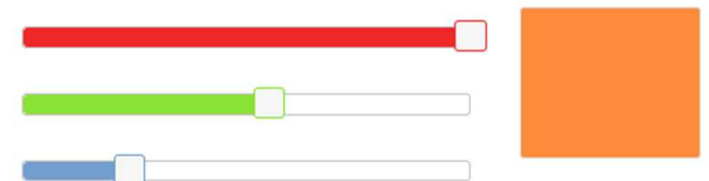


John Resig ▼

John Resig

Tauren Mills

Jane Doe



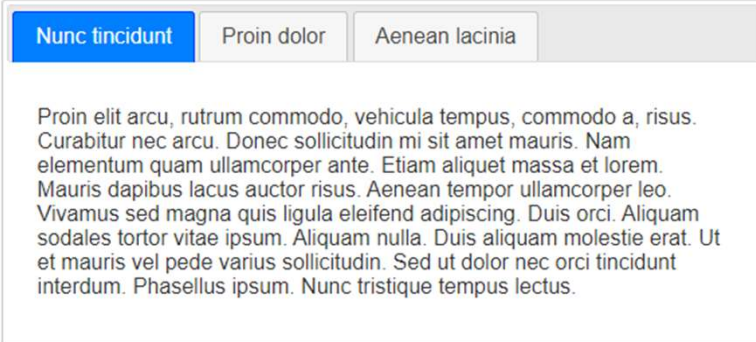
Latitude 44.797

Longitude -93.278



4.1.- jQueryUI Widgets

- Tabs – Tabbed user interface handling, with both inline and demand-loaded content
- Tooltip – Show a Tooltip when hovering an element



A jQueryUI Tabs widget with three tabs: "Nunc tincidunt" (active, blue), "Proin dolor", and "Aenean lacinia". The content area displays a paragraph of Lorem Ipsum text.

Nunc tincidunt Proin dolor Aenean lacinia

Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.



A form with three input fields: "Firstname", "Lastname", and "Address". A tooltip is shown over the "Firstname" field, displaying the text "Please provide your firstname.".

Firstname Please provide your firstname.

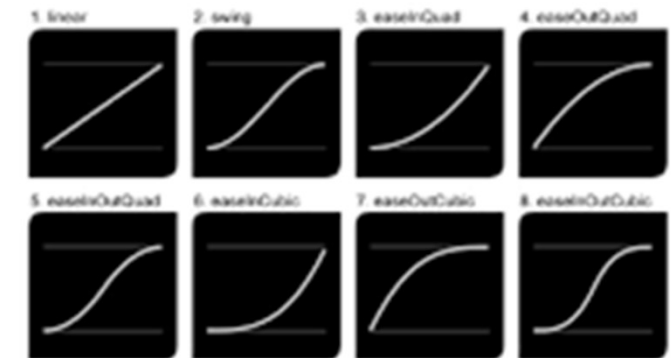
Lastname

Address



5.1.- jQueryUI Effects

- As we saw earlier, jQuery already had some built-in effects (fadeOut/fadeIn, hide/show, slideUp/slideDown, animate...)
- jQuery UI adds quite a bit of functionality on top of those effects.
- Some of them include...
 - animating colors
 - class transitions
 - additional easings
 - custom effects



5.1.- jQueryUI Effects

- **Blind Effect**
 - The blind effect hides or shows an element by wrapping the element in a container, and “pulling the blinds”
- **Bounce Effect**
 - The bounce effect bounces an element. When used with hide or show, the last or first bounce will also fade in/out.
- **Clip Effect**
 - The clip effect will hide or show an element by clipping the element vertically or horizontally.
- **Drop Effect**
 - The drop effect hides or shows an element fading in/out and sliding in a direction.



5.1.- jQueryUI Effects

- **Explode Effect**
 - The explode effect hides or shows an element by splitting it into pieces.
- **Fade Effect**
 - The fade effect hides or shows an element by fading it.
- **Fold Effect**
 - The fold effect hides or shows an element by folding it.
- **Highlight Effect**
 - The highlight effect hides or shows an element by animating its background color first.



5.1.- jQueryUI Effects

- Puff Effect
 - Creates a puff effect by scaling the element up and hiding it at the same time.
- Pulsate Effect
 - The pulsate effect hides or shows an element by pulsing it in or out.
- Scale Effect
 - Shrink or grow an element by a percentage factor.
- Shake Effect
 - Shakes the element multiple times, vertically or horizontally.



5.1.- jQueryUI Effects

- Size Effect
 - Resize an element to a specified width and height.
- Slide Effect
 - Slides the element out of the viewport.
- Transfer Effect
 - Transfers the outline of an element to another element.



