

vPad – HTML5 + CSS3 App Framework

Created: November 25, 2011
By: VivantDesigns
Email: admin@vivantdesigns.com

Thank you for purchasing my theme. If you have any questions that are beyond the scope of this help file, please feel free to email via my user page contact form [here](#). Thanks so much!

vPad is an easily customizable, feature packed web application framework.

Features:

1. HTML5+CSS3 Compliant
2. Fluid layout – iPhone and iPad friendly
3. iPhone style contact Lists
4. iPhone style drilldown side menu
5. Live Search
6. 960 grid system
7. iPad style Pop over
8. Gallery
9. Full featured form – clearable textfield, validation (email, website, etc), with tooltip, dropdown, autogrow textarea, with placeholder, autocomplete and a lot more.
10. Step by step wizard
11. Calendar
12. WYSIWYG Editor
13. Thoroughly Skinned elements – progress bar, tabs, accordions, dropdown, checkbox, radiobutton
14. Skinned Notifications and growl messages
15. Fully Documented

shared on themelock.com

Table of Contents

1. Setup
2. HTML Structure
3. CSS Files and Structure
4. Javascript
5. Libraries
6. Sources and credits

Setup

In order to run this template on your machine, make sure to run it from a webserver. Ajax will not work properly if the template is opened directly from the filesystem.

HTML Structure

Vpad uses a fluid layout based on 960 Grid system. It has top menu, left-side navigation menu, content area and footer. The template expands (or compresses) with the browsers window's width which is suitable for users using mobile devices, users with small monitor, or users with wide screens.

Here is the general structure.

```

1<div id="wrapper">
2  <!-- HEADER -->
3  <header>
4    <h1><a href="#">Logo or name of company</a></h1>
5    <nav>
6      <div class="container_12">
7        <div class="grid_12">
8          <!-- TOP MENU on LEFT-->
9          <ul class="toolbar clearfix fl">
10            <li>
11              <a href="#" title="Activity" class="icon-only">
12                
13              </a>
14            </li>
15            <li>
16              <a href="#notifications.html" title="Notifications" class="icon-only">
17                
18                <span class="message-count">3</span>
19              </a>
20            </li>
21            <li>
22              <a href="#" title="Settings" class="icon-only">
23                
24              </a>
25            </li>
26          </ul>
27          <!-- TOP MENU on RIGHT -->
28          <a href="#" title="Logout" class="button icon-with-text fr">Logout</a>
29          <div class="user-info fr">
30            Logged in as <a href="#">Administrator</a>
31          </div>
32        </div>
33      </nav>
34    </header>
35    <!-- END HEADER -->
36
37    <section>
38      <!-- Sidebar -->
39      <aside>
40        <nav class="drilldownMenu">
41          <h1>
42            <span class="title">Main Menu</span>
43            <button title="Go Back" class="back">Back</button>
44          </h1>
45
46          <ul class="tim">
47            <!-- basic drill down menu -->
48            <li class="current"><a href="#dashboard.html" title="Dashboard">
49              <span>Dashboard</span></a>
50            </li>
51
52            <!-- drill down with sub menu -->
53            <li class="hasul"><a href="#" title="Forms"><span>Forms</span></a>
54              <ul>
55                <li><a href="#forms.html" title="Form Sample"><span>Form Sample</span></a></li>
56                <li><a href="#forms2.html" title="Custom Form Elements"><span>Custom Form Elements</span></a></li>
57              </ul>
58            </li>
59
60            <!-- drill down with two level sub menu -->
61            <li class="hasul"><a href="#"><span>Documentation</span></a>
62              <ul>
63                <li><a href="#intro-docs.html" title="Documentation"><span>Introduction</span></a></li>
64                <li class="hasul"><a href="#"><span>Sub Menu 1</span></a>
65                  <ul>
66                    <li><a href="#"><span>Sub-Submenu 1</span></a></li>
67                    <li><a href="#"><span>Sub-Submenu 2</span></a></li>
68                  </ul>
69                </li>
70                <li class="hasul"><a href="#"><span>Sub Menu 2</span></a>
71                  <ul>
72                    <li><a href="#"><span>Sub-Submenu 1</span></a></li>
73                    <li><a href="#"><span>Sub-Submenu 2</span></a></li>
74                    <li><a href="#"><span>Sub-Submenu 2</span></a></li>
75                  </ul>
76                </li>
77              </ul>
78            </li>
79          </ul>
80        </nav>
81      </aside>
82      <!-- Sidebar End -->
83
84      <!-- Main Section -->
85      <section>
86        <header>
87          <!-- Main section title -->
88          <div class="container_12 clearfix">
89            <a href="#menu" class="showmenu button">Menu</a>
90            <h1 class="grid_12">Dashboard</h1>
91          </div>
92        </header>
93
94        <section id="main-content" class="clearfix">
95          -- main content --
96        </section>
97
98        <footer class="clearfix">
99          <div class="container_12">
100            <div class="grid_12">
101              -- Footer content here --
102            </div>
103          </div>
104        </footer>
105      </section>
106      <!-- Main Section End -->
107    </section>
108  </div>

```

CSS Files and Structure

These styling sheets are divided into sections for easier editing. If you would like to edit a specific section of the site, simply find the appropriate label in the CSS file, and then scroll down until you find the appropriate style that needs to be edited. The css files used in this template are:

reset.css

The first one is a generic reset file. Many browser interpret the default behavior of html elements differently. By using a general reset CSS file, we can work round this.

grids.css

This stylesheet implements a fixed width layout of 960px and is similar to many frameworks you can find on the Internet. Based on the 960 grid framework.

style.css

Imports device specific style sheet.

general.css

This style sheet contains specific styling for the template structure. It includes the base styles, text styles, header and footer.

tablet.css

A design framework for ipad and other tablets.

mobile.css

Makes the template more readable via mobile device. It makes the site mobile friendly.

wide-mobile.css

Adjust the template layout for wide screen mobile devices.

ui.css

jQuery UI provides a comprehensive set of core interaction plugins, UI widgets and visual effects that use a jQuery-style, event-driven architecture and a focus on web standards, accessibility, flexible styling, and user-friendly design. All plugins are tested for compatibility in IE 6.0+, Firefox 3+, Safari 3.1+, Opera 9.6+, and Google Chrome.

forms.css

Forms.css contains all the needed styling for the template's form.

notifications.css

Add styling to the message growl

jquery.uniform.css

This styles controls the look and feel of form elements such as checkboxes, drop down menus, radio buttons, and file upload inputs between all browsers.

jquery.itextsuggest.css

Style sheet for the search box and the suggestion list.

shared on themelock.com

jquery.popover.css

Add visual styles to the pop over feature.

jquery.icontacts.css

Style sheet that is used for the vertical sliding of the contact list.

We have also added different style sheet for the themes included in this package. These are located in ../css/themes/[name of the theme]/style.css.

1. dark/style.css
2. gray/style.css
3. lightblue/style.css

The default theme is set to lightblue, to change template theme, simply locate below code from index.html and change it to the desired theme.

```
<link rel="stylesheet" href="css/themes/lightblue/style.css" media="screen" />
```

Javascript

Vpad uses the following javascript files:

global.js

This includes all the necessary scripts needed for all the pages.

ie.js

shared on themelock.com

Fixes button mouse actions in IE, and makes use of PIE.js to enable rounded corners, box-shadows and linear-gradients in IE.

html5.js

HTML5 JavaScript shiv for IE to recognise and style the HTML5 elements. It is a way for non-modern browsers to acknowledge html5 elements such as <article>, <section>, etc.

PIE.js

PIE makes Internet Explorer 6–8 capable of rendering several of the most useful CSS3 decoration features.

selectivizr.js

selectivizr is a JavaScript utility that emulates CSS3 pseudo-classes and attribute selectors in Internet Explorer 6–8.

jquery.min.js

jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jquery.tools.min.js

jQuery Tools is plugin for tooltips, overlays, exposing effects and scrollables. It is minimized and/or packed to reduce the size of the files.

jquery.easing.js

A JQuery extension that allows you to set custom easing types for standard animations.

jquery.ui.totop.js

Small jQuery plugin inspired by David Walsh to create a dynamic to-top UI feature.

[jquery.itextsuggest.js](#)

iTextsuggest is a smart auto suggest script with highly customizable options. With this plugin a search text box within a web page could be easily converted into a smart suggestion box that would list out all related search entries even sentences.

[jquery.haschange.min.js](#)

This jQuery plugin enables very basic bookmarkable #hash history via a cross-browser HTML5 window.onhashchange event.

