Creating Beautiful Web Sites Easily Using Bootstrap

Doug Hennig
Stonefield Software Inc.
Email: dhennig@stonefield.com
Corporate Web sites: www.stonefieldquery.com
www.stonefieldsoftware.com
Personal Web site : www.DougHennig.com
Blog: DougHennig.BlogSpot.com
Twitter: DougHennig

Laying out a web page using HTML and CSS can be challenging. Do you use the older table mechanism or CSS floats to place objects side-by-side? How do you deal with differences in browsers? And what about handling different devices: phones, tablets, laptops, and desktops?

Bootstrap is a free, open source framework for developing responsive, mobile-first web sites. It solves many problems web developers typically face and makes it easy to create beautiful web sites in record time, even for inexperienced developers.

This document shows how to get started with Bootstrap, examines using its grid system to easily layout your page elements, and discusses how Bootstrap components add attractive and functional elements to your web site. We'll do a "makeover" of a real web site to show how easy is it to make it more attractive, functional, and mobile-friendly.

Introduction

Web development is getting both more complicated and easier at the same time. More complicated because there are more browsers (and versions of those browsers) and more types of devices (smartphones, tablets, laptops, and desktops), and users expect web sites to act like apps. Easier because there's a variety of frameworks to choose from that make development easier.

Like a framework in a VFP application, a web framework offers advantages over starting a web site from nothing:

- A framework provides code blocks you can use rather than creating them from scratch.
- Pages created using a framework have a consistent look and feel.
- Most frameworks handle the differences between different browsers and their versions, removing the need for you to learn what those differences are and what you have to do about them.
- Frameworks typically provide a responsive interface, automatically adjusting elements based on the device by using media queries (basically, asking the browser the size of the browser window).

Bootstrap is an open source framework that's very popular for developing both web sites and web applications. It was originally created by developers at Twitter as a framework for internal use but released as an open source project in 2011.

There are several reasons to consider using Bootstrap for your web development:

- It's easy to get started, as we'll see in the next section.
- Its grid system makes placing objects in the correct location much simpler than using a strictly CSS approach.
- It makes creating responsive, mobile-ready web sites much easier than doing it by hand.
- It (mostly) transparently handles browser differences.
- It provides attractive, consistent base styling for most HTML elements as we'll see later in this document.
- It has numerous pre-built gadgets such as panels, accordions, dropdowns, and so on that style simple HTML elements like unordered lists to provide attractive visual components for your web site.

There are a few downsides to using Bootstrap, including size (Bootstrap.min.js is 37K, jQuery.min.js [required by Bootstrap] is 84K, and Bootstrap.min.css is 119K) and the fact that due to its popularity, a lot of web sites look the same.

Getting started

Getting started with Bootstrap is easy: if you want to use a CDN (Content Delivery Network) to host the Bootstrap files, you don't have to download a thing! A benefit of using a CDN is that the user may have already downloaded Bootstrap from the CDN by visiting another site using Bootstrap hosted by that CDN. In that case, Bootstrap is loaded from their cache, which is faster than downloading it again. Otherwise, if you want a local copy of the Bootstrap files, do the following:

- Navigate your browser to http://getbootstrap.com and click the Download Bootstrap button to go to the Download page.
- Click the Download Bootstrap button to download the version that consists of compiled and minified CSS, JavaScript, and fonts (currently named bootstrap-3.3.7-dist.zip).
- Unzip the download in a folder of your choice. You'll find three subdirectories, named css, fonts, and js, which contain the CSS, fonts (used for icons rather than text), and JavaScript files that make up Bootstrap.
- Bootstrap requires jQuery (unlike earlier versions, Bootstrap 3.3.7, released July 25, 2016, works with jQuery version 3, so it's much easier to download now), so download it from https://jquery.com/download and save into a folder of your choosing (you may wish to remove the version number from the filename). For example, you might put it into the js subdirectory of the folder you extracted the Bootstrap files to.

If you're using Microsoft Visual Studio, you can use the NuGet Package Manager to download and install Bootstrap.

Whether you're using a CDN or a local copy, you just need to add a few things to an existing web page to enable Bootstrap:

- Specify HTML5 by adding a <doctype> element at the top of the page:

  ```
  <!DOCTYPE html>
  ```

- Although not necessary, it's a good idea to specify the lang attribute on the <html> tag and to include a <meta> tag specifying the character set:

  ```
  <html lang="en">
     <head>
        <meta charset="utf-8">
  ```

- Add the viewport <meta> tag to ensure proper rendering on mobile devices. "width=device-width" sets the width of the page to the width of the device. "initial-scale=1" sets the initial zoom level when the page is first loaded by the browser.

  ```
  <meta name="viewport" content="width=device-width, initial-scale=1">
  ```

- Add a <meta> tag telling Internet Explorer (IE) to display content in the highest mode available:

  ```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  ```

- Specify the Bootstrap CSS file, either from a CDN:

  ```
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/
  css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on
  3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
  ```

  or a local copy:

  ```
  <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
  ```

  (Specify the correct path and filename in the href attribute.)

- If some of your users use IE 8, add the following to support HTML5 elements and media queries:

  ```
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js">
  </script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]-->
  ```

- Before the closing body tag, add the jQuery and Bootstrap JavaScript files, either from a CDN:

  ```
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/
  jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/
  bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2
  mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>
  ```

  or a local copy:

  ```
  <script src="bootstrap/js/jquery.min.js"></script>
  <script src="bootstrap/js/bootstrap.min.js"></script>
  ```

  (Specify the correct path and filename in the href attributes.)

Listing 1 shows a basic Bootstrap page using a CDN which only shows “Hello, world!” The colored sections shows the tags added for Bootstrap.

Listing 1. A basic Bootstrap page.

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <!-- The above 3 meta tags *must* come first in the head; any other head
```

```
       content must come *after* these tags -->
       <title>My First Bootstrap Page</title>

       <!-- Bootstrap -->
       <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/
          css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUc
          ww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

       <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
       queries -->
       <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
       <!--[if lt IE 9]>
          <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js">
          </script>
          <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
       <![endif]-->
   </head>
   <body>
       <h1>Hello, world!</h1>

       <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
       <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/
          jquery.min.js"></script>
       <!-- Include all compiled plugins (below), or include individual files as
       needed -->
       <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/
          bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZx
          UPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>
   </body>
</html>
```

Of course, a simple "Hello world" page doesn't show any of Bootstrap's features, so let's dig in. We'll start by looking at Bootstrap's grid system.

Bootstrap's grid system

Bootstrap makes it easy to place elements beside each other because it's based on a grid. The grid allows up to twelve columns across the page. If you don't need twelve columns, you can group them together to make fewer but wider columns. For example, you could have a row with four columns grouped into one and eight columns grouping into another, for a total of two columns, one twice as big as the other. **Figure 1** shows three examples of differing number of columns. Note that the total of each row has to add up to twelve.

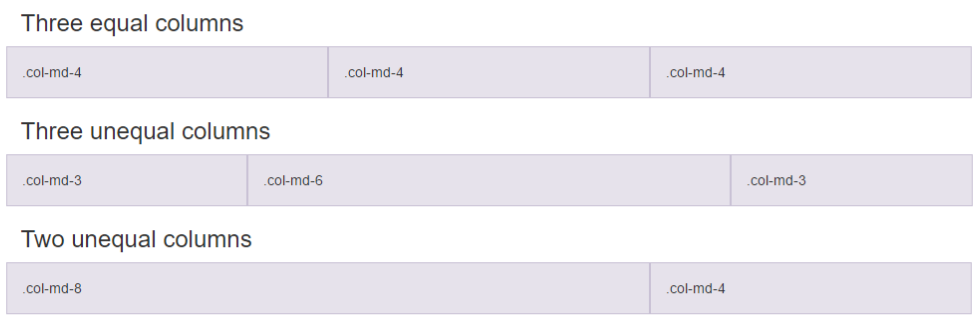


Figure 1. Bootstrap's grid system makes it easy to create columns of any size.

The grid system consists of four sizes:

- xs: extra-small (less than 768 pixels wide), for phones
- sm: small (768 – 991 pixels), for tablets
- md: medium (992 – 1199 pixels), for laptops and desktops
- lg: large (1200 or more pixels), for larger desktops

These sizes define what happens to the grid as the browser scales up or down. Each size scales up, so, for example, specifying "sm" defines what happens on tablets and higher.

Bootstrap has pre-defined classes that combine these sizes with the number of columns to use. You can see some of these in Figure 1: col-md-3, col-md-4, col-md-6, and col-md-8 for a three-, four-, six-, and eight-column wide cells, respectively. On laptops and larger desktops, the grid is laid out horizontally because the "md" size is specified. On tablets and phones, or smaller browser windows, the grid is laid out vertically, with the cells stacked; see **Figure 2** to see what happens when the browser displaying the page shown in Figure 1 is resized to less than 992 pixels wide.

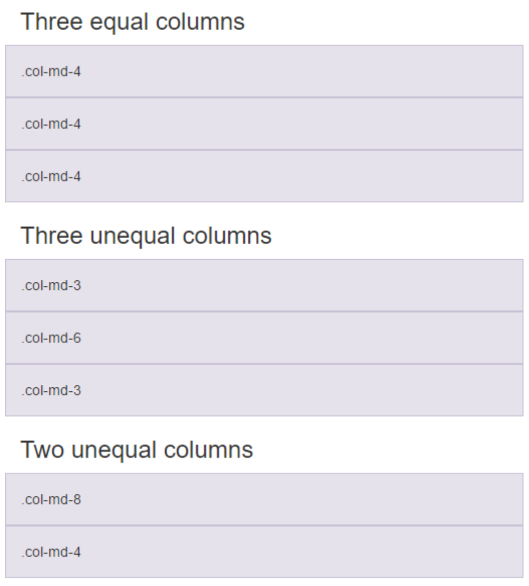


Figure 2. The page from **Figure 1** when the browser is resized to less than 992 pixels wide.

For best results, put the grid inside a <div> using the “container” (fixed-width: auto for xs, 750 for sm, 970 for md, and 1170 for lg) or “container-fluid” (taking the full width of the browser window) class so alignment and padding are correct, and use <div> elements with the “row” class to create horizontal groups of columns. For example, **Listing 2** shows the HTML that produces the grid layout in **Figure 1**.

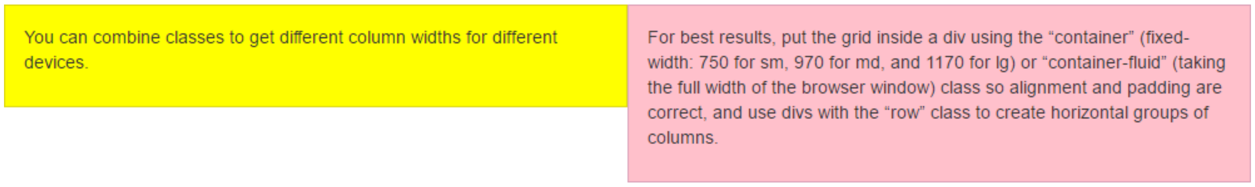
Listing 2. This HTML produces the grid layout in **Figure 1**.

```
<div class="container">
   <h3>Three equal columns</h3>
   <div class="row">
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
   </div>

   <h3>Three unequal columns</h3>
   <div class="row">
      <div class="col-md-3">.col-md-3</div>
      <div class="col-md-6">.col-md-6</div>
      <div class="col-md-3">.col-md-3</div>
   </div>

   <h3>Two unequal columns</h3>
   <div class="row">
      <div class="col-md-8">.col-md-8</div>
      <div class="col-md-4">.col-md-4</div>
   </div>
</div>
```

You can combine classes to get different column widths for different devices. For example, the HTML in **Listing 3** creates two columns with a 50-50 split on medium and larger devices (col-md-6 for both) and a 25-75 split on smaller devices (col-sm-3 for the left column and col-sm-9 for the right). **Figure 3** and **Figure 4** show the results in medium and small browser windows, respectively.

Listing 3. This creates two columns with a 50-50 split on medium and larger devices and a 25-75 split on smaller devices.

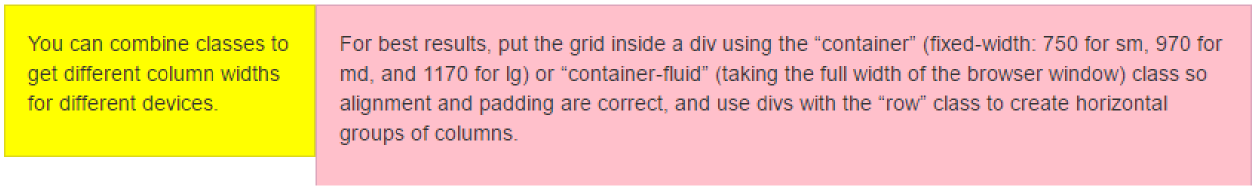
```
<div class="container-fluid">
   <h1>Hello World!</h1>
   <div class="row">
      <div class="col-sm-3 col-md-6" style="background-color:yellow;">
         <p>Lorem ipsum...</p>
      </div>
      <div class="col-sm-9 col-md-6" style="background-color:pink;">
         <p>Sed ut perspiciatis...</p>
      </div>
   </div>
</div>
```



You can combine classes to get different column widths for different devices.

For best results, put the grid inside a div using the “container” (fixed-width: 750 for sm, 970 for md, and 1170 for lg) or “container-fluid” (taking the full width of the browser window) class so alignment and padding are correct, and use divs with the “row” class to create horizontal groups of columns.

Figure 3. The results in a medium browser.



You can combine classes to get different column widths for different devices.

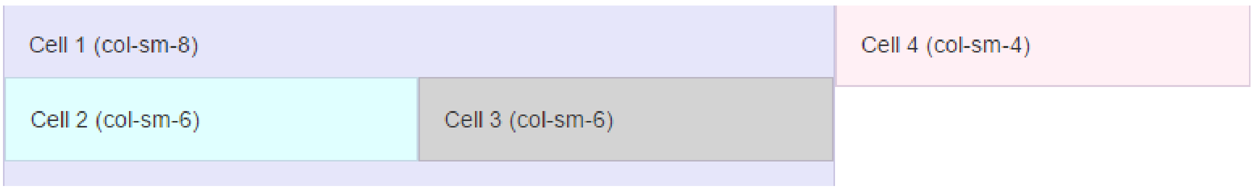
For best results, put the grid inside a div using the “container” (fixed-width: 750 for sm, 970 for md, and 1170 for lg) or “container-fluid” (taking the full width of the browser window) class so alignment and padding are correct, and use divs with the “row” class to create horizontal groups of columns.