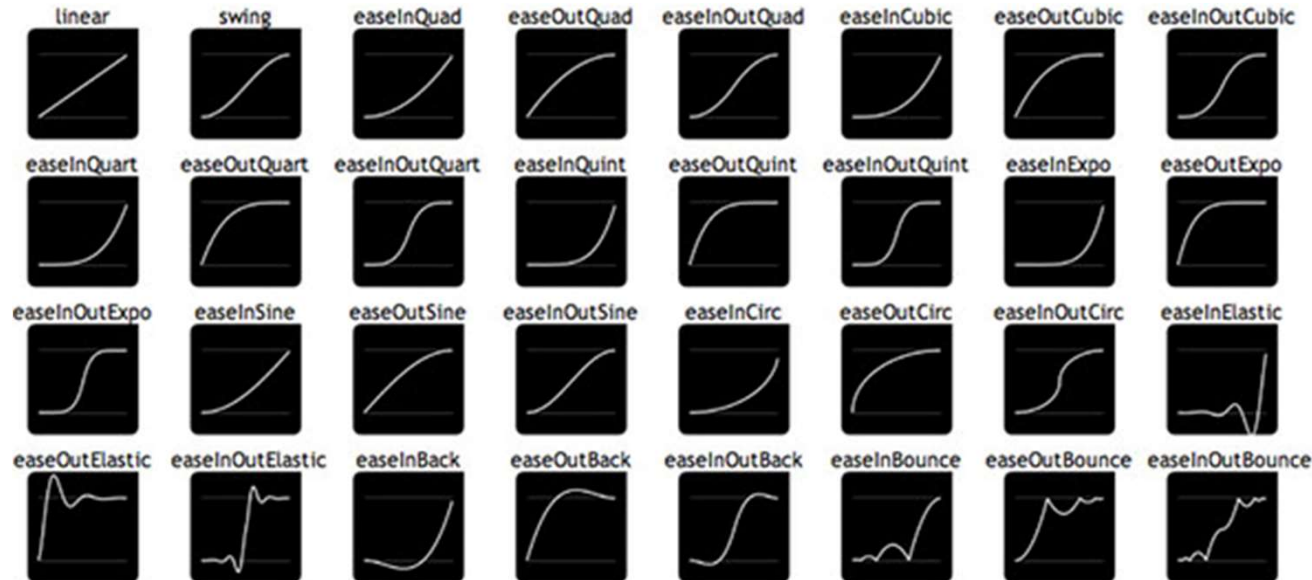
5.2.- jQueryUI Method Overrides

- jQuery UI can override several built-in jQuery methods in order to provide additional functionality:
 - `.addClass() / .removeClass()`
 - Adds/Removes the specified class(es) while animating all style changes.
 - `.toggleClass()`
 - Adds or removes one or more classes while animating all style changes.
 - `.switchClass()`
 - Adds and removes the specified classes while animating all style changes.
 - `.hide() / .show()`
 - Hides/Show using custom effects.
 - `.toggle()`
 - Shows or hides using custom effects.
 - `.position()`
 - Position an element relative to another (div, window, mouse...).



5.3.- jQueryUI Customizing easings

- *Easing* dictates the rate at which an animation progresses.
- jQuery includes two of them: linear and swing.
- jQuery UI allows to customize easing, providing more options.



5.4.- jQueryUI Color Animation

- jQuery UI effects adds the ability to animate color properties using `rgb()`, `rgba()`, hex values, or even color names.
- By simply including the jQuery UI file to our code, the `.animate()` method will gain support for colors.
- Support for color animation comes from the jQuery Color plugin which provides several functions for working with colors.
- The following properties are supported:

backgroundColor	borderBottomColor	borderLeftColor	borderRightColor	borderTopColor
color	columnRuleColor	outlineColor	textDecorationColor	textEmphasisColor

```
$("#elem").animate({  
  color: "green",  
  backgroundColor: "rgb( 20, 20, 20 )",  
  outlineColor: "rgba( 100, 70, 200, 0.5 )"  
});
```



5.4.- jQueryUI Color Animation

- While we can directly animate individual color properties, it is often a better approach to contain the styles in a class.
- As we know, jQuery UI can override some jQuery methods which will animate the addition or removal of a CSS class:
 - `.addClass()`, `.removeClass()`, `.toggleClass()`, and `.switchClass()`.
- These methods will automatically determine which properties need to change and apply the appropriate animations.

```
.style1 {  
  color: ■ green;  
  background-color: ■ rgb(200, 200, 200);  
  outline-color: ■ rgba(100, 70, 200, 0.5);  
}  
  
.style2 {  
  color: ■ white;  
  background-color: ■ rgb(150, 120, 75);  
  outline-color: ■ rgba(0, 170, 125, 1);  
}
```

```
<div id="elem" class="style1"></div>
```

```
$("#elem").toggleClass("style2", 1000);
```