[jquery.drilldownmenu.js](#)

The jquery drill down menu plugin takes standard nested lists and turns them into iPod style drill down menus.

[jquery.autocomplete.js](#)

A Javascript library which provides an autocomplete dropdown which you can attach to one or more form inputs.

[jquery.autogrowtextarea.js](#)

The Autogrow Textarea plugin is pretty straightforward. It allows textareas to grow vertically as more and more text is typed in it.

[jquery.isotope.min.js](#)

An exquisite javascript for magical layouts. Enables filtering, sorting, and dynamic layouts. Isotope works on a container element with a group of similar child items.

[jquery.itextclear.js](#)

A simple javascript that adds clearable feature to any textbox in HTML form. All you need is to call clearable() method on the textbox.

[jquery.notify.js](#)

shared on themelock.com

This is a simple Javascript that display Ubuntu like messages or simply growl messages.

[jquery.popover.js](#)

JQuery popover is a javascript to create iPad like popovers.

[jquery.ui.widget.min.js](#)

This is a jquery used for the growl message.

[jquery.uniform.min.js](#)

Uniform masks your standard form controls with custom themed controls. A plugin for jQuery that lets you style select, radio, and checkboxes however you desire.

[jquery.contacts.js](#)

Jquery used for the iPhone style contact list.

[jquery.wizard.js](#)

A simple jquery plugin for a javascript wizard flow.

Libraries

This template uses the following libraries:

- [FullCalendar](#) is a jQuery plugin that provides a full-sized, drag & drop calendar. It uses AJAX to fetch events on-the-fly for each month and is easily configured to use your own feed format (an extension is provided for Google Calendar). It is visually customizable and exposes hooks for user-triggered events (like clicking or dragging an event). It is open source and dual licensed under the MIT or GPL Version 2 licenses.

- [TinyMCE](#) - It is a platform independent web based Javascript HTML WYSIWYG editor control released as Open Source under LGPL by Moxiecode Systems AB. It has the ability to convert HTML TEXTAREA fields or other HTML elements to editor instances.
- [Datatables](#) is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, which will add advanced interaction controls to any HTML table.
- [jqPlot](#) is a plotting and charting plugin for the jQuery Javascript framework. jqPlot produces beautiful line, bar and pie charts with many features.

Sources and credits

The jQuery and its plugins are licensed under the MIT, GPL Version 2 or BSD licenses. You can read these licenses here:

- <http://www.opensource.org/licenses/mit-license.php>
 - <http://www.opensource.org/licenses/gpl-license.php>
-

Once again, thank you so much for purchasing this theme. As I said at the beginning, I'd be glad to help you if you have any questions relating to this theme. No guarantees, but I'll do my best to assist. If you have a more general question relating to the themes on ThemeForest, you might consider visiting the forums and asking your question in the "Item Discussion" section.

VivantDesigns

shared on themelock.com

Forms

Form general structure

```
<form class="form has-validation">
  <div class="clearfix">
    <label for="form-name" class="form-label">Name <em>*</em></label>
    <div class="form-input">
      <input type="text" id="form-name" name="name" required="required" placeholder="Enter your
name" />
    </div>
  </div>
  <div class="form-action clearfix">
    <button class="button" type="submit">OK</button>
    <button class="button" type="reset">Reset</button>
  </div>
</form>
```

Class "has-validation" enables validation on form inputs. Below form is the result of the above html code.

Name *

Enter your name

OKReset

Input properties:

Property	Values	Description
type	text email date password url file checkbox radio submit reset <code>type="text"</code>	This attribute specifies the type of control to create. The default value for this attribute is "text".
name	[user-defined] <code>name="form-name"</code>	This attribute assigns the control name.
id	[user-defined] <code>id="name"</code>	This attribute assigns the control id.
required	required <code>required="required"</code>	This means that the value should not be empty.
placeholder	[user-defined] <code>placeholder="Enter your name"</code>	Serve as additional info. Placeholder text is text that appears inside of a form input when the page loads, but disappears when the user places the cursor in that form input.
maxlength	[number]	When the type attribute has the value "text" or "password", this attribute specifies the maximum number of characters the user may enter.
data-equals	[name of another control] <code>data-equals="password"</code>	Will validate if the text matches the specified textfield. (as seen in Re-enter password)

* Form will automatically check for valid email, date and url.

* Form will check empty required input or invalid data input, hence it will show error message.

Custom Form elements

Basic HTML Code:

```
<input type="text" id="form-name" name="name"/>
```

To make the textfield clearable, add below javascript code:

```
$("input[type=text]").iTextClear();
```

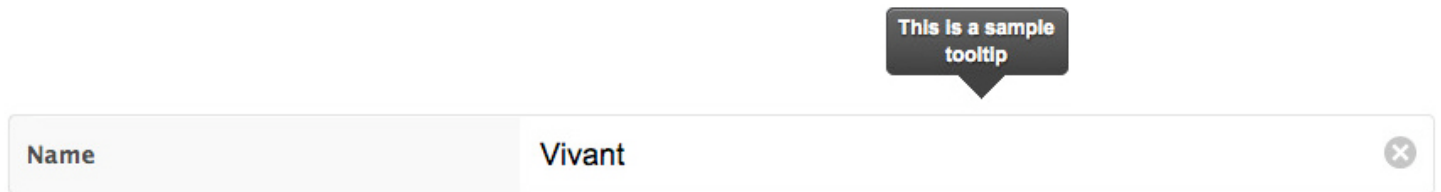
Specify placeholder attribute to add placeholder text

```
<input type="text" id="form-name" name="name" placeholder="Enter your name" />
```

To get tooltip, add "has-tooltip" class and specify title attribute

```
<input type="text" id="form-name" name="name" placeholder="Enter your name" title="This is a sample tooltip" class="has-tooltip"/>
```

The above codes will output below textfield:

A screenshot of a web form element. It consists of a light gray rectangular box. On the left side of the box, the word "Name" is written in a dark gray font. In the center of the box, the word "Vivant" is displayed. On the right side of the box, there is a small circular button with a white 'x' inside. Above the right side of the box, there is a dark gray tooltip bubble with a white border and a white arrow pointing down towards the box. Inside the tooltip, the text "This is a sample tooltip" is written in white.

Autocomplete Textfield

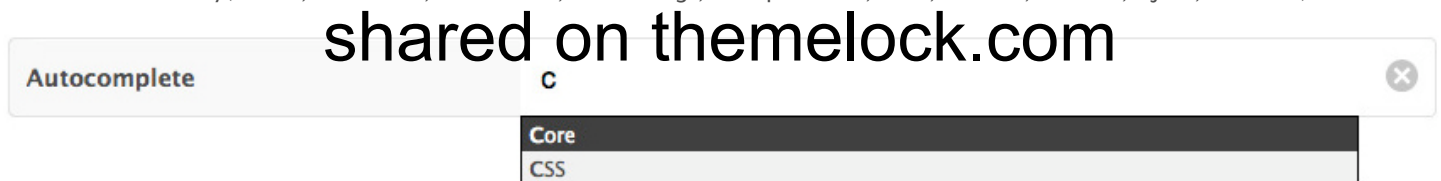
Autocomplete feature is important when typing for a very long words. It also decreases the amount of time spent typing words.

To add autocomplete in textfield, use the following javascript:

```
$("#autocomplete").autocomplete(data-source);
```

Sample usage:

```
data-source = array("Core","Selectors","Attributes","Traversing","Manipulation","CSS","Events","Effects","Ajax","Utilities")
```

A screenshot of an autocomplete text input field. The input field is a light gray box with the word "Autocomplete" on the left and a clear button (a circle with an 'x') on the right. The text "c" is entered into the field. Below the input field, a dropdown menu is open, showing two suggestions: "Core" and "CSS". The "Core" suggestion is highlighted with a dark gray background. A large, semi-transparent watermark text "shared on themelock.com" is overlaid across the center of the image.

Autoresize Textarea

We use auto grow textarea for sites needing large user input. Auto resize or autogrow textarea will grow vertically as it becomes full.

To enable autogrow textarea, add the following javascript:

```
$("#autotextarea").autoGrow();
```

Check on some other custom form element on forms2.html of the live preview.

Media

Create a dynamic and intelligent media layouts with sorting, filtering, and re-ordering features using [isotope](#) plugin.