Figure 4. The results in a small browser.

You can nest columns inside columns by simply adding a row <div> inside a col <div>. **Listing 4** creates the page shown in **Figure 5**.

Listing 4. This HTML creates the nested columns shown in Figure 5.

```
<div class="container-fluid">
   <div class="row">
      <div class="col-sm-8" style="background-color:lavender;">Cell 1 (col-sm-8)
         <div class="row">
            <div class="col-sm-6" style="background-color:lightcyan;">Cell 2
               (col-sm-6)</div>
            <div class="col-sm-6" style="background-color:lightgray;">Cell 3
               (col-sm-6)</div>
         </div>
      </div>
      <div class="col-sm-4" style="background-color:lavenderblush;">Cell 4
         (col-sm-4)</div>
   </div>
</div>
```



Cell 1 (col-sm-8)

Cell 4 (col-sm-4)

Cell 2 (col-sm-6)

Cell 3 (col-sm-6)

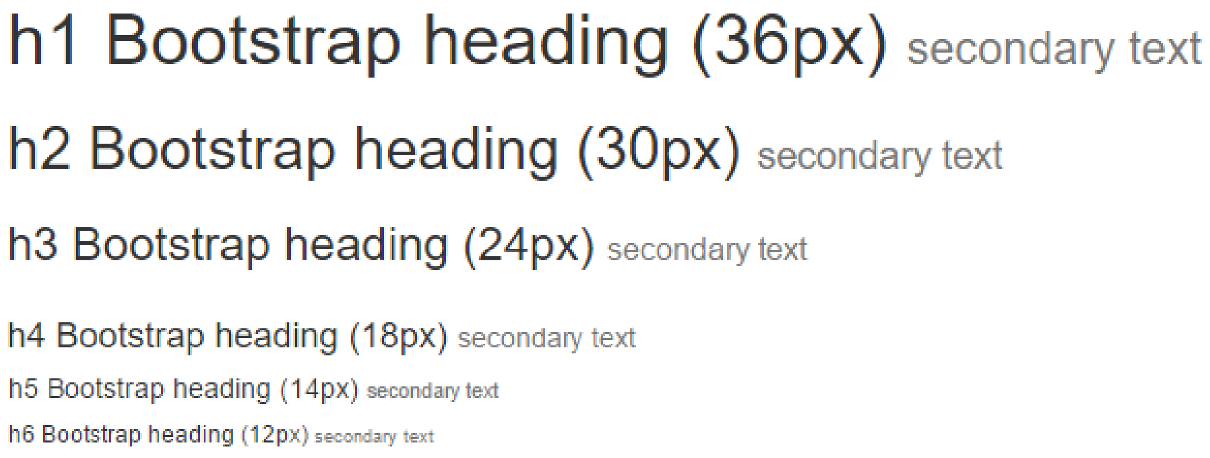
Figure 5. Nested columns are easy to create.

If you want to indent an element by a certain number of columns, use one of the “col-*-offset-*” classes, such as “col-sm-offset-2” to indent by two columns.

Text

Bootstrap’s default font size is 14px (pixels), with a line height of 1.428. <p> elements have a bottom margin that equals half their computed line height (10px by default).

Figure 6 shows the sizes of <h> elements. You can also add secondary text, which displays as lighter and smaller, using <small>.


h1 Bootstrap heading (36px) secondary text

h2 Bootstrap heading (30px) secondary text

h3 Bootstrap heading (24px) secondary text

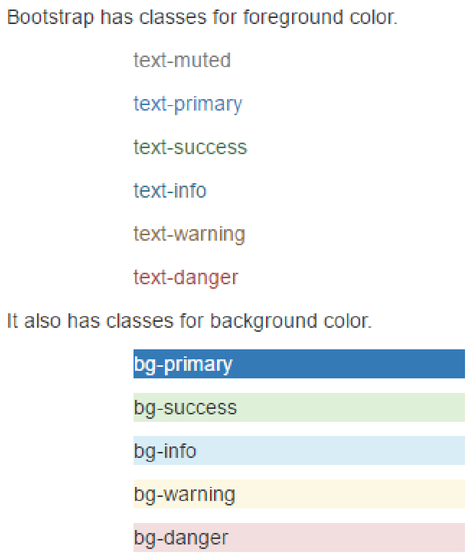
h4 Bootstrap heading (18px) secondary text

h5 Bootstrap heading (14px) secondary text

h6 Bootstrap heading (12px) secondary text

Figure 6. The sizes of Bootstrap headings.

Bootstrap has classes for both contextual foreground and background color. **Figure 7** shows example of these. To use these colors, specify the desired class in a <span> or other element; for example, “<p>This is <span class="text-primary">special text</span>.</p>.”


Bootstrap has classes for foreground color.

text-muted

text-primary

text-success

text-info

text-warning

text-danger

It also has classes for background color.

bg-primary

bg-success

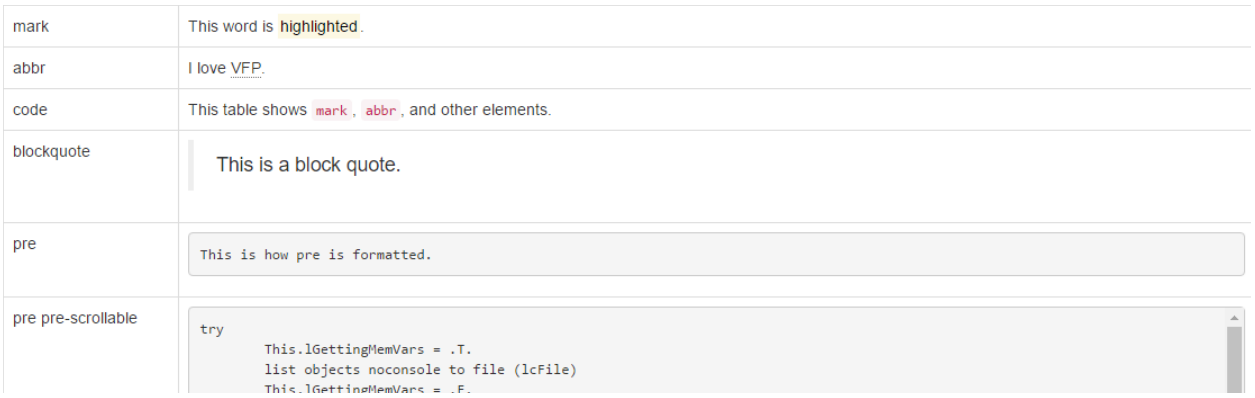
bg-info

bg-warning

bg-danger

Figure 7. Bootstrap has classes for both contextual foreground and background color.

Bootstrap has other elements that provide additional styling choices. **Figure 8** shows a few of them; see http://getbootstrap.com/css/#type for a more complete list.



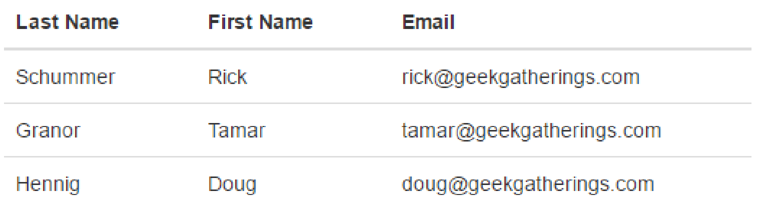
mark	This word is highlighted.
abbr	I love VFP.
code	This table shows `mark`, `abbr`, and other elements.
blockquote	This is a block quote.
pre	`This is how pre is formatted.`
pre pre-scrollable	see below

```
try
        This.lGettingMemVars = .T.
        list objects noconsole to file (lcFile)
        This.lGettingMemVars = .F.
```

Figure 8. Other elements provide additional styling choices.

Tables

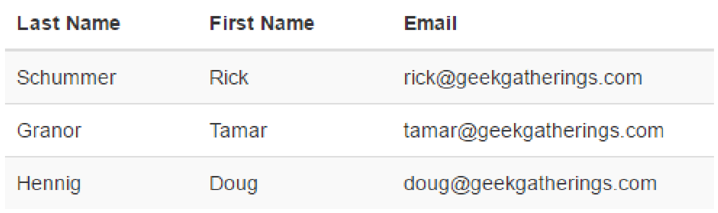
The "table" class adds basic styling (light padding and horizontal dividers) to tables, see **Figure 9**. For example, "<table class="table">."



Last Name	First Name	Email
Schummer	Rick	rick@geekgatherings.com
Granor	Tamar	tamar@geekgatherings.com
Hennig	Doug	doug@geekgatherings.com

Figure 9. The table class provides basic table styling.

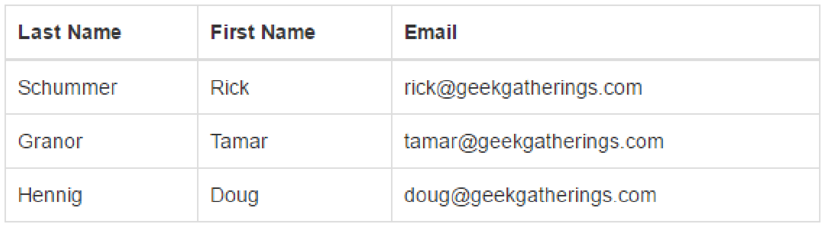
The "table-striped" class provides alternating bands to table rows; see **Figure 10** for an example. Note that you use both the "table" and "table-striped" classes; for example, "<table class="table table-striped">."



Last Name	First Name	Email
Schummer	Rick	rick@geekgatherings.com
Granor	Tamar	tamar@geekgatherings.com
Hennig	Doug	doug@geekgatherings.com

Figure 10. The table-striped class provides alternating bands to table rows.

The table-bordered class adds borders to the table and cells. **Figure 11** shows an example of "<table class="table table-bordered">."

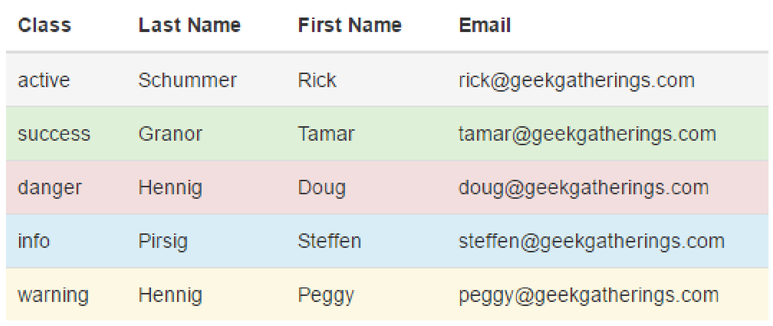


Last Name	First Name	Email
Schummer	Rick	rick@geekgatherings.com
Granor	Tamar	tamar@geekgatherings.com
Hennig	Doug	doug@geekgatherings.com

Figure 11. The table-bordered class adds borders to the table and cells.

The table-hover class highlights rows as the mouse moves over them. I didn't include a figure to show this because it's a dynamic effect you have to see in a browser.

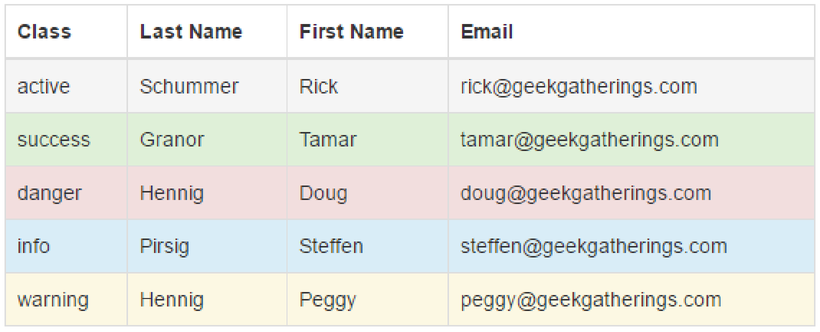
As you can see in **Figure 12**, like text, table rows can use contextual colors. Specify one of the contextual color classes on the <tr> element.



Class	Last Name	First Name	Email
active	Schummer	Rick	rick@geekgatherings.com
success	Granor	Tamar	tamar@geekgatherings.com
danger	Hennig	Doug	doug@geekgatherings.com
info	Pirsig	Steffen	steffen@geekgatherings.com
warning	Hennig	Peggy	peggy@geekgatherings.com

Figure 12. Specify one of the contextual color classes on the <tr> element.

These classes can be combined as desired. The table shown in **Figure 13** uses "table-hover," "table-bordered," and contextual colors.



Class	Last Name	First Name	Email
active	Schummer	Rick	rick@geekgatherings.com
success	Granor	Tamar	tamar@geekgatherings.com
danger	Hennig	Doug	doug@geekgatherings.com
info	Pirsig	Steffen	steffen@geekgatherings.com
warning	Hennig	Peggy	peggy@geekgatherings.com

Figure 13. This table uses table-hover, table-bordered, and contextual colors.

Buttons

You can use button-related classes, the main one being "btn," on <a>, <button>, or <input> elements to create buttons, although Bootstrap recommends using <button>. If you use <a>, you should also add "role="button"" to the element.

For a basic button, use the "btn-default" class. For example, this HTML results in the buttons shown in **Figure 14**.

```
<a class="btn btn-default" href="#" role="button">Link</a>
<button class="btn btn-default" type="button">Button</button>
<button class="btn btn-default" type="submit">Submit</button>
<input class="btn btn-default" type="button" value="Input">
<input class="btn btn-default" type="submit" value="Submit">
```

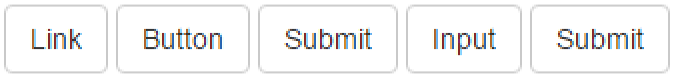


Figure 14. Basic buttons using the btn-default class.

You can style buttons using the btn-primary, btn-success, btn-info, btn-warning, btn-danger, and btn-link classes or disable a button with the "disabled="disabled"" attribute (**Figure 15**). For example, this HTML creates the right-most button in Figure 15:

```
<button type="button" class="btn btn-primary" disabled="disabled">Disabled</button>
```

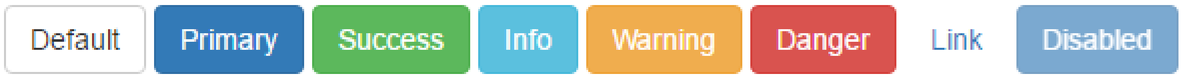


Figure 15. Create styled buttons using btn-* style classes or disable a button using the disabled attribute.

You can create larger or smaller buttons using btn-lg, btn-md (the default if no size is specified), btn-sm, and btn-xs (**Figure 16**). For example, this creates the large button:

```
<button type="button" class="btn btn-default btn-lg">Large button</button>
```

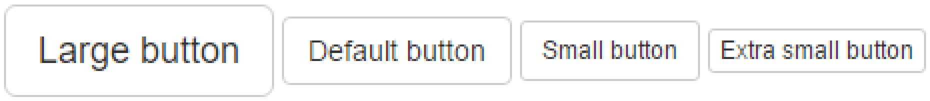


Figure 16. Specify button size using btn-* size classes.

You can group buttons together by putting them in a <div class="btn-group"> for a horizontal group or <div class="btn-group-vertical"> for a vertical group. The HTML in **Listing 5** creates the two button groups shown in **Figure 17**.

Listing 5. This HTML creates horizontal and vertical button groups.

```
<div class="btn-group">
   <button type="button" class="btn btn-primary">C#</button>
   <button type="button" class="btn btn-primary">VB</button>
   <button type="button" class="btn btn-primary">VFP</button>
</div>
<p/>
<div class="btn-group-vertical">
   <button type="button" class="btn btn-primary">C#</button>
   <button type="button" class="btn btn-primary">VB</button>
   <button type="button" class="btn btn-primary">VFP</button>
</div>
```

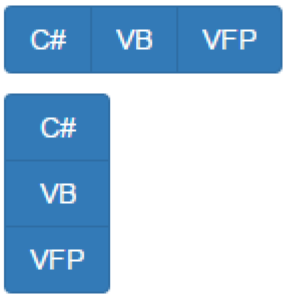


Figure 17. You can create horizontal and vertical button groups.

You can create a split button (one that acts like a button but also includes a dropdown menu) like the one shown in **Figure 18** using HTML like that in **Listing 6**. Note that the menu is an unordered list. Use the "divider" class to create a divider or "disabled" for a disabled menu item.

Listing 6. This HTML creates a split button.

```
<div class="btn-group">
   <button type="button" class="btn btn-primary">VFP</button>
   <button type="button" class="btn btn-primary dropdown-toggle"
      data-toggle="dropdown">
      <span class="caret"></span>
   </button>
   <ul class="dropdown-menu" role="menu">
      <li><a href="#">VFP 8.0</a></li>
      <li><a href="#">VFP 9.0</a></li>
      <li role="separator" class="divider"></li>
      <li><a href="#">VFPX</a></li>
   </ul>
</div>
```

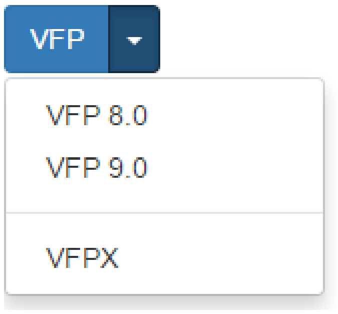


Figure 18. A split button acts like a button but includes a dropdown menu.

Glyphicons

Bootstrap includes 260 glyphicons as a font. You can use them in lots of places, including text, buttons, links, menus, and so on. To use a glyphicon, specify the "glyphicon" class followed by the name of the icon (for example, "class="glyphicon glyphicon-ok"" for a checkmark). See http://getbootstrap.com/components/#glyphicons for a complete list of the glyphicons and their names. Using glyphicons rather than images means fewer files for the user's browser to download.

Figure 19 shows the page created by the HTML in **Listing 7**.

Listing 7. This HTML uses glyphicons in text, a link, and a button.

```
<p>glyphicon-ok in text: <span class="glyphicon glyphicon-ok"></span></p>
<p>glyphicon-envelope as a link: <a href="#"><span class="glyphicon
   glyphicon-envelope"></span></a></p>
<p>glyphicon-search on a button:
   <button type="button" class="btn btn-info">
      <span class="glyphicon glyphicon-search"></span> Search
   </button>
</p>
```

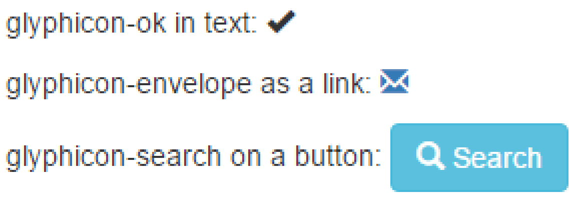


Figure 19. Glyphicons can be used in lots of places, including text, links, and buttons.

Images

Use the “img-rounded” class to add rounded corners to an image, “img-circle” to draw the image as a circle, or “img-thumbnail” to draw it as a thumbnail. See **Figure 20** for an example.

Use the “img-responsive” class to create images that automatically scale to fit their parent element.

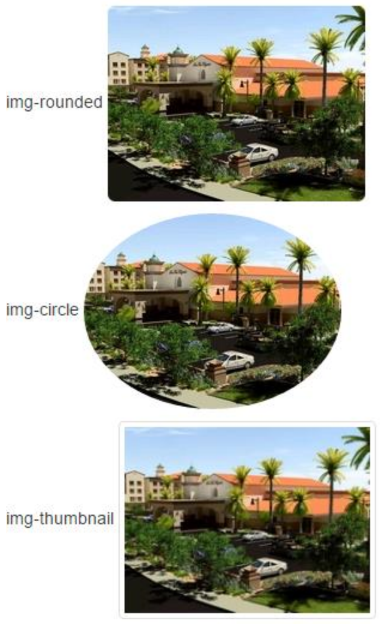


Figure 20. You can style images using img-* classes.

Form controls

Bootstrap supports input, textarea, checkbox, radio, and select controls. Use the "form-control" class for <input>, <textarea>, and <select> to apply a default styling and a width of 100%.

Bootstrap supports all HTML5 <input> types: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, and color. Note that the type attribute must be set properly for styling to work. For example, this creates text and password controls:

```
<input type="text" class="form-control" placeholder="User name">
<input type="password" class="form-control" placeholder="Password">
```

The following creates a textarea:

```
<textarea class="form-control" rows="3" placeholder="Text area"></textarea>
```

Figure 21 shows unstyled and styled input and textarea controls.

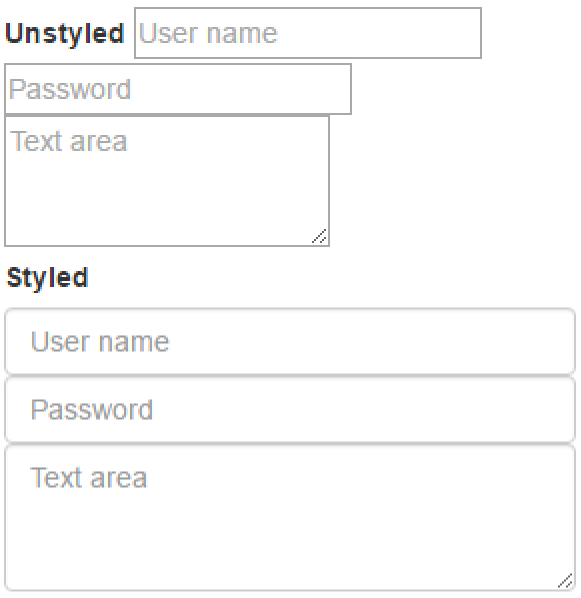


Figure 21. Unstyled and style input and textarea controls.

Use a <div class="checkbox"> containing a label and <input type="checkbox"> to create a checkbox. For a disabled checkbox, add the "disabled" class to the <div> and the disabled attribute to the <input>:

```
<div class="checkbox">
   <label><input type="checkbox" value="">Remember me</label>
</div>
<div class="checkbox disabled">
   <label><input type="checkbox" value="" disabled>Disabled option</label>
</div>
```

For inline checkboxes, you don't need the <div>; use the "checkbox-inline" class for the label:

```
<label class="checkbox-inline"><input type="checkbox" value="">Remember me</label>
<label class="checkbox-inline disabled"><input type="checkbox" value=""
   disabled>Disabled option</label>
```

Figure 22 shows unstyled, styled, and inline checkboxes.

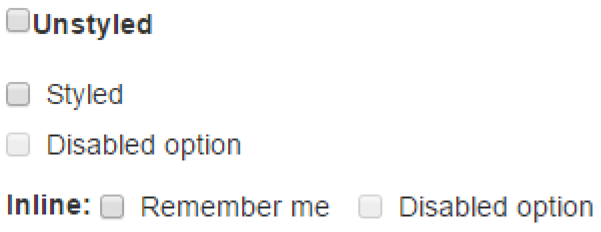


Figure 22. Unstyled, styled, and inline checkboxes.

Use a <div class="radio"> containing a label and <input type="radio"> to create a radio button. All radio buttons in a group must have the same value for the name attribute. For a disabled radio button, add the "disabled" class to the <div> and the disabled attribute to the <input>:

```
<div class="radio">
   <label><input type="radio" name="radio1">Option 1</label>
</div>
<div class="radio">
   <label><input type="radio" name="radio1">Option 2</label>
</div>
<div class="radio disabled">
   <label><input type="radio" name="radio1" disabled>Disabled option</label>
</div>
```

For inline radio buttons, use the "radio-inline" class:

```
<label class="radio-inline"><input type="radio" name="radio2">Option 1</label>
<label class="radio-inline"><input type="radio" name="radio2">Option 2</label>
<label class="radio-inline disabled"><input type="radio" name="radio2"
disabled>Disabled option</label>
```

Figure 23 shows unstyled, styled, and inline radio buttons.

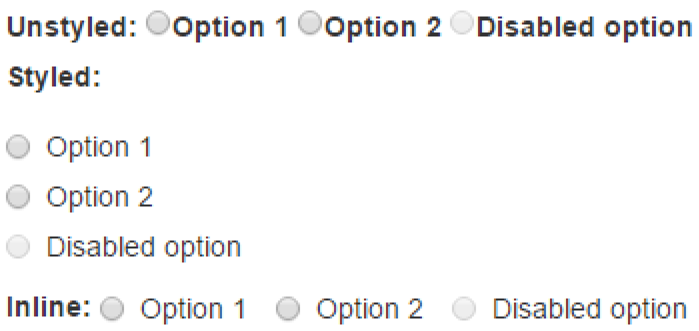


Figure 23. Unstyled, styled, and inline radio buttons.

Use the "form-control" class to style a <select> control:

```
<select class="form-control">
   <option>Choice 1</option>
   <option>Choice 2</option>
   <option>Choice 3</option>
</select>
```

Figure 24 shows unstyled and styled select controls.

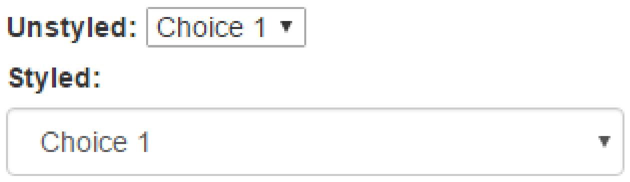


Figure 24. Unstyled and styled select controls.

You can extend <input> controls by adding text or buttons before, after, or on both sides. To do that, use <div class="input-group"> to wrap the elements and "input-group-addon" or "input-group-btn" on the add-on elements. **Listing 8** creates the page shown in **Figure 25.**

Listing 8. This HTML extends <input> controls using input groups.

```
<div class="input-group">
   <input type="text" class="form-control" placeholder="Username">
   <span class="input-group-addon">@swfox.net</span>
</div>

<div class="input-group">
   <span class="input-group-addon">$</span>
   <input type="text" class="form-control">
   <span class="input-group-addon">.00</span>
</div>

<div class="input-group">
   <input type="text" class="form-control" placeholder="Search for...">
   <span class="input-group-btn">
      <button class="btn btn-default" type="button">Go!</button>
   </span>
</div>
```

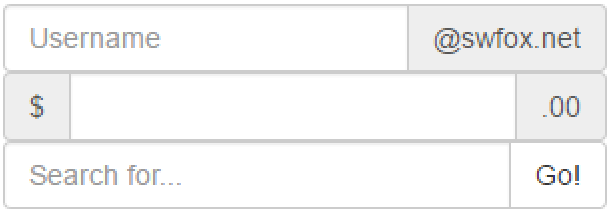


Figure 25. You can extend <input> controls using input groups.

Forms

Bootstrap has three types of form layouts: vertical (the default), horizontal, and inline. Forms work best when specified as <form role="form"> and when <div class="form-group"> wraps labels and controls.

The HTML shown in **Listing 9** creates the vertical form displayed in **Figure 26**.

Listing 9. This HTML creates a vertical form.

```
<form role="form">
   <div class="form-group">
      <label for="email1">Email address</label>
      <input type="email" class="form-control" id="email1" placeholder="Email">
   </div>
   <div class="form-group">
      <label for="password1">Password</label>
      <input type="password" class="form-control" id="password1"
      placeholder="Password">
   </div>
   <div class="checkbox">
      <label><input type="checkbox">Remember me</label>
   </div>
   <div class="form-group">
      <label for="file">File input</label>
      <input type="file" id="file">
   </div>
   <button type="submit" class="btn btn-default">Submit</button>
</form>
```

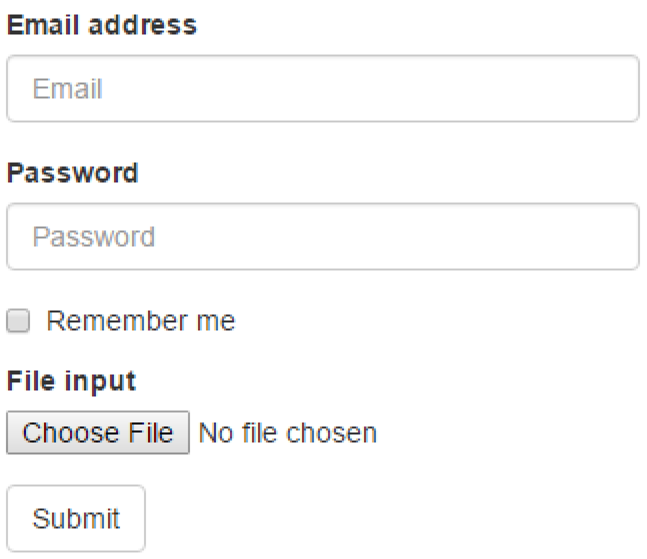


Figure 26. Bootstrap vertical form.

For an inline form, use the "form-inline" class on the <form> element. This causes the width for the controls to be set to "auto." Note that this only applies to forms within viewports that are at least 768px wide. **Figure 27** uses the same HTML as **Figure 26** except for the "form-inline" class.