Options:

Option	Values	Description
Itemselector	[selector]	Restricts Isotope item elements to selector.
transformsEnabled	default: true	Isotope uses CSS3 transform to position item elements, when available in the browser. Setting transformsEnabled to false will disable this feature so all browsers use top/left absolute positioning.
visibleStyle	default: { opacity : 1, scale : 1 }	The style applied to item elements made visible by Filtering.
hiddenStyle	default: { opacity : 0, scale : 0.001 }	The style applied to item elements hidden by Filtering.
layoutMode	default: 'masonry'	The way the items are arranged or laid out.

Check for complete isotope option here

Method:

Method	Script	Description
addItem	.isotope('addItem', \$items, callback)	Adds item elements to the pool of item elements of the container, but does sort, filter or layout.
appended	.isotope('appended', \$items, callback)	Adds item elements via addItem method, then triggers layout just for those new elements. Useful for Infinite Scroll.
destroy	.isotope('destroy')	Removes Isotope functionality completely. Returns element back to pre-init state.
reLayout	.isotope('reLayout', callback)	Resets layout properties and lays-out every item element.
shuffle	.isotope('shuffle', callback)	Shuffles order of items. Sets sortBy option to 'random'.

Check for complete isotope method here

Isotope methods has the following syntax:

```
$("#container").isotope("methodName", [optionalParameters] )
```

Lets get started.

Let's start by creating the HTML code. Isotope works on a container element with a group of similar child items.

```
<ul id="media">
  <li class="architecture"></li>
  <li class="architecture"></li>
  <li class="architecture"></li>
  <li class="flowers"></li>
  <li class="flowers"></li>
  <li class="holiday_wm"></li>
  <li class="holiday_wm"></li>
  <li class="holiday_wm"></li>
  <li class="holiday_wm"></li>
  <li class="industrial"></li>
  <li class="industrial"></li>
  ...
</ul>
```

Now for the javascript script:

```
$('#media').isotope({
  // options
  itemSelector : 'li',
  transformsEnabled: false,
  visibleStyle: { opacity: 1 },
```

```
hiddenStyle: { opacity: 0 }  
});
```

Check out `general.css` for the following css:

Add these styles to your CSS for filtering, animation with CSS transitions, and adding items.

```
.isotope-item {  
  z-index: 2;  
}  
  
.isotope-hidden.isotope-item {  
  pointer-events: none;  
  z-index: 1;  
}  
  
.isotope,  
.isotope .isotope-item {  
  -webkit-transition-duration: 0.8s;  
  -moz-transition-duration: 0.8s;  
  transition-duration: 0.8s;  
}  
  
.isotope {  
  -webkit-transition-property: height, width;  
  -moz-transition-property: height, width;  
  transition-property: height, width;  
}  
  
.isotope .isotope-item {  
  -webkit-transition-property: -webkit-transform, opacity, background, -webkit-box-shadow;  
  -moz-transition-property: -moz-transform, opacity, background, -moz-box-shadow;  
  transition-property: transform, opacity, background, box-shadow;  
}  
  
.isotope .isotope-item {  
  -webkit-transition-property: top, left, opacity;  
  -moz-transition-property: top, left, opacity;  
  transition-property: top, left, opacity;  
}
```

shared on themelock.com

Also don't forget to include the following:

- `jquery.isotope.min.js`
- javascript 1.4.3 and above

* Check isotope complete documentation [here](#).

* Commercial use of Isotope requires purchase of one-time license fee per developer seat. Use in non-commercial and personal applications is free.

Notifications

There are four notification types.

- Information (info)
- Success (success)
- Warning (warning)
- Error (error)

Basic Notification

To show notification, simply add the class "message" and the type of the notification to show.

Sample Usage:

Information

This is an info message.

HTML Code:

```
<div class="message info">
  <h3>Information</h3>
  <p>This is an info message.</p>
</div>
```

Success!

This is a success message.

shared on themelock.com

HTML Code:

```
<div class="message success">
  <h3>Success!</h3>
  <p> This is a success message.</p>
</div>
```

Closeable Notification

To make notification closable, add class "message", "closeable" and the type of the notification to show. Also add span (with class "message-close") to hold on the close icon.

Sample Usage:

Warning!

This is a closeable warning message.



HTML Code:

```
<div class="message warning closeable">
  <span class="message-close"></span>
  <h3>Warning!</h3>
  <p> This is a closeable warning message. </p>
</div>
```

Error!

This is a closeable error message.



HTML Code:

```
<div class="message error closeable">
    <span class="message-close"></span>
    <h3>Error!</h3>
    <p> This is a closeable error message. </p>
</div>
```

You can edit the background color, style or font of the message in notifications.css.

Growl

To use growl the following JavaScript and CSS must be included:

- jQuery
- jquery.ui.widget.min.js
- jquery.notify.js
- notifications.css

Now create container to hold the growl notifications.

```
<div id="container">
    <div id="default-container">
        <h1>#{title}</h1>
        <p>#{text}</p>
    </div>

    <div id="sticky-container">
        <a class="ui-notify-close ui-notify-cross" href="#">x</a>
        <h1>#{title}</h1>
        <p>#{text}</p>
    </div>
</div>
```

shared on themelock.com

Next is to initialize widget on a container, passing in all the defaults

```
$("#container").notify();
```

Now, we are ready to call the create method.

Syntax:

```
$("#container").notify("create", ["template-name",] {parameter}, {options});
```

where "template-name" is an optional id of the containing template

```
$("#container").notify("create", "default-container" {
    title: 'Default Notification',
    text: 'This is an example of the default config, and will fade out after five seconds.'
});
```

Or set specific options for the notification, just like below code that will make a sticky notification.

```
$("#container").notify("create", "sticky-container", {
    title: 'Sticky Notification',
    text: 'This is an example of the default config but will not fade.'
},{
    expires: false,    speed: 1000
});
```

Growl options:

Option	Default	Description
speed	500	The amount of time in milliseconds to fade notifications in and out.
expires	5000	Notifications will automatically close after this amount of time, in milliseconds. Set to 0 or false to create a "sticky" notifications.
stack	Below	Notifications will stack downwards if set to "below" (default), or upwards if set to "above."
custom	False	A boolean value denoting whether or not the widget should apply its own styling classes. Set to false to roll your own notification themes (including ThemeRoller). You could simply overwrite the included CSS, but using this option allows you to create default AND custom notifications within the same containers, as well as maintain a clear upgrade path as you won't have to touch any of the included files.

Sample Usage:

Default Notification

This is an example of the default config, and will fade out after five seconds.

Sticky Notificationx

This is an example of the default config but will not fade.

shared on themelock.com

Calendar

Add a fully functional calendar to your webpages using FullCalendar.

FullCalendar is a jQuery plugin that provides a full-sized, drag & drop calendar like the one below.

Basic Usage

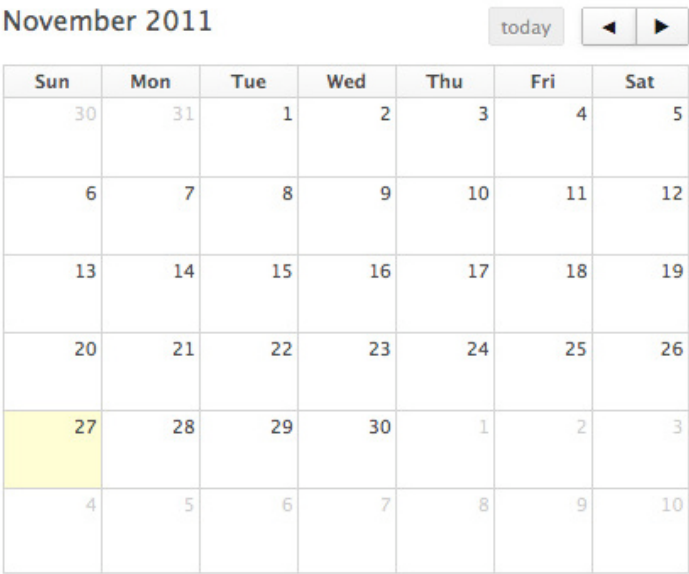
HTML Code is as simple as:

```
<div id="fullcalendar"></div>
```

Next is to set up the JavaScript and the css files in the head of your html files. Then we need to write the JavaScript code that initializes the calendar.

```
Javascript:
$(document).ready(function() {
// page is ready, initialize calendar...
$('#calendar1').fullCalendar({
// put options and callbacks here
})
});
```

As simple as that. Now we have the basic calendar that has no events on it.



Customize Display

Options	
header	Defines the buttons and title at the top of the calendar. <div>{ left: 'title', center: '', right: 'today prev,next' }</div>
theme	Enables/disables use of jQuery UI theming. <div>Boolean, default: false</div>
firstDay	The day that each week begins. <div>Integer, default: 0 (Sunday)</div>
weekends	Whether to include Saturday/Sunday columns in any of the calendar views. <div>Boolean, default: true</div>
height	Will make the entire calendar (including header) a pixel height. <div>Integer</div>

Check complete Calendar Display options here.