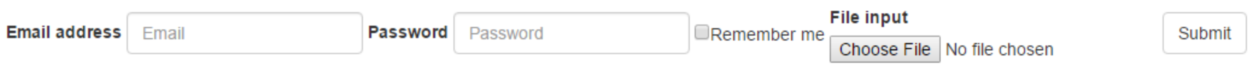


Figure 27. Bootstrap inline form.

For a horizontal form, use the "form-horizontal" class on the <form> element and "control-label" on <label> elements. You can also use grid classes to align labels and controls as desired. For example, the HTML in **Listing 10** uses col-sm-2 and col-sm-10 to create a grid for the controls and col-sm-offset-2 to indent the checkbox and submit button (see **Figure 28**).

Listing 10. This HTML creates a horizontal form.

```
<form role="form" class="form-horizontal">
   <div class="form-group">
      <label class="control-label col-sm-2" for="email3">Email address</label>
      <div class="col-sm-10">
         <input type="email" class="form-control" id="email3" placeholder="Email">
      </div>
   </div>
   <div class="form-group">
      <label class="control-label col-sm-2" for="password3">Password</label>
      <div class="col-sm-10">
         <input type="password" class="form-control" id="password3"
         placeholder="Password">
      </div>
   </div>
   <div class="form-group">
```

```
      <div class="col-sm-offset-2 col-sm-10">
         <div class="checkbox">
            <label><input type="checkbox">Remember me</label>
         </div>
      </div>
   </div>
   <div class="form-group">
      <label class="control-label col-sm-2" for="file3">File input</label>
      <div class="col-sm-10">
         <input type="file" id="file3">
      </div>
   </div>
   <div class="form-group">
      <div class="col-sm-offset-2 col-sm-10">
         <button type="submit" class="btn btn-default">Submit</button>
      </div>
   </div>
</form>
```

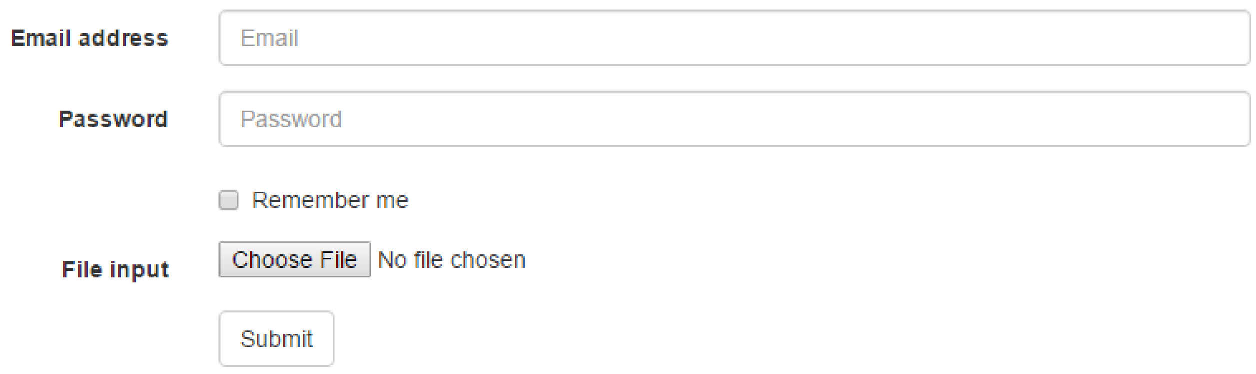


Figure 28. Bootstrap horizontal form.

If you use the placeholder attribute to display text inside an <input> control, be sure to include a label even though one isn't required. Otherwise, a screen reader for visually disabled users won't properly read the contents of the page. You can hide the labels using the "sr-only" class. **Listing 11** creates the form shown in **Figure 29**.

Listing 11. This HTML creates a form with hidden labels.

```
<form role="form">
   <div class="form-group">
      <label class="sr-only" for="email4">Email address</label>
      <input type="email" class="form-control" id="email4" placeholder="Email">
   </div>
   <div class="form-group">
      <label class="sr-only for="password4">Password</label>
      <input type="password" class="form-control" id="password4"
      placeholder="Password">
   </div>
   <button type="submit" class="btn btn-default">Submit</button>
</form>
```

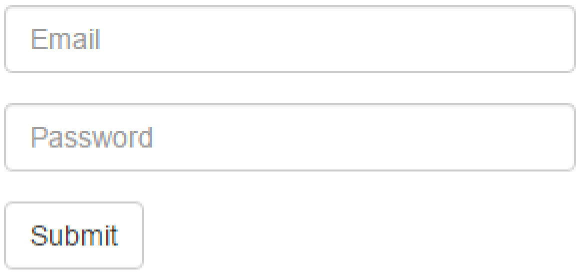


Figure 29. This form has hidden labels that'll be properly handled by a screen reader.

Navbar

A navbar is a navigation component, the most common example of which is a menu bar. However, a navbar can contain any controls you wish, not just dropdowns with links to other pages.

A basic navbar is created using <nav class="navbar navbar-default">. Inside that is a container div. Inside the div are an optional div for the navbar header and an unordered list for the menu items.

The navbar header can contain anything you wish, but it should use a class of "navbar-brand". For example, this puts a link in the header:

```
<a class="navbar-brand" href="#">NavBar Example</a>
```

The header in **Listing 12** uses an image, shown in **Figure 30**.

Listing 12. This HTML creates a basic navbar.

```
<nav class="navbar navbar-default">
   <div class="container-fluid">
      <div class="navbar-header">
         <img class="navbar-brand" src="KokoWhite.jpg">
      </div>
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">Home</a></li>
         <li><a href="BootstrapForms.html">Forms</a></li>
         <li><a href="BootstrapTables.html">Tables</a></li>
         <li><a href="BootstrapText.html">Text</a></li>
      </ul>
   </div>
</nav>
```

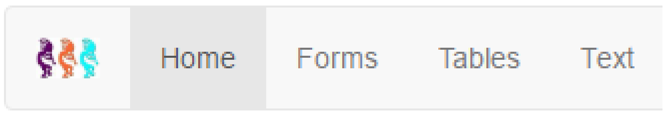


Figure 30. This is a basic navbar.

Note in Figure 30 that "Home" appears selected. That's because of the "active" class on the <li> element.

Use “navbar-inverse” instead of “navbar-default” to create an inverse navbar (**Figure 31**).

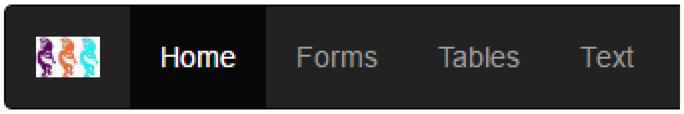


Figure 31. An inverse navbar.

These navbars can scroll off the top as the page is scrolled, which isn’t normal behavior for a menu. To prevent this, add the “navbar-fixed-top” class to the <navbar> element (use “navbar-fixed-bottom” to place the navbar at the bottom of the page).

To place an item at the right edge of the menu, close the first <ul> element and add another one with <ul class="nav navbar-nav navbar-right">. The <li> items that appear inside the <ul> are placed at the right.

Typically, the navbar contains dropdown items that display list of choices. To do that, use this for the menu item:

```
<li class="dropdown">
   <a class="dropdown-toggle" data-toggle="dropdown" href="#">
      Menu Text <span class="caret"></span>
   </a>
```

Inside the <li class="dropdown"> element, include another unordered list with a class of “dropdown-menu” for the menu items. **Listing 13** creates the page shown in **Figure 32**.

Listing 13. This HTML creates a dropdown menu bar.

```
<nav class="navbar navbar-default navbar-fixed-top">
   <div class="container-fluid">
      <div class="navbar-header">
         <img class="navbar-brand" src="images/KokoWhite.jpg">
      </div>
      <ul class="nav navbar-nav">
         <li><a href="index.aspx">Home</a></li>
         <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown">Information
            <span class="caret"></span></a>
            <ul class="dropdown-menu">
               <li><a href="news.aspx">News</a></li>
               <li class="divider"></li>
               <li><a href="sponsors.aspx">Sponsors</a></li>
               <li><a href="exhibitors.aspx">Exhibitors</a></li>
               <li><a href="testimonials.aspx">Testimonials</a></li>
               <li class="divider"></li>
               <li><a href="photos.aspx">Photos</a></li>
            </ul>
         </li>
      </ul>
   </div>
</nav>
```

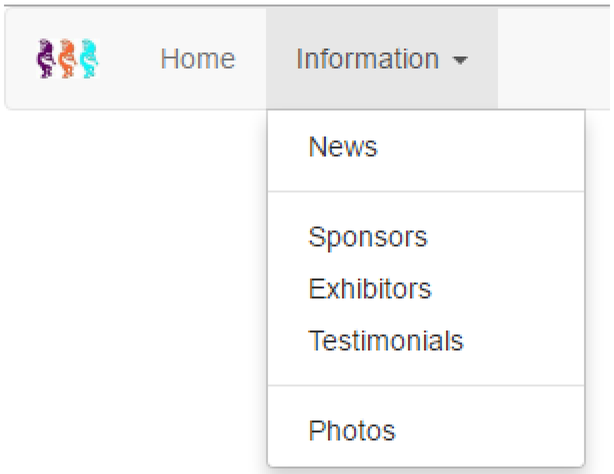


Figure 32. A dropdown menu bar.

On a small device, you typically want a “hamburger” rather than a full menu. To do that, use HTML like this in the navbar header:

```
<div class="navbar-header">
   <button type="button" class="navbar-toggle" data-toggle="collapse"
      data-target=".navbar-collapse">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
   </button>
</div>
```

Surround the <ul class="nav navbar-nav"> element with a <div class="navbar-collapse collapse"> to provide the collapsing behavior. **Figure 33** shows how the menu appears in a small viewport and **Figure 34** shows what it looks like when you click the “hamburger.”

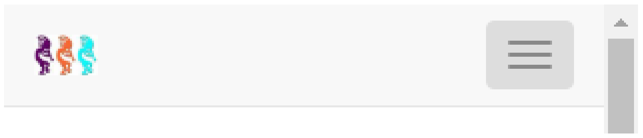


Figure 33. A collapsed "hamburger" menu.

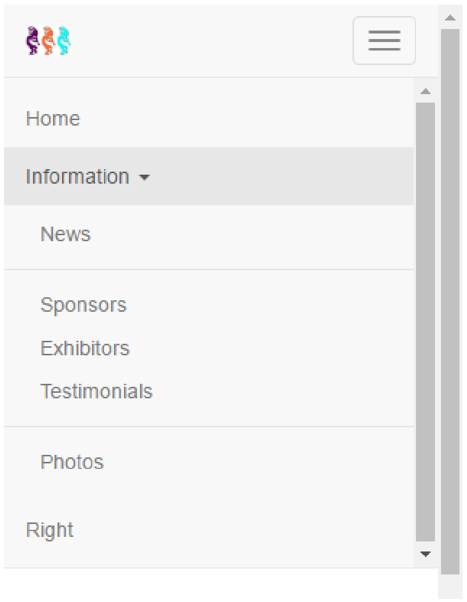


Figure 34. An expanded "hamburger" menu.

List groups

A list group provides the items in an unordered list with boxes. This HTML produces the basic list group shown in **Figure 35**:

```
<ul class="list-group">
   <li class="list-group-item">First item</li>
   <li class="list-group-item">Second item</li>
   <li class="list-group-item">Third item</li>
</ul>
```

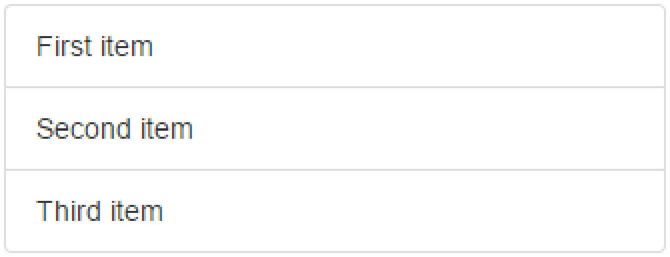


Figure 35. A basic list group.

A list group can include badges, which are numeric indicators. Badges automatically appear at the right (**Figure 36**).

```
<ul class="list-group">
   <li class="list-group-item"><span class="badge">5</span>First item</li>
   <li class="list-group-item"><span class="badge">21</span>Second item</li>
   <li class="list-group-item"><span class="badge">11</span>Third item</li>
</ul>
```

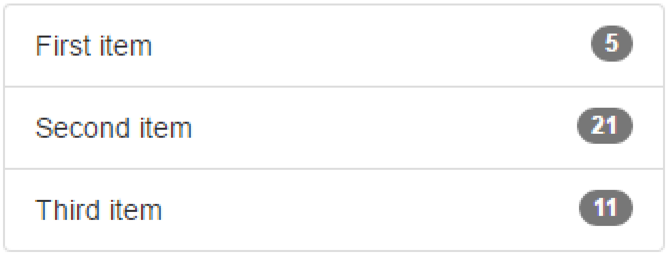


Figure 36. A list group with badges.

You can also create a list group using a <div> containing anchors instead of an unordered list. In that case, a grey background appears on hover.

```
<div class="list-group">
   <a href="#" class="list-group-item">First item</a>
   <a href="#" class="list-group-item">Second item</a>
   <a href="#" class="list-group-item">Third item</a>
</div>
```

Use the “active” class to highlight an item and “disabled” to disable an item (**Figure 37**):

```
<div class="list-group">
   <a href="#" class="list-group-item active">First item</a>
   <a href="#" class="list-group-item">Second item</a>
   <a href="#" class="list-group-item disabled">Third item</a>
</div>
```

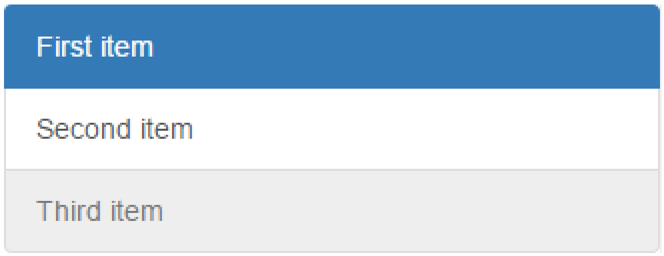
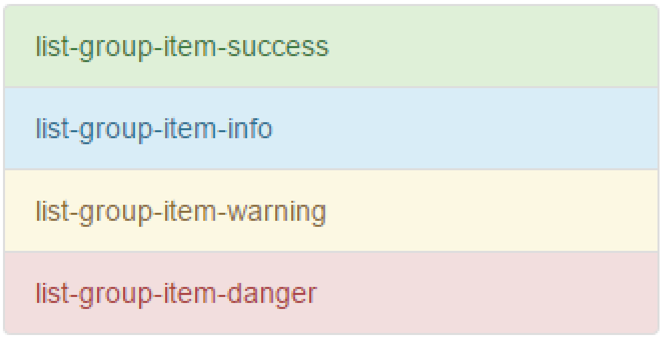


Figure 37. This list group has active and disabled items.

You can use contextual colors (**Figure 38**):

```
<ul class="list-group">
   <li class="list-group-item list-group-item-success">list-group-item-success</li>
   <li class="list-group-item list-group-item-info">list-group-item-info</li>
   <li class="list-group-item list-group-item-warning">list-group-item-warning</li>
   <li class="list-group-item list-group-item-danger">list-group-item-danger</li>
</ul>
```


list-group-item-success

list-group-item-info

list-group-item-warning

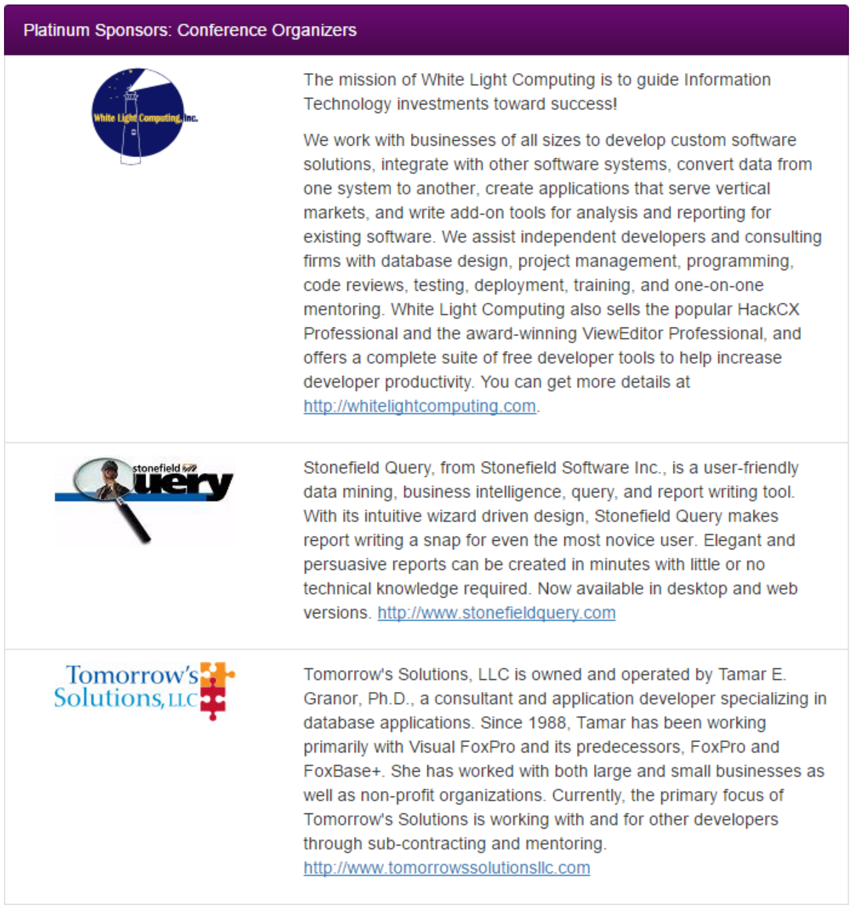
list-group-item-danger

Figure 38. A list group with contextual colors.

A list group can be more complex than just a simple list. For example, the Sponsors page of the Southwest Fox/Southwest Xbase++ web site (http://www.swfox.net/sponsors.aspx; **Figure 39**) displays sponsor information using a list group, with "active" used for the sponsor type headings (**Listing 14**).

Listing 14. This HTML, taken from www.swfox.net/sponsors.aspx, uses a complex list group.

```
<ul class="list-group">
   <li class="list-group-item active">Platinum Sponsors: Conference Organizers</li>
   <li class="list-group-item">
      <div class="row">
         <div class="col-md-4">
            <a target="_blank" href="http://www.whitelightcomputing.com">
               <img src="images/whitelightcomputing.png" width="88" height="81"
                  class="center-block"/>
            </a>
         </div>
         <div class="col-md-8">
            <p>The mission of White Light Computing ...
         </div>
      </div>
   </li>
   ... more sponsors ...
</ul>
```


Platinum Sponsors: Conference Organizers

The mission of White Light Computing is to guide Information Technology investments toward success!

We work with businesses of all sizes to develop custom software solutions, integrate with other software systems, convert data from one system to another, create applications that serve vertical markets, and write add-on tools for analysis and reporting for existing software. We assist independent developers and consulting firms with database design, project management, programming, code reviews, testing, deployment, training, and one-on-one mentoring. White Light Computing also sells the popular HackCX Professional and the award-winning ViewEditor Professional, and offers a complete suite of free developer tools to help increase developer productivity. You can get more details at http://whitelightcomputing.com.

Stonefield Query, from Stonefield Software Inc., is a user-friendly data mining, business intelligence, query, and report writing tool. With its intuitive wizard driven design, Stonefield Query makes report writing a snap for even the most novice user. Elegant and persuasive reports can be created in minutes with little or no technical knowledge required. Now available in desktop and web versions. http://www.stonefieldquery.com

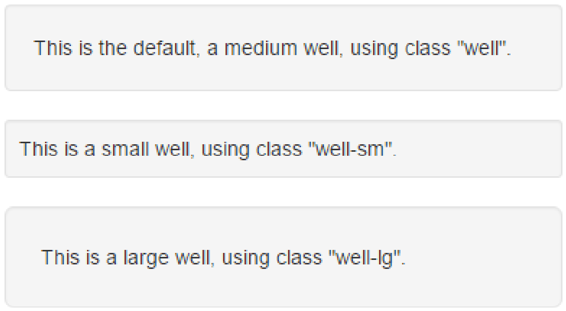
Tomorrow's Solutions, LLC is owned and operated by Tamar E. Granor, Ph.D., a consultant and application developer specializing in database applications. Since 1988, Tamar has been working primarily with Visual FoxPro and its predecessors, FoxPro and FoxBase+. She has worked with both large and small businesses as well as non-profit organizations. Currently, the primary focus of Tomorrow's Solutions is working with and for other developers through sub-contracting and mentoring. http://www.tomorrowssolutionsllc.com

Figure 39. www.swfox.net/sponsors.aspx displays sponsors in a list group.

Wells and panels

Wells and panels are bordered boxes with content. The main difference is that a panel can optionally have a header and footer.

Wells come in small (<div class="well well-sm">), medium (<div class="well">), and large (<div class="well well-lg">). **Figure 40** shows examples of these.


This is the default, a medium well, using class "well".

This is a small well, using class "well-sm".

This is a large well, using class "well-lg".

Figure 40. Wells are rounded boxes with content.

The HTML in **Listing 15** creates several panels and results in the page shown in **Figure 41**.

Listing 15. This HTML creates panels with and without headers and footers.

```
<h1>Basic</h1>
<div class="container-fluid">
   <div class="panel panel-default">
      <div class="panel-body">
         Basic panel example
      </div>
   </div>
</div>

<h1>With Header</h1>
<div class="container-fluid">
   <div class="panel panel-default">
      <div class="panel-heading">Panel heading</div>
      <div class="panel-body">
         Panel content
      </div>
   </div>
</div>

<h1>With Footer</h1>
<div class="container-fluid">
   <div class="container-fluid">
      <div class="panel panel-default">
         <div class="panel-body">
            Panel content
         </div>
      <div class="panel-footer">Panel footer</div>
   </div>
</div>
```

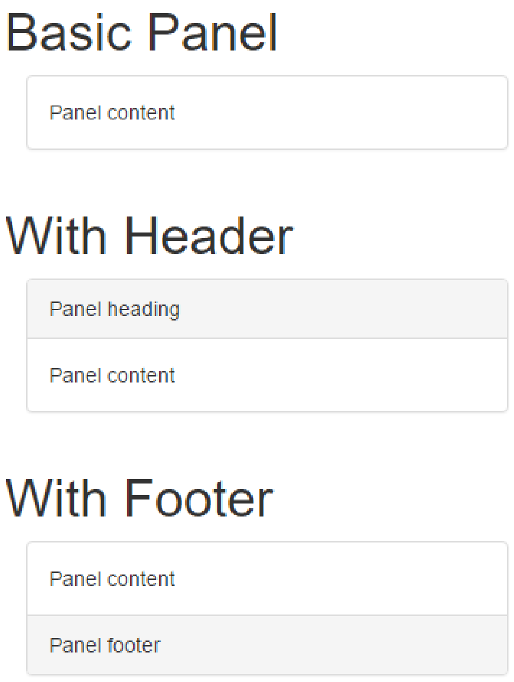
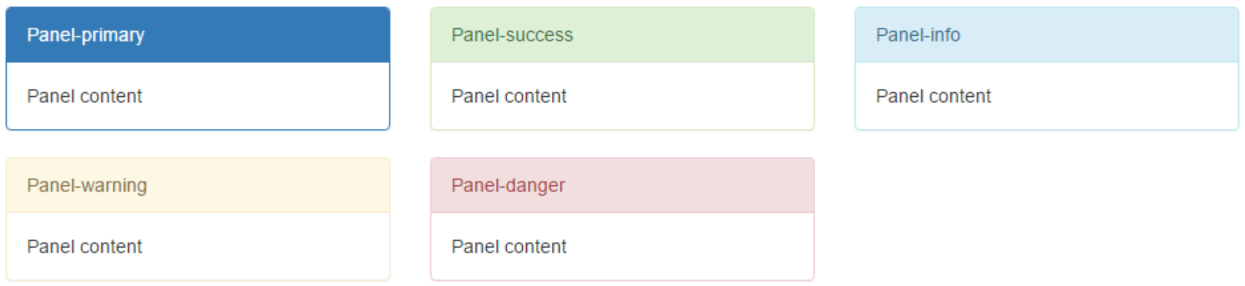


Figure 41. Panels are bordered boxes with content.

Instead of using panel-default, you can use one of the contextual classes to get the results shown in **Figure 42**.



Panel-primary

Panel content

Panel-success

Panel content

Panel-info

Panel content

Panel-warning

Panel content

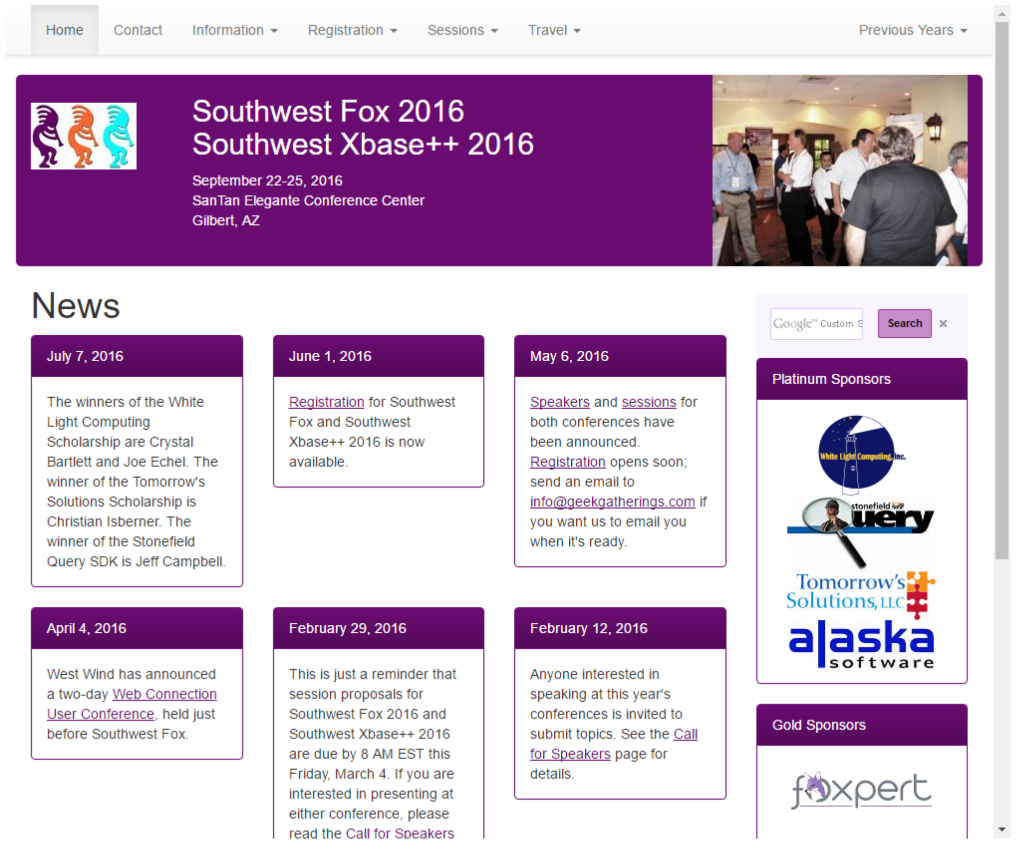
Panel-danger

Panel content

Figure 42. Panels can use contextual colors.

You can group several panels together by wrapping them in a <div class="panel-group">.

The Southwest Fox/Southwest Xbase++ uses panels for news items, sponsor information, and testimonials (**Figure 43**).



Southwest Fox 2016
Southwest Xbase++ 2016

September 22-25, 2016
SanTan Elegante Conference Center
Gilbert, AZ

News

July 7, 2016

The winners of the White Light Computing Scholarship are Crystal Bartlett and Joe Echel. The winner of the Tomorrow's Solutions Scholarship is Christian Isberner. The winner of the Stonefield Query SDK is Jeff Campbell.

June 1, 2016

Registration for Southwest Fox and Southwest Xbase++ 2016 is now available.

May 6, 2016

Speakers and sessions for both conferences have been announced. Registration opens soon; send an email to info@geekgatherings.com if you want us to email you when it's ready.

April 4, 2016

West Wind has announced a two-day Web Connection User Conference, held just before Southwest Fox.

February 29, 2016

This is just a reminder that session proposals for Southwest Fox 2016 and Southwest Xbase++ 2016 are due by 8 AM EST this Friday, March 4. If you are interested in presenting at either conference, please read the Call for Speakers

February 12, 2016

Anyone interested in speaking at this year's conferences is invited to submit topics. See the Call for Speakers page for details.