◀	▶	today	November 2011				month	day
Mon	Tue	Wed	Thu	Fri	Sat	Sun		
31	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	1	2	3	4		
5	6	7	8	9	10	11		

Javascript:

```
$(document).ready(function() {
    $('#calendar2').fullCalendar({
        header: {
            left: 'prev,next today',
            center: 'title',
            right: 'month,agendaDay'
        },
        firstDay:1, //Monday
        height:320
    });
});
```

Available Views

There are 5 ways of displaying days and events and these views are all built in to FullCalendar.

- month
- basicWeek
- basicDay
- agendaWeek
- agendaDay

You can set the initial view of the calendar with the `defaultView` option.

◀	▶	Nov 27 — Dec 3 2011				today
Sun 11/27	Mon 11/28	Tue 11/29	Wed 11/30	Thu 12/1	Fri 12/2	Sat 12/3

Javascript:

```
$(document).ready(function() {
    $('#calendar3').fullCalendar({
        header: {
            left: 'prev,next',
            center: 'title',
            right: 'today'
        },
        defaultView: 'basicWeek'
    });
});
```

Events

To add array of event that will be displayed on the calendar:

```
$('#calendar').fullCalendar({
    events: [
        {
            title : 'event1',
            start : '2011-11-05'
        },
        {
            title : 'event2',
            start : '2011-11-15',
            end : '2011-11-18'
        },
        {
            title : 'event3',
            start : '2011-11-25 12:30:00',
            allDay : false // will make the time show
        }
    ]
});
```

```
    }  
  ]  
});
```

Here are some other common options used:

Options	Description
editable	Determines whether the events on the calendar can be modified. Boolean, default: false This determines if the events can be dragged and resized. Enables/disables both at the same time.
disableDragging	Disables all event dragging, even when events are editable. Boolean, default: false
disableResizing	Disables all event resizing, even when events are editable. Boolean, default: false
selectable	Allows a user to highlight multiple days or timeslots by clicking and dragging. Boolean/View Option Hash, default: false

Check for live preview for calendar with added events, enable event dragging and enable editing.

* If you plan on doing dragging or resizing, you need some additional jQuery UI files.

* Check FullCalendar complete documentation here.

shared on themelock.com

WYSIWYG

Create a fully featured text editor developed using javascript and css. Uses the [TinyMCE](#) WYSIWYG Editor.

Basic Setup

HTML Code is as simple as:

```
<textarea id="elm1" name="elm1" style="width:100%"> </textarea>
```

To set up tinyMCE we have to include the jquery.tinymce.js in the html files. Then we can initialize the tinymce plugin for the textareas using \$('textarea.tinymce').tinymce function.

Below code displays all plugins and buttons that comes with the TinyMCE package..

```
$( 'textarea.tinymce' ).tinymce({
  // Location of TinyMCE script
  script_url : './lib/tinymce/tinymce.js',

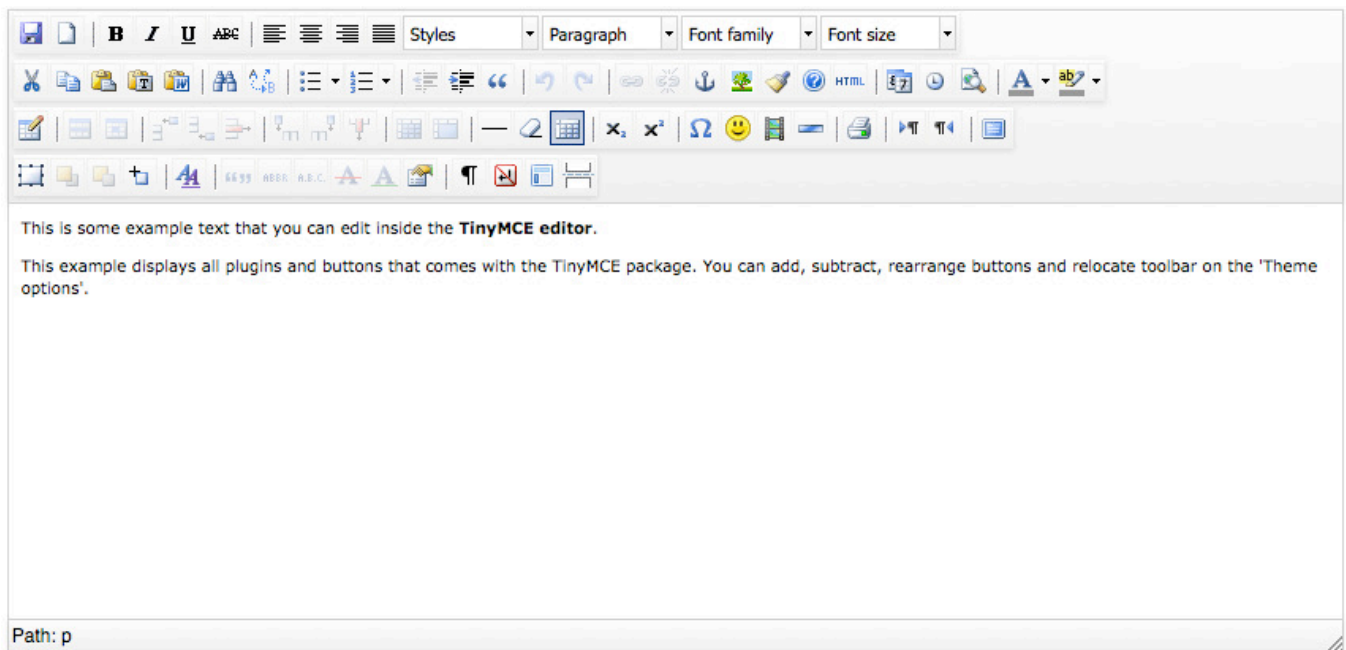
  // General options
  elements : "elm1", //id of the textarea
  theme : "advanced", // can also be 'simple'
  plugins :
    "autolink,lists,spellchecker,pagebreak,style,layer,table,
    save,advhr,advimage,advlink,emotions,iespell,inlinepopups,insertdatetime,
    preview,media,searchreplace,print,contextmenu,paste,directionality,
    fullscreen,noneditable,visualchars,nonbreaking,xhtmlxtras,template,advlist",

  // Theme options
  theme_advanced_buttons1 :
    "save,newdocument,l,bold,italic,underline,striktthrough,l,justifyleft,
    justifycenter,justiftright,justifyfull,styleselect,formatselect,fontselect,
    fontsizeselect",
  theme_advanced_buttons2 :
    "cut,copy,paste,pastetext,pasteword,l,search,replace,l,bulldist,numlist,l,
    outdent,indent,blockquote,l,undo,redo,l,link,unlink,anchor,image,cleanup,
    help,code,l,insertdate,inserttime,preview,l,forecolor,backcolor",
  theme_advanced_buttons3 :
    "tablecontrols,l,hr,removeformat,visualaid,l,sub,sup,l,charmap,emotions,
    iespell,media,advhr,l,print,l,ltr,rtl,l,fullscreen",
  theme_advanced_buttons4 :
    "insertlayer,moveforward,movebackward,absolute,l,styleprops,l,cite,abbr,
    acronym,del,ins,attribs,l,visualchars,nonbreaking,template,pagebreak",
  theme_advanced_toolbar_location : "top",
  theme_advanced_toolbar_align : "left",
  theme_advanced_statusbar_location : "bottom",
  theme_advanced_resizing : true,

  // Example content CSS (should be your site CSS)
  content_css : "css/content.css",

  // Drop lists for link/image/media/template dialogs
  template_external_list_url : "lists/template_list.js",
  external_link_list_url : "lists/link_list.js",
  external_image_list_url : "lists/image_list.js",
  media_external_list_url : "lists/media_list.js",

  // Replace values for the template plugin
  template_replace_values : {
    username : "Some User",
    staffid : "991234"
  }
});
```

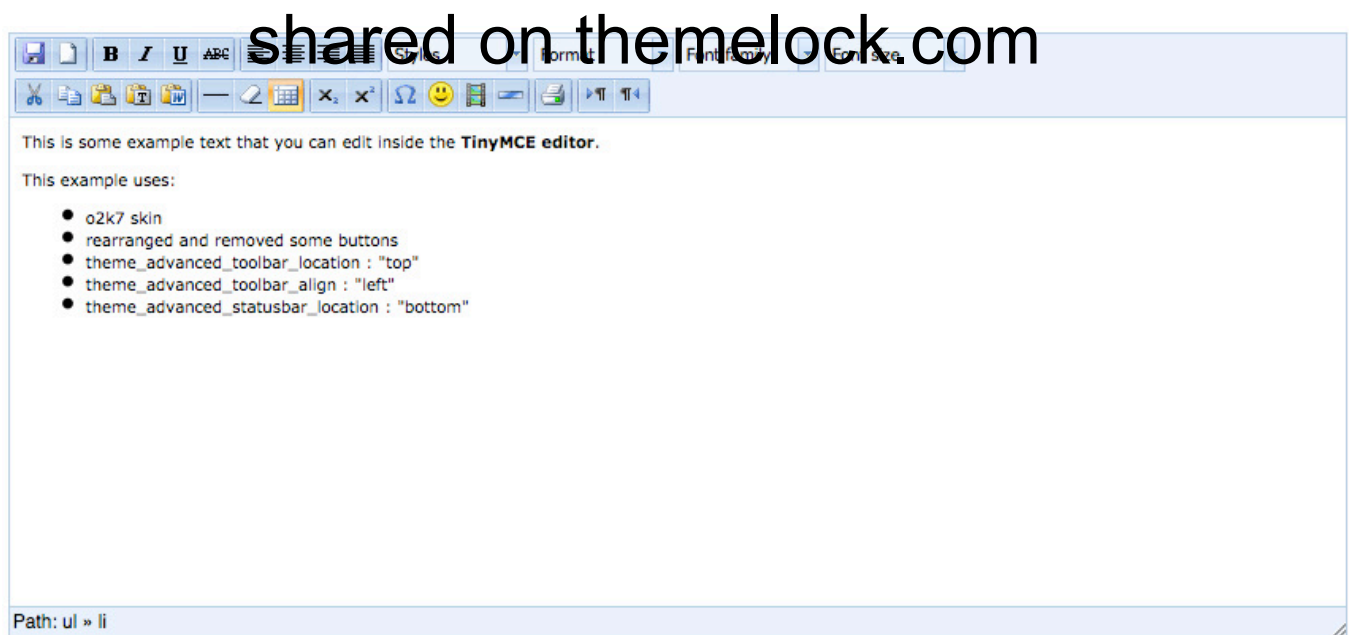


Customize Skin:

TinyMCE has pre-made skins to choose from. Using this skins, user can create a well designed text editor.

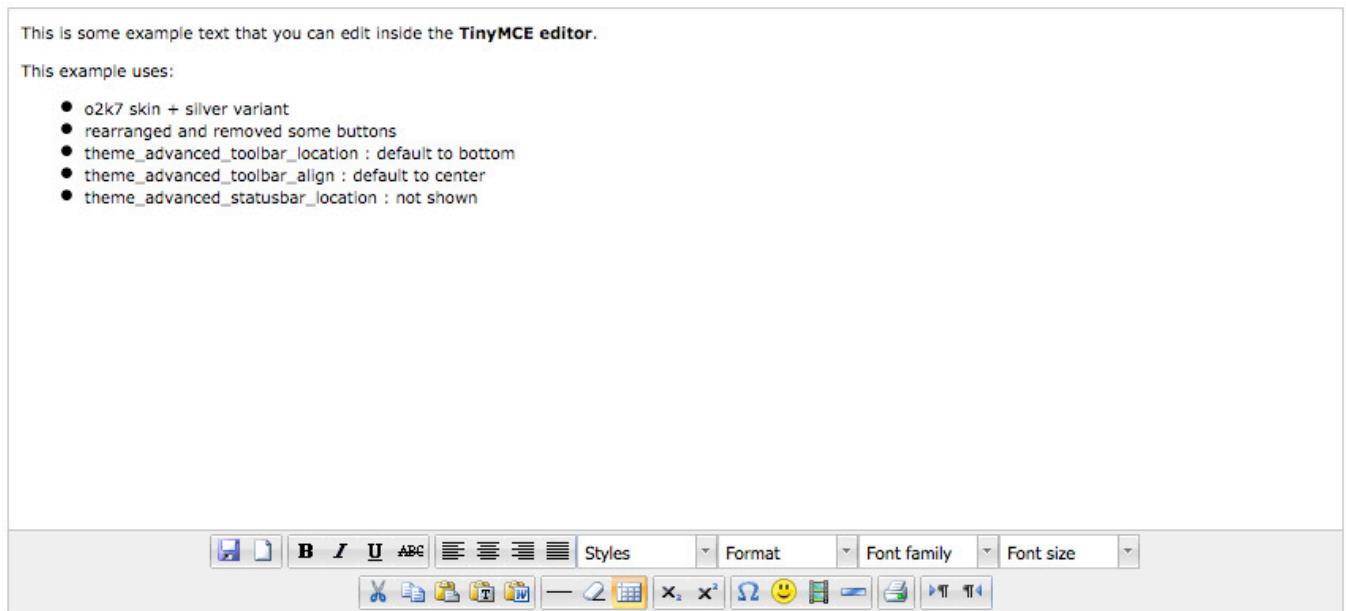
Editor with o2k7 skin enabled

```
// General options
mode : "exact",
theme : "advanced",
skin : "o2k7"
```



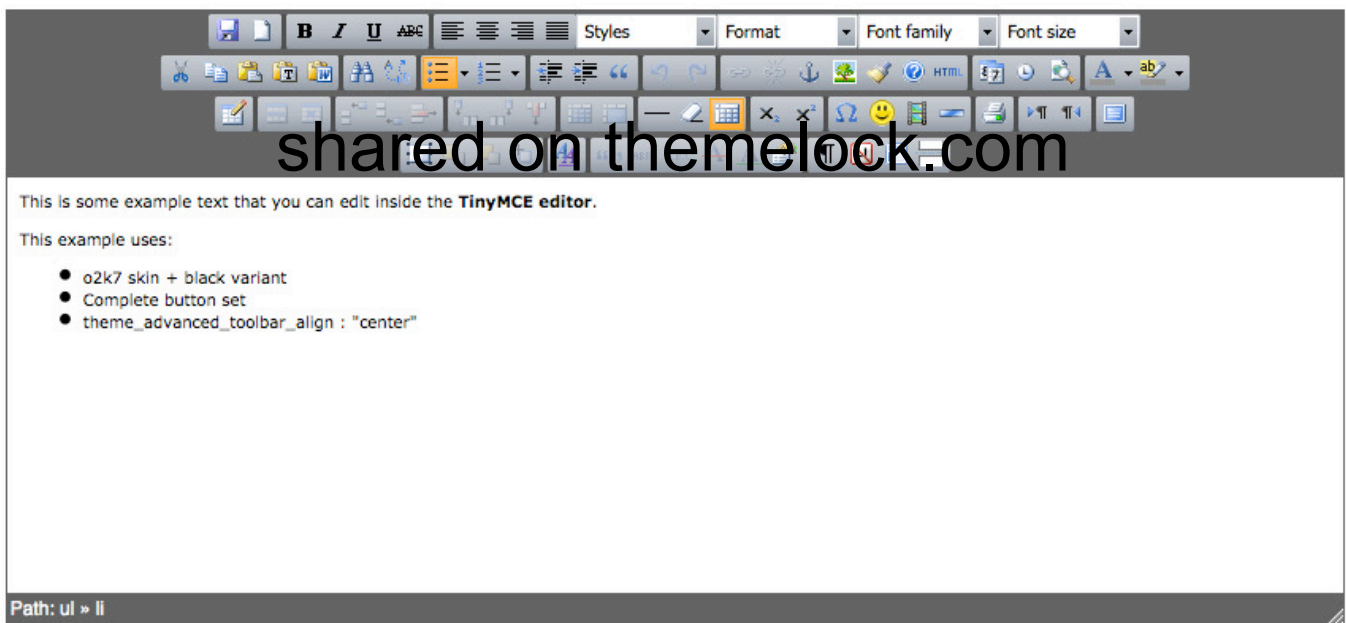
Editor with o2k7 skin + silver variant enabled

```
// General options
mode : "exact",
theme : "advanced",
skin : "o2k7",
skin_variant : "silver"
```



Editor with o2k7 skin + black variant enabled

```
// General options
mode : "exact",
theme : "advanced",
skin : "o2k7",
skin_variant : "black"
```



* Check TinyMCE complete documentation [here](#).

Graph

Create graphs with beautiful lines, bar and charts using jqplot plugin. jqPlot is a jQuery plugin to generate pure client-side javascript charts in your web pages.