Figure 43. The Southwest Fox/Southwest Xbase++ site uses panels in several places.

Bootstrap 4, which was in alpha at the time this document was written, replaces wells and panels with a new component called cards. Cards are similar to panels but use "card-*" classes. See http://tinyurl.com/hjbyavq for details.

Accordions

Collapsing behavior allows you to hide and show content, typically when there's a lot of it. A commonly-seen example of collapsing behavior is in accordions, which is the example we'll look at. However, pretty much anything in Bootstrap can have collapsing behavior by adding "data-toggle="collapse"" to an element.

An accordion is a section of content in which only headings are initially displayed. Clicking a heading causes it to expand to show the content for the heading. Clicking another heading collapses the previously expanded item and expands the selected one. The session pages of www.swfox.net are examples of an accordion.

Listing 16 contains a stripped-down version of the Southwest Fox session page. Here are the important things to note:

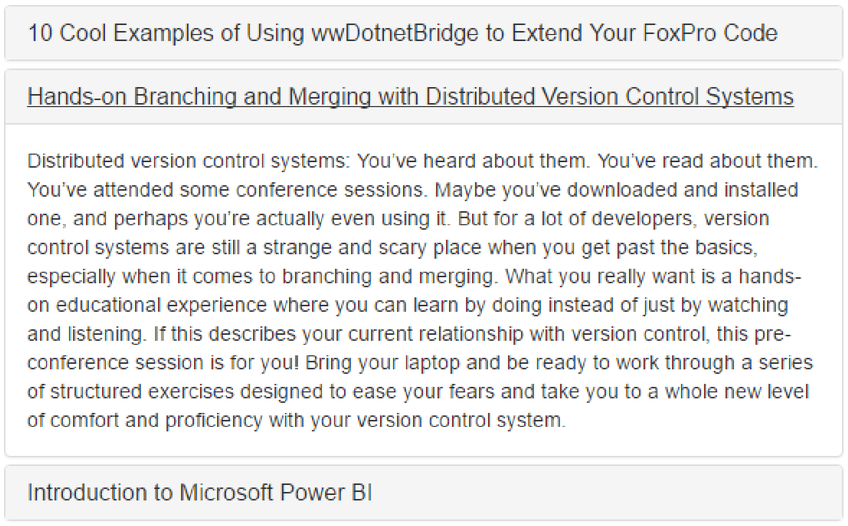
- The content is displayed in panels with headers. The header shows the title of the session and the body shows the abstract.
- The panels are contained within a panel group with an ID of "accordion" (the actual name isn't important as long as it's used consistently). This is necessary to get the behavior of the expanded panel collapsing when another one is expanded.
- The anchor tag in the panel heading uses "data-toggle="collapse"" to specify that clicking it toggles the collapse behavior, "data-parent="#accordion"" (where "accordion" is the ID of the panel group) to indicate the panel group affected by the behavior, and "href="ID"" (where "ID" is the ID of a "panel-collapse" panel containing the body to show when the panel is expanded).
- The panel body is contained within "panel-collapse collapse" panel.

Listing 16. This HTML creates a three-panel accordion.

```
<div class="panel-group" id="accordion">
   <div class="panel panel-default">
      <div class="panel-heading">
         <h3 class="panel-title">
            <a href="#10_Cool_Examples_of_Using_wwDotnetBridge"
               data-toggle="collapse" data-parent="#accordion">
               10 Cool Examples of Using wwDotnetBridge to Extend Your FoxPro Code
            </a>
         </h3>
      </div>
      <div id="10_Cool_Examples_of_Using_wwDotnetBridge"
         class="panel-collapse collapse">
         <div class="panel-body">
            wwDotnetBridge lets you call just about any .NET code directly ...
         </div>
      </div>
   </div>

   <div class="panel panel-default">
      <div class="panel-heading">
```

```
         <h3 class="panel-title">
            <a href="#Hands-on_Branching_and_Merging_with_DVCS_1"
               data-toggle="collapse" data-parent="#accordion">
               Hands-on Branching and Merging with …
            </a>
         </h3>
      </div>
      <div id="Hands-on_Branching_and_Merging_with_DVCS_1"
         class="panel-collapse collapse">
         <div class="panel-body">
            Distributed version control systems: You've heard about them. ...
         </div>
      </div>
   </div>

   <div class="panel panel-default">
      <div class="panel-heading">
         <h3 class="panel-title">
            <a href="#Introduction_to_Microsoft_Power_BI"
               data-toggle="collapse" data-parent="#accordion">
               Introduction to Microsoft Power BI
            </a>
         </h3>
      </div>
      <div id="Introduction_to_Microsoft_Power_BI" class="panel-collapse collapse">
         <div class="panel-body">
            Power BI (Business Intelligence) is Microsoft's newest ...
         </div>
      </div>
   </div>
</div>
```

Figure 44 shows the results, with the second panel expanded.



10 Cool Examples of Using wwDotnetBridge to Extend Your FoxPro Code

Hands-on Branching and Merging with Distributed Version Control Systems

Distributed version control systems: You've heard about them. You've read about them. You've attended some conference sessions. Maybe you've downloaded and installed one, and perhaps you're actually even using it. But for a lot of developers, version control systems are still a strange and scary place when you get past the basics, especially when it comes to branching and merging. What you really want is a hands-on educational experience where you can learn by doing instead of just by watching and listening. If this describes your current relationship with version control, this pre-conference session is for you! Bring your laptop and be ready to work through a series of structured exercises designed to ease your fears and take you to a whole new level of comfort and proficiency with your version control system.

Introduction to Microsoft Power BI

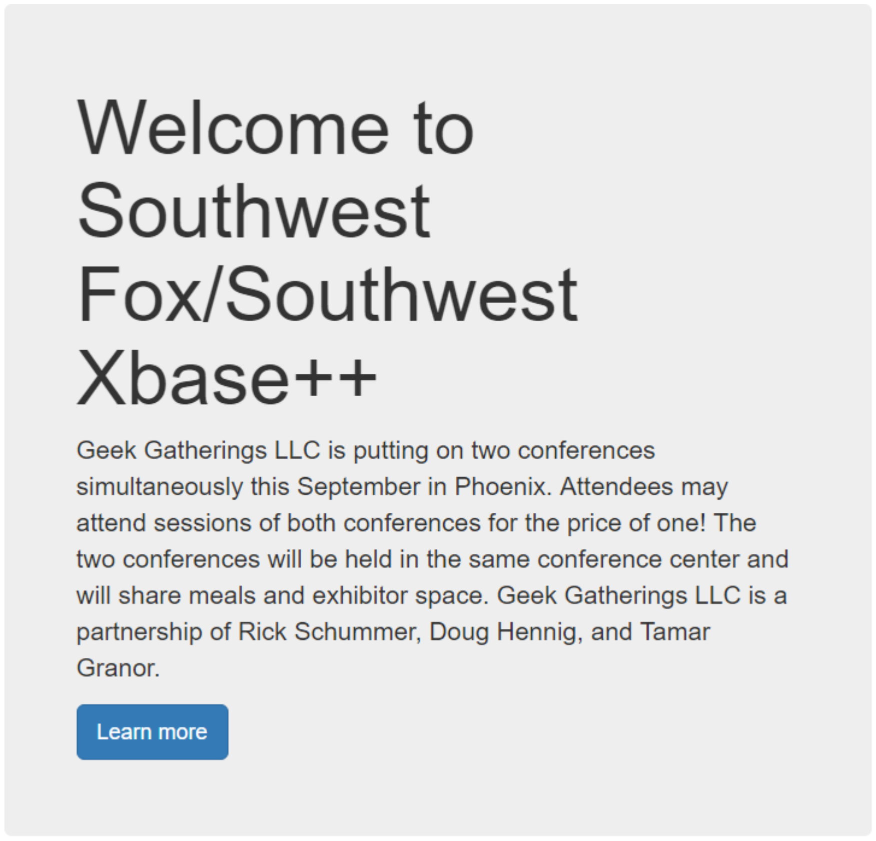
Figure 44. An accordion is an example of collapsing behavior.

Jumbotron

A jumbotron is a large box with rounded corners and large text used to draw attention to special content. It's simply a div using the "jumbotron" class. You can put anything you want into a jumbotron. The HTML in **Listing 17** creates the page shown in **Figure 45**.

Listing 17. This HTML creates a jumbotron.

```
<div class="container">
   <div class="jumbotron">
      <h1>Welcome to Southwest Fox/Southwest Xbase++</h1>
      <p>Geek Gatherings LLC is putting on two conferences …</p>
      <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a></p>
   </div>
</div>
```


Welcome to Southwest Fox/Southwest Xbase++

Geek Gatherings LLC is putting on two conferences simultaneously this September in Phoenix. Attendees may attend sessions of both conferences for the price of one! The two conferences will be held in the same conference center and will share meals and exhibitor space. Geek Gatherings LLC is a partnership of Rick Schummer, Doug Hennig, and Tamar Granor.

Figure 45. A jumbotron draws attention to special content.

Other components

Bootstrap has numerous other components as well: breadcrumbs, pagination, page header, thumbnails, alerts, progress bars, media objects, and so on. See http://getbootstrap.com/components and http://getbootstrap.com/javascript for information on what's available.

Pulling it all together

Let's take a look at an example of a common multi-panel layout for a web site. The HTML is shown in **Listing 18** and the result in **Figure 46**. I color-coded the HTML to make it easy to identify the sections.

The first section provides a pull-down menu for the site. The href attribute for the links is set to "#" in this example but in a real site they would likely be a page to navigate to.

The next section is the header. I used a jumbotron containing an image, some text, and a button. The top padding is necessary to leave room for the menu bar above the header.

Next there's a container for the middle section. It has a row for the columns that make up the middle section.

The next section is a sidebar. This is often used for things like testimonials, product descriptions, and so on. I used "bg-info" to provide a background color.

The main content section is where the majority of the content for the page goes.

The panel at the right includes a set of images, which could be ads, sponsor logos, or something similar. I used "img-rounded img-responsive" to make the images look attractive.

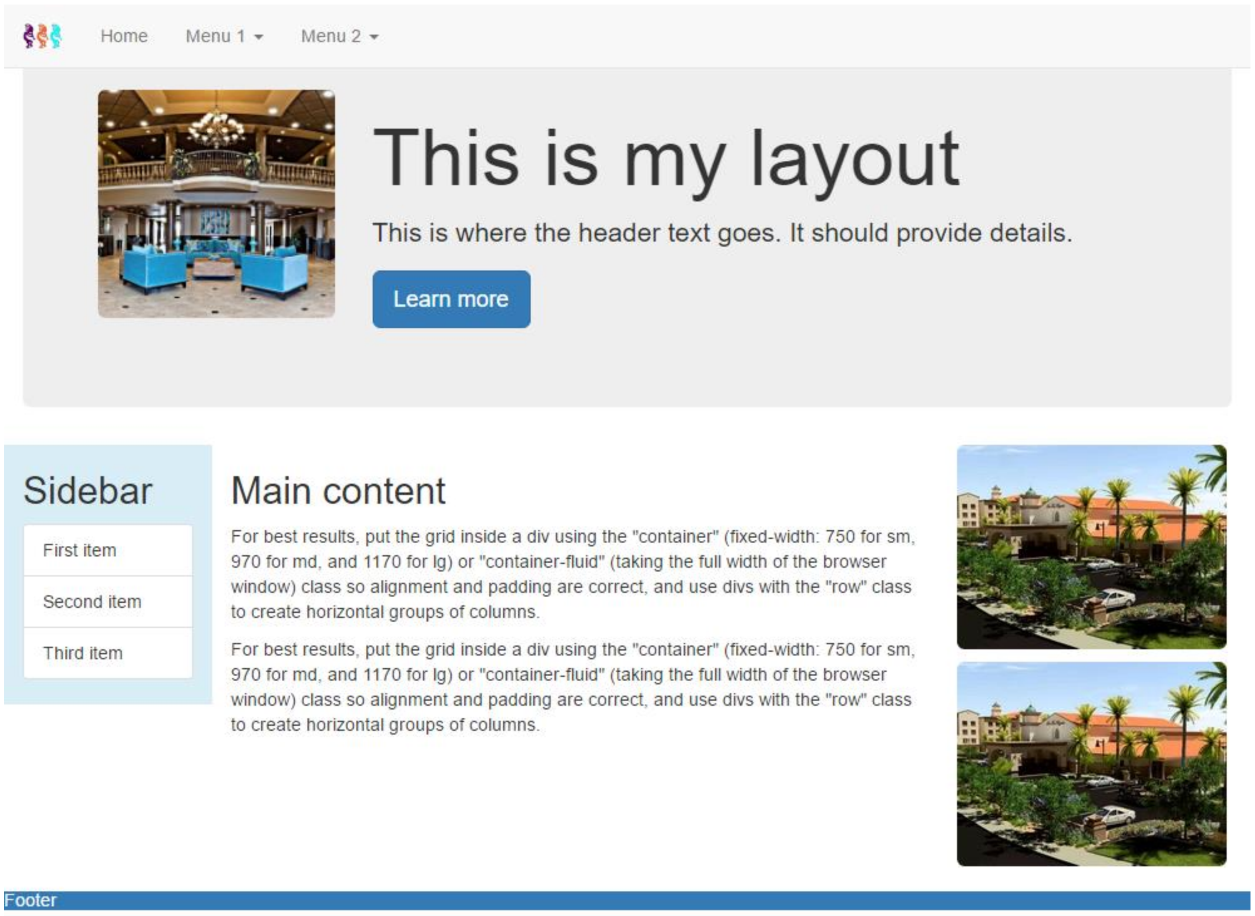
The next section in the HTML closes the divs for the middle section.

The final section is a footer for the page. This often contains things like a copyright notice or a set of links to common pages (privacy policy, contact us, etc.).

Listing 18. This HTML provides a multi-panel layout for a web site.

```
<nav class="navbar navbar-default navbar-fixed-top">
   <div class="container-fluid">
      <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse"
            data-target=".navbar-collapse">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
         </button>
         <img class="navbar-brand" src="KokoWhite.jpg">
      </div>
      <div class="navbar-collapse collapse">
         <ul class="nav navbar-nav">
            <li><a href="#">Home</a></li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Menu 1 <span class="caret"></span></a>
               <ul class="dropdown-menu">
                  <li><a href="#">Item 1</a></li>
                  <li><a href="#">Item 2</a></li>
                  <li><a href="#">Item 3</a></li>
               </ul>
            </li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Menu 2 <span class="caret"></span></a>
               <ul class="dropdown-menu">
                  <li><a href="#">Item 1</a></li>
                  <li><a href="#">Item 2</a></li>
                  <li><a href="#">Item 3</a></li>
               </ul>
            </li>
         </ul>
      </div>
   </div>
</nav>

<div class="container-fluid" style="padding-top:60px">
   <div class="jumbotron">
      <div class="row">
         <div class="col-lg-3">
            <img src="legado_hotel_lobby.jpg" class="img-rounded img-responsive"
               alt="Hotel lobby" width="400" height="386">
         </div>
         <div class="col-lg-9">
            <h1>This is my layout</h1>
            <p>This is where the header text goes. It should provide details.</p>
            <p><a class="btn btn-primary btn-lg" href="#" role="button">
               Learn more</a></p>
         </div>
      </div>
   </div>
</div>

<div class="container-fluid" id="body">
   <div class="row">
      <div class="col-md-2 bg-info">
         <h2>Sidebar</h2>
         <ul class="list-group">
            <li class="list-group-item">First item</li>
            <li class="list-group-item">Second item</li>
            <li class="list-group-item">Third item</li>
         </ul>
      </div>

      <div class="col-md-7">
         <h2>Main content</h2>
         <p>For best results...</p>
      </div>

      <div class="col-md-3">
         <p><img src="conference.jpg" class="img-rounded img-responsive"
            alt="Conference center" width="215" height="163"></p>
         <p><img src="conference.jpg" class="img-rounded img-responsive"
```

```
                alt="Conference center" width="215" height="163"></p>
        </div>
    </div>
</div>

<div class="bg-primary" id="Footer">
    <h5>Footer</h5>
</div>
```



This is my layout

This is where the header text goes. It should provide details.

Sidebar

First item

Second item

Third item

Main content

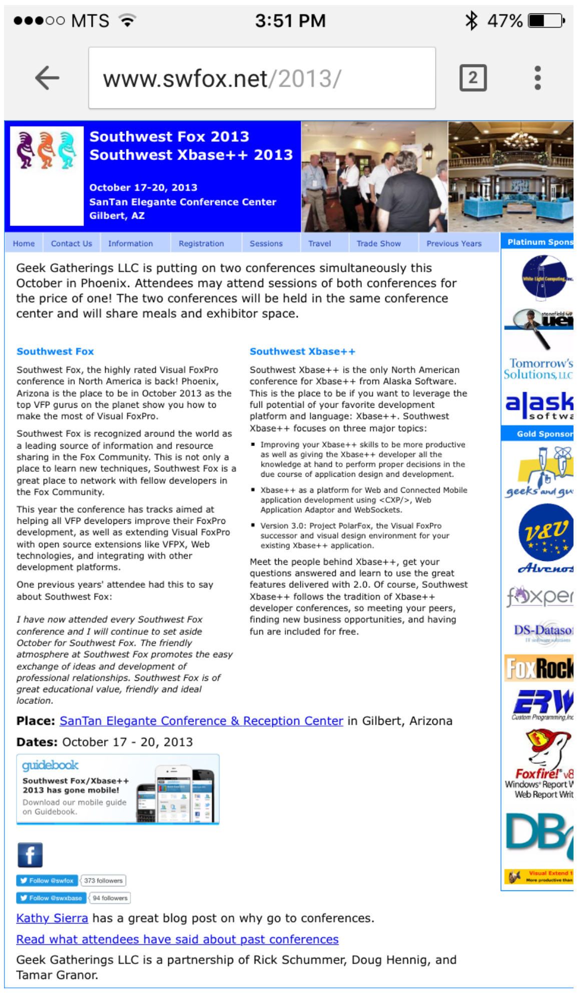
For best results, put the grid inside a div using the "container" (fixed-width: 750 for sm, 970 for md, and 1170 for lg) or "container-fluid" (taking the full width of the browser window) class so alignment and padding are correct, and use divs with the "row" class to create horizontal groups of columns.

For best results, put the grid inside a div using the "container" (fixed-width: 750 for sm, 970 for md, and 1170 for lg) or "container-fluid" (taking the full width of the browser window) class so alignment and padding are correct, and use divs with the "row" class to create horizontal groups of columns.

Figure 46. A multi-panel layout.

Web site makeover

Prior to 2014, the Southwest Fox/Southwest Xbase++ conference web site (http://www.swfox.net) used tables and custom CSS to get the desired appearance. For example, the home page (**Figure 47** is a screen shot taken from my iPhone) has two columns discussing the two conferences. Each column is one cell in a two-column, one-row table.



Southwest Fox 2013
Southwest Xbase++ 2013

October 17-20, 2013
SanTan Elegante Conference Center
Gilbert, AZ

Geek Gatherings LLC is putting on two conferences simultaneously this October in Phoenix. Attendees may attend sessions of both conferences for the price of one! The two conferences will be held in the same conference center and will share meals and exhibitor space.

Southwest Fox

Southwest Fox, the highly rated Visual FoxPro conference in North America is back! Phoenix, Arizona is the place to be in October 2013 as the top VFP gurus on the planet show you how to make the most of Visual FoxPro.

Southwest Fox is recognized around the world as a leading source of information and resource sharing in the Fox Community. This is not only a place to learn new techniques, Southwest Fox is a great place to network with fellow developers in the Fox Community.

This year the conference has tracks aimed at helping all VFP developers improve their FoxPro development, as well as extending Visual FoxPro with open source extensions like VFPX, Web technologies, and integrating with other development platforms.

One previous years' attendee had this to say about Southwest Fox:

I have now attended every Southwest Fox conference and I will continue to set aside October for Southwest Fox. The friendly atmosphere at Southwest Fox promotes the easy exchange of ideas and development of professional relationships. Southwest Fox is of great educational value, friendly and ideal location.

Southwest Xbase++

Southwest Xbase++ is the only North American conference for Xbase++ from Alaska Software. This is the place to be if you want to leverage the full potential of your favorite development platform and language: Xbase++. Southwest Xbase++ focuses on three major topics:

- Improving your Xbase++ skills to be more productive as well as giving the Xbase++ developer all the knowledge at hand to perform proper decisions in the due course of application design and development.
- Xbase++ as a platform for Web and Connected Mobile application development using <CXP/>, Web Application Adaptor and WebSockets.
- Version 3.0: Project PolarFox, the Visual FoxPro successor and visual design environment for your existing Xbase++ application.

Meet the people behind Xbase++, get your questions answered and learn to use the great features delivered with 2.0. Of course, Southwest Xbase++ follows the tradition of Xbase++ developer conferences, so meeting your peers, finding new business opportunities, and having fun are included for free.

Place: SanTan Elegante Conference & Reception Center in Gilbert, Arizona

Dates: October 17 - 20, 2013

guidebook
Southwest Fox/Xbase++ 2013 has gone mobile!
Download our mobile guide on Guidebook.

Kathy Sierra has a great blog post on why go to conferences.

Read what attendees have said about past conferences

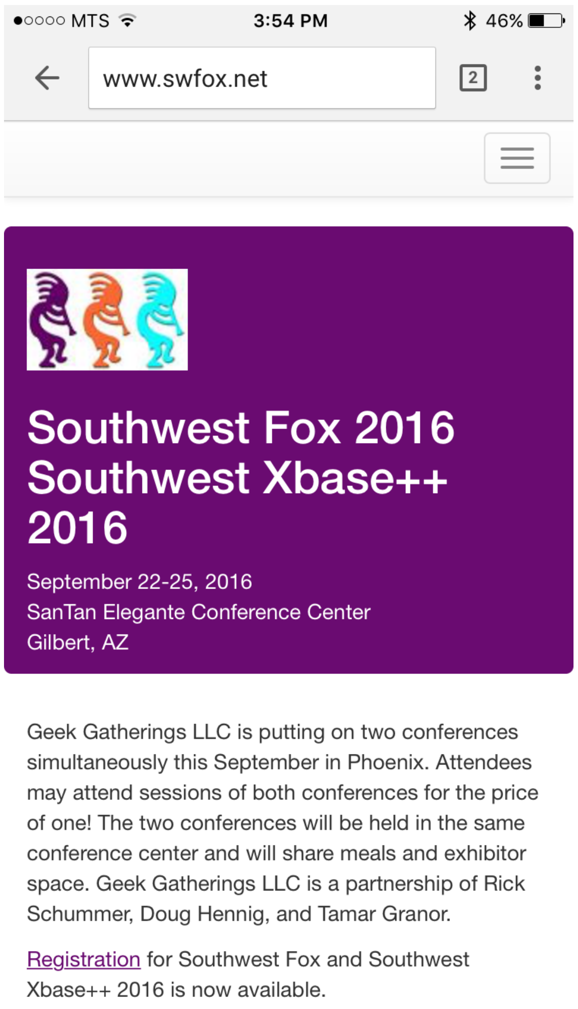
Geek Gatherings LLC is a partnership of Rick Schummer, Doug Hennig, and Tamar Granor.

Figure 47. swfox.net: the "before."

The two big issues with this web site are:

- It was a lot of work to lay out each page as desired: there's lots of custom CSS in swfox.css and lots of styles in the HTML that had to be hand-tweaked over and over until it worked correctly.
- It's not mobile-friendly. The text is too small to read without pinching, zooming, and horizontal scrolling, and the page is cut off at the right side.

Compare that to **Figure 48**, also taken from my iPhone, which shows the current site (the makeover was actually done in 2014, so the sites in that year and later look similar to the current one).



Southwest Fox 2016
Southwest Xbase++ 2016

September 22-25, 2016
SanTan Elegante Conference Center
Gilbert, AZ

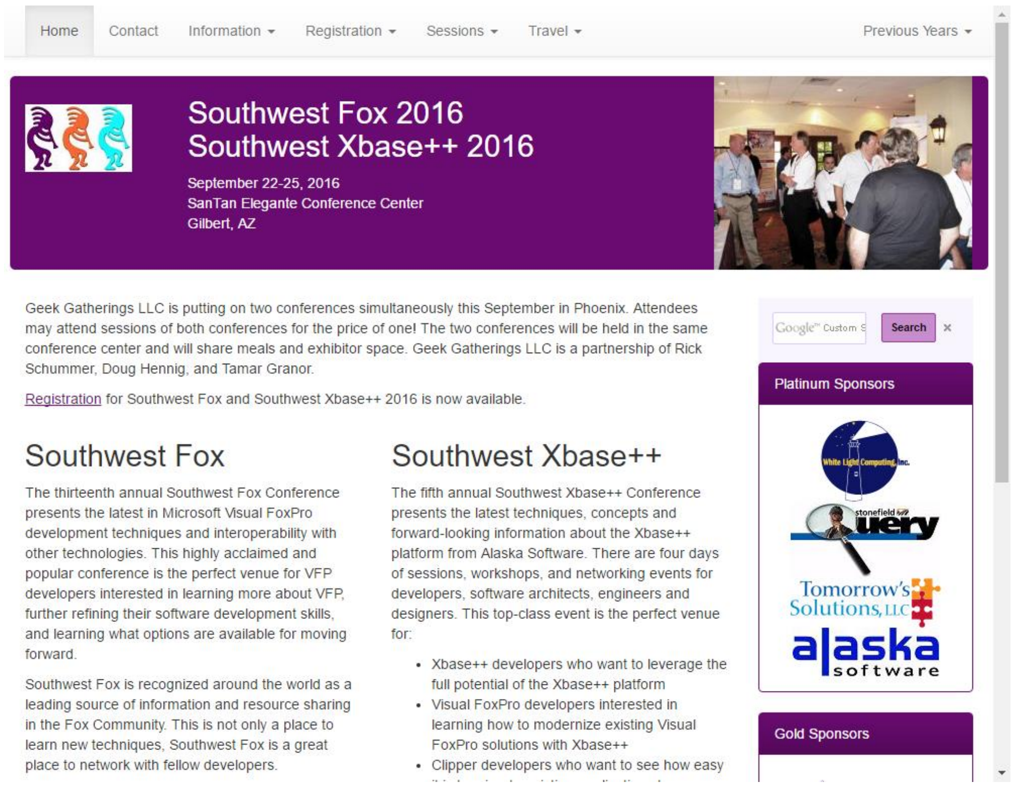
Geek Gatherings LLC is putting on two conferences simultaneously this September in Phoenix. Attendees may attend sessions of both conferences for the price of one! The two conferences will be held in the same conference center and will share meals and exhibitor space. Geek Gatherings LLC is a partnership of Rick Schummer, Doug Hennig, and Tamar Granor.

Registration for Southwest Fox and Southwest Xbase++ 2016 is now available.

Figure 48. swfox.net: the "after."

The updated site has several advantages:

- The text is large and readable without zooming on mobile devices.
- The menu appears as a hamburger menu on mobile devices and a menu bar on larger devices.
- The layout is responsive: the text appears in multiple columns on larger devices (**Figure 49**) and a single column on smaller ones.
- There's very little custom CSS and only a couple of tables in the whole site (and those are only there for easy conversion to Microsoft Word for creation of the conference guide), so maintenance is easy.



Southwest Fox 2016
Southwest Xbase++ 2016

September 22-25, 2016
SanTan Elegante Conference Center
Gilbert, AZ

Geek Gatherings LLC is putting on two conferences simultaneously this September in Phoenix. Attendees may attend sessions of both conferences for the price of one! The two conferences will be held in the same conference center and will share meals and exhibitor space. Geek Gatherings LLC is a partnership of Rick Schummer, Doug Hennig, and Tamar Granor.

Registration for Southwest Fox and Southwest Xbase++ 2016 is now available.

Southwest Fox

The thirteenth annual Southwest Fox Conference presents the latest in Microsoft Visual FoxPro development techniques and interoperability with other technologies. This highly acclaimed and popular conference is the perfect venue for VFP developers interested in learning more about VFP, further refining their software development skills, and learning what options are available for moving forward.

Southwest Fox is recognized around the world as a leading source of information and resource sharing in the Fox Community. This is not only a place to learn new techniques, Southwest Fox is a great place to network with fellow developers.

Southwest Xbase++

The fifth annual Southwest Xbase++ Conference presents the latest techniques, concepts and forward-looking information about the Xbase++ platform from Alaska Software. There are four days of sessions, workshops, and networking events for developers, software architects, engineers and designers. This top-class event is the perfect venue for:

- Xbase++ developers who want to leverage the full potential of the Xbase++ platform
- Visual FoxPro developers interested in learning how to modernize existing Visual FoxPro solutions with Xbase++
- Clipper developers who want to see how easy

Figure 49. swfox.net as it appears on a desktop.