Sample Data

We will be using the following table for all our graphs.

	January	February	March	April	May	June	Total
Profit	100	120	59	93	46	62	480
Expenses	82	46	51	130	88	48	445
Sales	29	50	110	115	89	54	447

Lets get start

Creating the HTML code. Add a container (target) to your web page where you want your plot to show up. Be sure to give your target a width and a height:

```
<div id="chartdiv" style="width:100%;height:160px;"></div>
```

Now create a plot. Create the actual plot by calling the \$.jqplot plugin with the id of your target and some data:

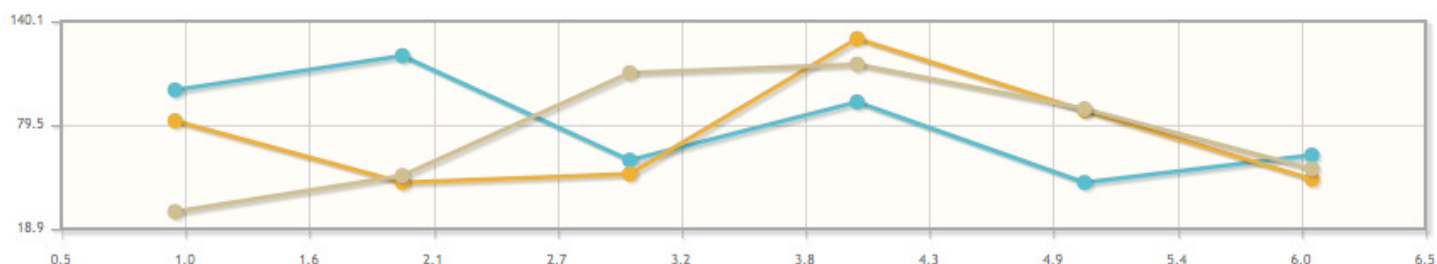
```
Javascript Syntax:  
chart = $.jqplot("targetElemId", [dataArray,...], {optionsObject});
```

Check for complete jqPlot option here

Line Graph and some customization

The most basic jqPlot chart takes a series of data and plots a line. No options need to be supplied.

```
$(document).ready(function(){  
    var profits = [100,120, 59,93,46,62];  
    var expenses = [82,46, 51,130,88,48];  
    var sales = [29,50,110,115,89,54];  
    var tickslabel = ['January', 'February', 'March', 'April','May','June'];  
    var chart1 = $.jqplot ('chartdiv', [profits, expenses, sales]);  
});
```



Now let's add some customization:

The following plot uses a number of options to set the title, add legend, highlighter and shows how to use the CategoryAxisRenderer plugin to render a category style axis, with equal pixel spacing between y data values of a series.

```
var chart2 = $.jqplot('chart2', [profits,expenses,sales], {  
    title:'Line graph with Options',  
    legend:{  
        show:true,  
        labels:['Profits','Expenses','Expenses'],  
        location:'ne'  
    },  
    axes:{  
        xaxis:{
```

```

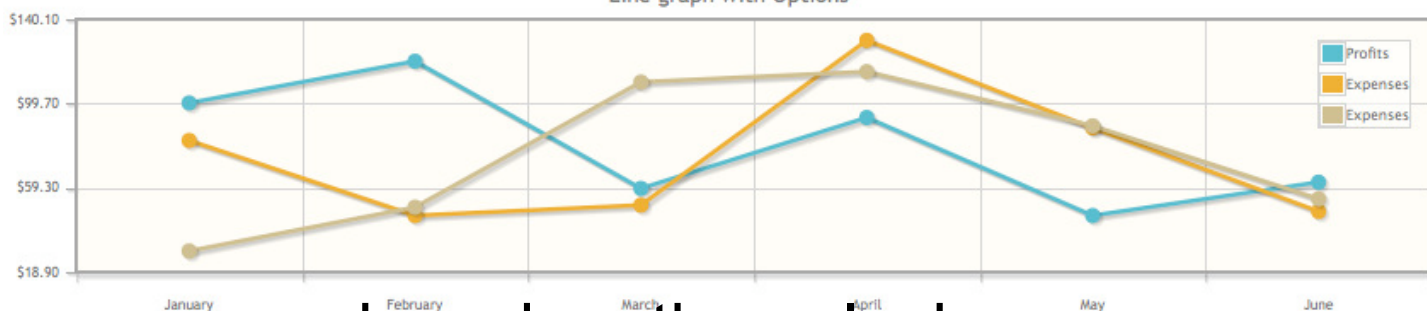
        renderer:$.jqplot.CategoryAxisRenderer ,
        ticks:tickslabel
    },
    yaxis:{
        tickOptions:{
            formatString:'$%.2f'
        }
    }
},
highlighter: {
    show: true,
    sizeAdjust: 7.5,
    tooltipAxes:'y'
},
cursor: {
    show: false
}
});

```

Below charts depend on the following files:

- `<script type="text/javascript" src="../jqplot.categoryAxisRenderer.min.js"></script>`

Line graph with Options



shared on themelock.com

Bar Graph and some customization

Below is a default bar plot. Bars will highlight on mouseover. Events are triggered when you mouseover a bar and also when you click on a bar. Here we capture 'jqplotDataMouseOver' and 'jqplotDataClick'. 'jqplotDataMouseOver' event will fire continuously as the user mouses over the bar. Additionally, a 'jqplotDataUnhighlight' event is fired when the user moves out of a bar (if highlighting is enabled).

```

var chart3 = $.jqplot('chart3',[profits,expenses,sales],{
    title:"Bar graph with 'jqplotDataMouseOver' and 'jqplotDataClick'",
    seriesDefaults: {
        renderer:$.jqplot.BarRenderer,
        pointLabels: { show: true }
    },
    axes: {
        xaxis: {
            renderer: $.jqplot.CategoryAxisRenderer,
            ticks: tickslabel
        },
        yaxis:{
            tickOptions:{
                formatString:'$%.2f'
            }
        }
    }
});

$('#chart3').bind('jqplotDataHighlight',
    function (ev, seriesIndex, pointIndex, data) {
        $('#info1').html(tickslabel[seriesIndex] + ' - '+ data[1] );
    }
);
$('#chart3').bind('jqplotDataClick',
    function (ev, seriesIndex, pointIndex, data) {

```

```

    $('#info2').html(ticklabel[seriesIndex] + ' - ' + data[1]);
  }
);

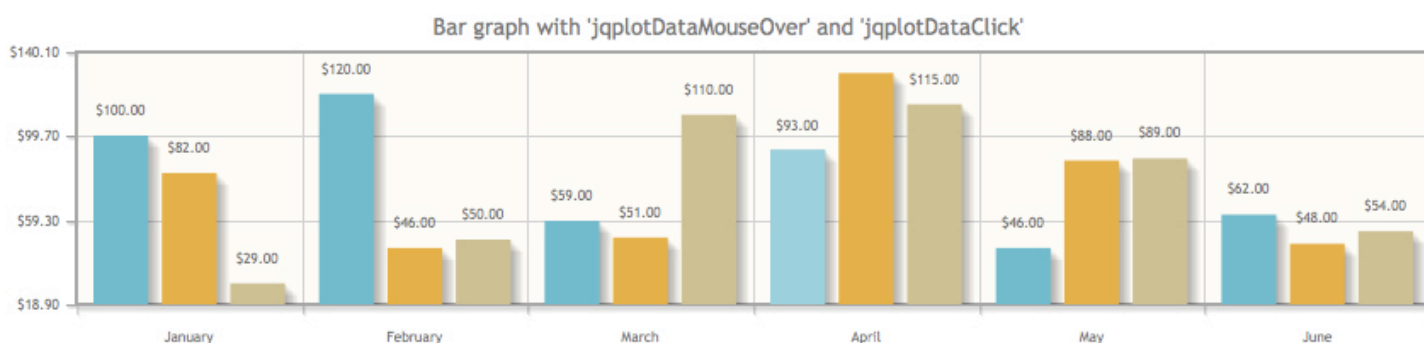
$('#chart3').bind('jqplotDataUnhighlight',
  function (ev) {
    $('#info1').html('');
  }
);

```

Below charts depend on the following files: - `<script type="text/javascript" src="../../jqplot.barRenderer.min.js"></script>` - `<script type="text/javascript" src="../../jqplot.categoryAxisRenderer.min.js"></script>` - `<script type="text/javascript" src="../../jqplot.pointLabels.min.js"></script>`

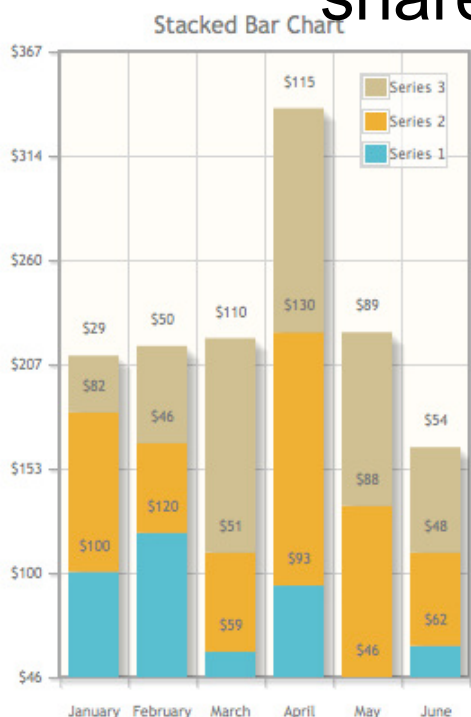
Mouse Over: January - 93

Clicked: March - 110



To create stack bar chart, set the 'stackSeries' option to true like below chart.

You clicked: April Sales : 115



shared on themelock.com

```

var plot3 = $.jqplot('chart4', [profits, expenses, sales], {
  title: "Stacked Bar Chart",
  stackSeries: true, // Tell the plot to stack the bars.
  captureRightClick: true,
  seriesDefaults: {
    renderer: $.jqplot.BarRenderer,
    rendererOptions: { barMargin: 10, highlightMouseDown:
true },
    pointLabels: { show: true }
  },
  axes: {
    xaxis: {
      renderer: $.jqplot.CategoryAxisRenderer,
      ticks: ticklabel
    },
    yaxis: { padMin: 0,
      tickOptions: { formatString: '$%.0f' }
    }
  },
  legend: { show: true, location: 'ne' }
});

$('#chart4').bind('jqplotDataClick',
  function (ev, seriesIndex, pointIndex, data) {
    $('#info3').html(ticklabel[pointIndex] + ' ' +
      ylabel[seriesIndex] + ' : ' + data[1]);
  }
);

```

Let's check on some other samples:

Pie Chart

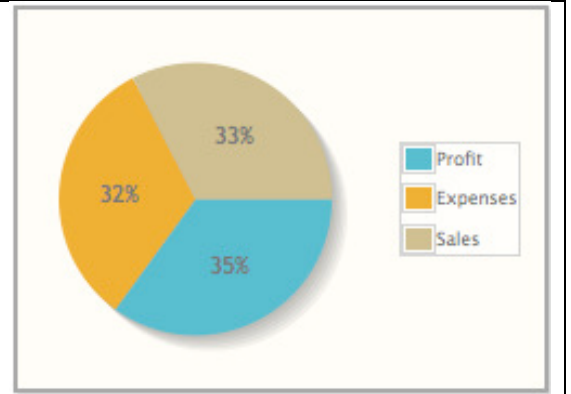
Pie Chart depend on:

– `<script type="text/javascript" src="../../jqplot.pieRenderer.min.js"></script>`

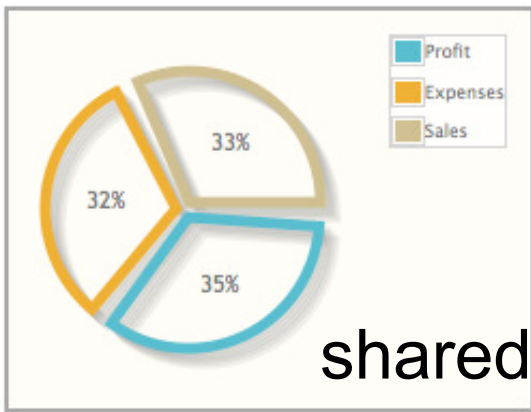
jqPlot bakes up the best pie and donut charts you've ever tasted! Like bar and filled line plots, pie and donut slices highlight when you mouse over.

```
var total = [['Profit', 480], ['Expenses', 445], ['Sales', 447]];

var chart5 = jQuery.jqplot ('chart5', [total], {
  seriesDefaults: {
    renderer: jQuery.jqplot.PieRenderer,
    rendererOptions: { showDataLabels: true }
  },
  legend: {
    show: true,
    location: 'e'
  }
});
```



Too many calories in that pie? Get all the taste without the filling! Highlighting and data labels are still supported. You can even cut out the slices!



shared on themelock.com

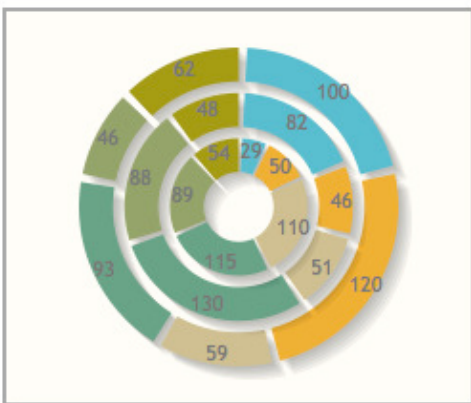
```
var chart6 = jQuery.jqplot ('chart6', [total], {
  seriesDefaults: {
    renderer: jQuery.jqplot.PieRenderer,
    rendererOptions: {
      fill: false, // Turn off filling of slices.
      showDataLabels: true,
      // Add a margin to separate the slices.
      sliceMargin: 4,
      // stroke the slices with a little thicker line.
      lineWidth: 5
    }
  },
  legend: { show: true, location: 'ne' }
});
```

Donut Chart

Donut Chart depend on:

– `<script type="text/javascript" src="../../jqplot.donutRenderer.min.js"></script>`

Coming straight from the same bakery, donut plots have nearly identical options as pie charts.



```
var chart7 = $.jqplot('chart7', [profits, expenses, sales], {
  seriesDefaults: {
    //make this a donut chart.
    renderer: $.jqplot.DonutRenderer,
    rendererOptions: {
      //Donut's can be cut into slices like pies.
      sliceMargin: 3,
      //Pies and donuts can start at any arbitrary angle
      startAngle: -90,
      showDataLabels: true,
      // By default, data labels show the percentage of the
      donut/pie.
      // You can show the data 'value' or data 'label'
      instead.
      dataLabels: 'value'
    }
  }
});
```


Bubble Chart

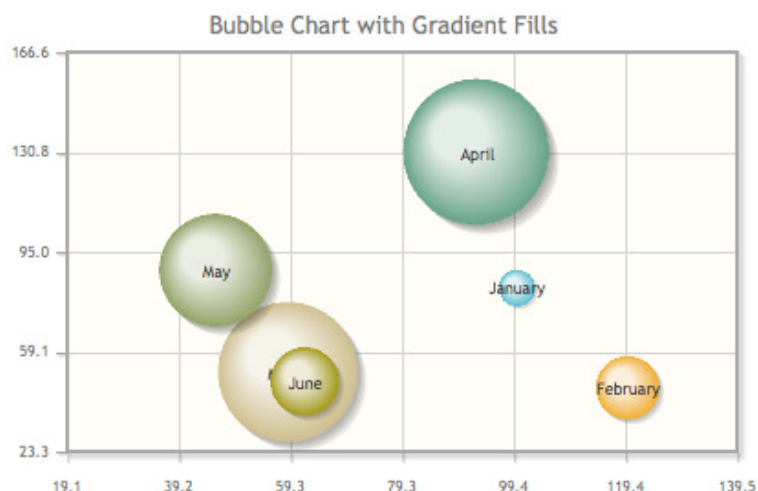
Bubble Chart depend on:

– `<script type="text/javascript" src="../../jqplot.bubbleRenderer.min.js"></script>`

Bubble charts represent 3 dimensional data. First, a basic bubble chart with the "bubbleGradients: true" option to specify gradient fills. Radial gradients are not supported in IE version before IE 9 and will be automatically disabled.

```
var arr = [[100, 82, 29, "January"],
[120, 46, 50, "February"],
[59, 51, 110, "March"],
[93, 130, 115, "April"],
[46, 88, 89, "May"],
[62, 48, 54, "June"]];

var chart8 = $.jqplot('chart8',[arr],{
  title: 'Bubble Chart with Gradient Fills',
  seriesDefaults:{
    renderer: $.jqplot.BubbleRenderer,
    rendererOptions: {
      bubbleGradients: true
    },
    shadow: true
  }
});
```



Also don't forget to include the following:

- javascript 1.4.3 and above
- jquery.jqplot.min.js
- excanvas.js – optionally the excanvas script for IE support (for IE 9 and below)
- jquery.jqplot.min.css

shared on themelook.com

* Check on more examples here or here.

* jqPlot has been tested on IE 6, IE 7, IE 8, Firefox, Safari, and Opera.

Popover

Add customizable overlay on your website. Set the popover header and its content for more elegant look. Pop over can be attach to the click event of any DOM element.

Features:

- iPad like pop over with arrow
- Can be closed using ESC
- Can appear in four sides of the originating element.
- Has option to prevent it from opening in a certain side of the element.

Usage:



HTML:

```
<input type="button" id="popover-button" value="Popover Button"/>
```

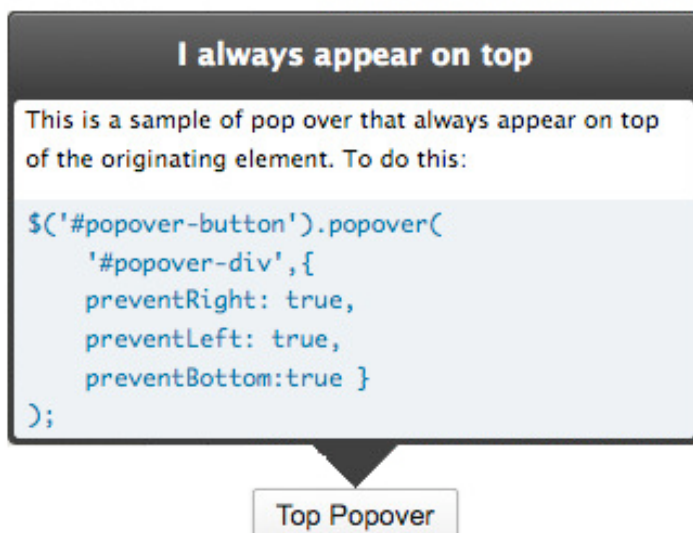
```
<div id="popover-div" class="popover">
  <header>My Header</header>
  <section>
    <div class="content">
      ... content here
    </div>
  </section>
</div>
```

Javascript:

```
$('#popover-button').popover('#popover-div',{Options});
```

shared on themelock.com

The popover can appear in all 4 sides of the element depending on the screen. To prevent the pop over to appear on the right of the originating element, set preventRight to true. Do the same with other position.



Options

Options	Default Value	Description
preventLeft	false (Boolean)	Prevent popover from opening on left side
preventRight	false (Boolean)	Prevent popover from opening on right side
preventTop	false (Boolean)	Prevent popover from opening on top side
preventBottom	false (Boolean)	Prevent popover from opening on bottom side
closeOnEsc	true (Boolean)	Enable/disable keyboard ESC to close popover
padding	18 (Number)	Minimum padding to be kept between popover borders and edge of document
offset	[0,0]	Fixed offset location

Events

Event	Description
onOpen	This callback function is triggered after the popover is opened
onClose	This callback function is triggered after the popover is closed.

Scripting

Method	Syntax	Description
open	<code>\$('#element').data("popover").open();</code>	Programmatically open a popover
close	<code>\$('#element').data("popover").close();</code>	Programmatically close a popover
close	<code>\$.popover.close();</code>	Close the currently open popover without knowing the trigger

shared on themelock.com

Style

Make your Web pages fantastic and creative with these skinned elements

Progress Bars

Vpad theme include a very nice yet simple progress bar that can be used to convey the progress of a certain task.

Basic Progress Bar



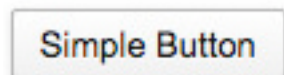
```
<div class="progress"><span style="width: 10%;"></span></div>
```



```
<div class="progress"><span style="width: 50%;"><b>50%</b></span></div>
```

Buttons

Enhance your website with a professional-look button.

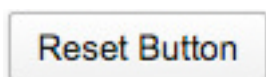


```
<button>Simple Button</button>
```

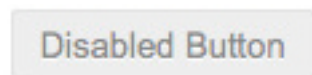
shared on themelock.com



```
<button type="submit">Submit Button</button>
```



```
<button type="reset">Reset Button</button>
```



```
<button disabled>Disabled Button</button>
```



```
<a href="#" class="button">Anchor Button</a>
```

Pagination

Pagination is easy to get right with Vpad pagination. Use 'pagination' class to make a beautiful pagination on your webpages.

```
<ul class="pagination clearfix">
  <li><a href="#">First</a></li>
  <li><a href="#">&laquo;</a></li>
  <li class="current"><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">&raquo;</a></li>
  <li><a href="#">Last</a></li>
</ul>
```

Tab

Creating Javascript driven tab can now be as easy as HTML and css.

First: Create a container div for both the tabs link and the content with a class of 'tabs'.

```
<div class="tabs">
  ... tab link
    <section>
      ... content
    </section>
</div>
```

Second: The tabs themselves are simply links within an unordered list:

```
<ul class="clearfix">
  <li><a href="#">Overview</a></li>
  <li><a href="#">Tab 2</a></li>
  <li><a href="#">Tab 3</a></li>
</ul>
```

shared on themelock.com

Third: Store each tab's content within a section element with the necessary class.

```
<section class="clearfix">
  <header class="alpha omega">
    <h2>Tabbed Content</h2>
  </header>
  <p>This is tab 1. Type here your tab content</p>
</section>
```

Here is a complete code for creating tabs:

```
<div class="tabs leading">
  <ul class="clearfix">
    <li><a href="#">Overview</a></li>
    <li><a href="#">Tab 2</a></li>
    <li><a href="#">Tab 3</a></li>
  </ul>
  <section>
    <section class="clearfix">
      <header class="alpha omega">
        <h2>Tabbed Content</h2>
      </header>
      <p>This is tab 1. Type here your tab content</p>
    </section>
    <section class="clearfix">
      <header class="alpha omega">
        <h2>Tab 2 Header</h2>
      </header>
```

```
</section>
<section class="clearfix">
  <header class="alpha omega">
    <h2>Tab 3 Header</h2>
  </header>
</section>
</section>
</div>
```

Overview

Tab 2

Tab 3

Tabbed Content

This is tab 1. Type here your tab content

Accordion

Let's see how an accordion is put together.

Let's keep the HTML as simple as possible, so we'll use a div element with a class of 'accordion', then we'll list the accordion items using <header> and <section>.

```
<div class="accordion">
  <header>
    <h2>This is an accordion</h2>
  </header>
  <section class="clearfix">
    ... content here
  </section>

  <header>
    <h2>Accordion title here</h2>
  </header>
  <section class="clearfix">
    ... content here
  </section>

  ...
</div>
```

shared on themelock.com

As simple as that. To add more accordion items, just create additional <header> and <section> in a similar fashion:

This is an accordion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed turpis diam, facilisis nec egestas quis, pharetra eget diam. Aliquam sagittis mauris elit, nec pulvinar leo. Duis felis urna.

3 Column Grid inside Accordion

Lorem ipsum dolor

Lorem ipsum dolor

Wizard

Lead your website user through the interface with step by step tasks in a prescribed order.

1. Create Account

2. Contact Information

3. Finalize

Step 1: Account Information

Enter your login information:

1. Enter Your Email Address *

Your password will be sent to this address. Your address will not made public.

2. Pick a username

Your preferred username to be used when logging in.

3. Choose a Password *

Must be at least 8 characters long.

Verify Password *

Proceed »

Javascript

First, add the Javascript file to your <head> tag

```
<script type="text/javascript" src="js/jquery.wizard.js" ></script>
```

HTML Coding

Follow the basic html structure for the wizard just like below. Pages are wrap inside a div with class="items". Each page is divided into three part, <header>, <section>, <footer>

Inside the <section> part holds the main content.

```
<form action="#" novalidate>
  <div class="wizard">
    <!-- navigation -->
    <nav>
      <ul class="clearfix">
        <li class="active"><strong>1.</strong> Create Account</li>
        <li><strong>2.</strong> Contact Information</li>
        <li><strong>3.</strong> Finalize</li>
      </ul>
    </nav>

    <!-- steps / pages -->
    <div class="items">
      <!-- page1 -->
      <section>
        <header> Page title </header>
        <section> Main content </section>
        <footer> Next button </footer>
      </section>

      <!-- page2 -->
      <section>
```

```
        ... page content
    </section>

    <!-- page3 -->
    <section>
        ... page content
    </section>
</div>
</div>
</form>
```

Wizard Setup

Now, your form is set. To turn it into a wizard form, call the wizard method:

```
$(function(){    $('wizard').wizard(); });
```

shared on themelock.com