Let's see what it takes to update the 2013 site to the new one. The site uses a template file named template.master to provide the common content all pages have, so let's start there.

First, let's clean up the header information. 2013:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ Master Language="C#" %>
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-
   com:vml" xmlns:o="urn:schemas-microsoft-com:office:office">
```

New:

```
<!DOCTYPE html>
<%@ Master Language="C#" %>
<html lang="en">
```

Next the <head> section. 2013:

```
<head runat="server">
   <meta http-equiv="Content-Language" content="en-us" />
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
   <meta name="keywords" content="Visual FoxPro, Xbase, Xbase++, conference, VFP,
      programming, development, Sedna, SQL Server, FoxPro, Southwest Fox, Fox,
      Office, Microsoft Office, SQL, developers, Phoenix, Mesa,
      Schummer, Hennig, Granor, White Light Computing, Stonefield, Tomorrow's
      Solutions, keynote, session, networking, Geek Gatherings, Geek" />

   <meta name="application-name" content="Southwest Fox" />
   <meta name="msapplication-starturl" content="http://www.swfox.net" />
   <meta name="msapplication-navbutton-color" content="#3480C0" />
   <meta name="msapplication-tooltip" content="Southwest Fox" />

   <meta name="msapplication-task" content="name=Register for Southwest Fox;
      action-uri=http://www.swfox.net/register.aspx;icon-uri=/favicon.ico" />
   <meta name="msapplication-task" content="name=Speakers;
      action-uri=http://www.swfox.net/speakers.aspx;icon-uri=/favicon.ico" />
   <meta name="msapplication-task" content="name=Sessions;
      action-uri=http://www.swfox.net/sessions.aspx;icon-uri=/favicon.ico" />
   <meta name="msapplication-task" content="name=News;
      action-uri=http://www.swfox.net/news.aspx;icon-uri=/favicon.ico" />
   <meta name="msapplication-task" content="name=Blog;
      action-uri=http://swfox.net/blog/index.php;icon-uri=/favicon.ico" />

   <link rel="shortcut icon" href="favicon.ico" type="image/ico"/>
   <link rel="alternate" type="application/rss+xml" title="Southwest Fox Blog"
      href="http://swfox.net/blog/feed/" />
   <link rel="stylesheet" type="text/css" href="swfox.css" />
   <link rel="stylesheet" type="text/css" href="js/jquery-ui-1.8.10.custom.css" />
   <link rel="stylesheet" type="text/css" href="js/superfish.css" media="screen" />

   <title>Southwest Fox</title>
   <asp:ContentPlaceHolder id="head" runat="server">
   </asp:ContentPlaceHolder>
</head>
```

New: the section in blue was added for Bootstrap support, including a custom Bootstrap theme that provides the purple color instead of blue for "primary" classes, created at http://getbootstrap.com/customize. The meta tags using "msapplication" were removed as they added shortcut support that no one uses and the jQuery and Superfish CSS were removed because they aren't used anymore (the sections shown in orange in the 2013 listing).

```
<head runat="server">
   <meta http-equiv="Content-Language" content="en-us" />
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="keywords" content="Visual FoxPro, Xbase, Xbase++, conference, VFP,
      programming, development, Sedna, SQL Server, FoxPro, Southwest Fox, Fox,
      Office, Microsoft Office, SQL, developers, Phoenix, Mesa,
      Schummer, Hennig, Granor, White Light Computing, Stonefield, Tomorrow's
      Solutions, keynote, session, networking, Geek Gatherings, Geek" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta charset="utf-8">

   <!-- Bootstrap core CSS -->
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/
      bootstrap.min.css" integrity="sha384-1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDD
      djZlpLegxhjVME1fgjWPGmkzs7" crossorigin="anonymous">
```

```
    <!-- Optional theme -->
    <link rel="stylesheet" href="css/bootstrap-theme.min.css">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media
    queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
       <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js">
       </script>
       <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

    <link rel="shortcut icon" href="favicon.ico" type="image/ico" />
    <link rel="alternate" type="application/rss+xml" title="Southwest Fox Blog"
       href="http://swfox.net/blog/feed/" />
    <link rel="stylesheet" type="text/css" href="css/swfox.css" />

    <title>Southwest Fox</title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
```

The <body> section was changed a lot. First, in 2013, it uses a form (although I don't remember why) and the content is contained in a <div id="wrapper">. The banner section comes first: it displays the Kokopelli logo floated to the left, information about the conference, and a couple of photos of the conference center floated to the right. Several custom classes ("bannerleft," "bannerheading," and "bannertext") provide color and font sizes. The clear at the end is needed to start the rest of the content at the left.

```
<form id="form1" runat="server">
    <div id="wrapper">
       <div id="banner">
          <div id="bannerleft">
             <img alt="" src="images/kokotall.jpg" width="126" height="170"
                style="float:left"/>
             <p id="bannerheading">Southwest Fox 2013<br />
                Southwest Xbase++ 2013</p>
             <p class="bannertext">October 17-20, 2013<br/>
                SanTan Elegante Conference Center<br />
                Gilbert, AZ</p>
          </div>

          <img src="images/Hotel2.jpg" width="252" height="192"
             style="float:right"/>
          <img src="images/image008.jpg" width="251" height="192"
             style="float:right" />
       </div>

       <p style="clear:both; margin-bottom:0"></p>
```

New: the menu comes first, but to compare apples to apples, let's look at the banner HTML first:

```
<div class="container">
   <div class="row header-row round-corners">
      <div class="col-md-2">
         <img src="images/KokoWhite.jpg" class="img-responsive top-padding" />
      </div>
      <div class="col-md-6 white-text">
         <h2>Southwest Fox 2016<br />
            Southwest Xbase++ 2016</h2>
         <p>
            September 22-25, 2016<br />
            SanTan Elegante Conference Center<br />
            Gilbert, AZ
         </p>
      </div>
      <div class="col-md-4">
         <img src="images/image008.jpg" class="img-responsive pull-right visible-md
            visible-lg" />
      </div>
   </div>
```

As you can see, there was a bit of conversion in the original HTML (the changes are shown in blue). Here are some comments about this:

- The page content is included in a Bootstrap container.
- The banner uses a one-row, three-column grid, with the first column containing the Kokopelli logo, the second column the conference information, and the last column a photo of the conference center. This is a lot simpler as no floats are needed.
- The row uses a couple of custom classes ("header-row" and "round-corners" to provide the background color and rounded corners of the banner.
- The Kokopelli image uses the "top-padding" custom class to position the image correctly using top padding.
- Notice from Figure 48 that the photo doesn't appear on mobile devices: that's due to the "visible-md" and "visible-lg" classes which make the image only visible on medium and large devices. It's also moved to the right thanks to the "pull-right" class.

Next in 2013 is the menu. It uses Superfish, a JavaScript menu system that uses unordered lists for the menu definition (we saw that superfish.css was removed in the new site because it doesn't use Superfish). For brevity, I omitted all but the first item in each menu.

```
<ul class="sf-menu">
   <li class="current"><a href="default.aspx">Home</a></li>

   <li><a href="contact.aspx">Contact Us</a></li>

   <li><a href="#">Information</a>
      <ul>
         <li><a href="news.aspx">News</a></li>
      </ul>
   </li>

   <li><a href="#">Registration</a>
      <ul>
         <li><a href="register.aspx">Register</a></li>
      </ul>
   </li>
   <li><a href="#">Sessions</a>
      <ul>
         <li><a href="speakers.aspx">Speakers</a></li>
      </ul>
   </li>

   <li><a href="#">Travel</a>
      <ul>
         <li><a href="hotel.aspx">Hotel</a></li>
      </ul>
   </li>

   <li><a href="#">Trade Show</a>
      <ul>
         <li><a href="exhibitors.aspx">Exhibitors</a></li>
      </ul>
   </li>

   <li><a href="#">Previous Years</a>
      <ul>
         <li><a target="_blank"
            href="http://www.swfox.net/2012/default.aspx">2012</a></li>
      </ul>
   </li>
</ul>
```

New: as I mentioned earlier, the menu is the first section in the body because it appears at the top rather than under the banner as it does in the 2013 site. The HTML is actually a little more complicated because of the additional <div> elements used for the navbar, but since Bootstrap navbars also use unordered lists, the conversion mostly required adding additional elements to the existing ones (the changes are shown in blue). Also three changes were made to the menu: the Trade Show menu was removed, the Previous Years menu was moved to the far right, and since navbars support dividers, they were added to make the menu easier to read. Again, for brevity, I omitted all but the first item in each menu.

```
<div class="navbar navbar-default navbar-fixed-top" role="navigation">
   <div class="container">
      <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse"
            data-target=".navbar-collapse">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
         </button>
      </div>
      <div class="navbar-collapse collapse">
         <ul class="nav navbar-nav">
            <li class="active"><a href="default.aspx">Home</a></li>
            <li><a href="contact.aspx">Contact</a></li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Information <b class="caret"></b></a>
               <ul class="dropdown-menu">
                  <li><a href="news.aspx">News</a></li>
               </ul>
            </li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Registration <b class="caret"></b></a>
               <ul class="dropdown-menu">
                  <li><a href="register.aspx">Register</a></li>
               </ul>
            </li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Sessions <b class="caret"></b></a>
                     <ul class="dropdown-menu">
                        <li><a href="speakers.aspx">Speakers</a></li>
                     </ul>
            </li>
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Travel <b class="caret"></b></a>
               <ul class="dropdown-menu">
                  <li><a href="hotel.aspx">Hotel</a></li>
               </ul>
            </li>
         </ul>
         <ul class="nav navbar-nav navbar-right">
            <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                  Previous Years <b class="caret"></b></a>
               <ul class="dropdown-menu">
                  <li><a target="_blank"
                     href="http://www.swfox.net/2015/default.aspx">2015</a></li>
               </ul>
            </li>
         </ul>
      </div>
   </div>
</div>
```

Next in 2013 comes the sponsors section, which displays sponsor logos down the right edge, followed by the template placeholder (where the ASP.NET engine inserts content from pages into the template to create the final result). This uses a couple of custom classes, “sponsors” and “sponsorsheader,” to float the section to the right and provide the desired font and colors. For brevity, only the first three sponsors are shown.

```
<div id="sponsors">
   <p class="sponsorsheader">Platinum Sponsors</p>
   <p><a target="_blank" href="http://www.whitelightcomputing.com">
      <img alt="White Light Computing" src="images/whitelightcomputing.gif"
         width="88" height="81"/></a></p>
   <p><a target="_blank" href="http://www.stonefieldquery.com">
      <img alt="Stonefield Query" src="images/SFQuerySmall.jpg" width="149"
         height="76"/></a></p>
   <p><a target="_blank" href="http://www.tomorrowssolutionsllc.com">
      <img alt="Tomorrow's Solutions" src="images/TSNarrow149.jpg" width="149"
         height="49"/></a></p>
</div>

<div id="content">
   <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server"
      EnableViewState="False">
      <p>*** Specify the title for the page</p>
   </asp:ContentPlaceHolder>
</div>
```

New: rather than using floats to display the sponsors at the right, the new site uses a two-column Bootstrap row with the left column being the placeholder for content and the right column containing the sponsor information. Also, to make it look nicer, I used panels to hold the images. Again, for brevity, only the first three sponsors are shown, and I omitted the HTML providing Google search functionality since that wasn't present in the 2013 site.

```
<div class="row">
   <div class="col-md-9">
      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server"
         EnableViewState="False">
         <p>*** Specify the title for the page</p>
      </asp:ContentPlaceHolder>
   </div>
   <div class="col-md-3 top-padding">
      <div class="panel panel-primary">
         <div class="panel-heading">Platinum Sponsors</div>
         <div class="panel-body">
            <a target="_blank" href="http://www.whitelightcomputing.com">
               <img alt="White Light Computing"
                  src="images/whitelightcomputing.png" width="88" height="81"
                  class="center-block" /></a>
            <a target="_blank" href="http://www.stonefieldquery.com">
               <img alt="Stonefield Query" src="images/SFQuerySmall.jpg"
                  width="149" height="76" class="center-block" /></a>
            <a target="_blank" href="http://www.tomorrowssolutionsllc.com">
               <img alt="Tomorrow's Solutions" src="images/TSNarrow149.jpg"
                  width="149" height="49" class="center-block" /></a>
         </div>
      </div>
   </div>
</div>
```

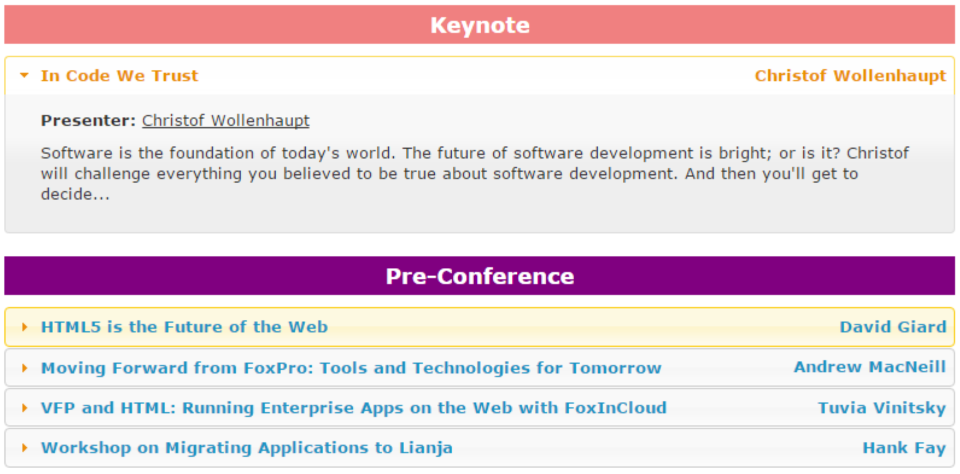
Finally there's the script section at the end. Most of the JavaScript from 2013 is no longer required because it's used to initialize Superfish, the shortcut I mentioned earlier, a rotating set of images, and a jQuery accordion for the sessions pages:

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/
   jquery/1.7.2/jquery.min.js"></script>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/
   jqueryui/1.8.18/jquery-ui.min.js"></script>
<script type="text/javascript" src="js/jquery.cycle.lite.min.js"></script>
<script type="text/javascript" src="js/hoverIntent.js"></script>
<script type="text/javascript" src="js/superfish.js"></script>

<script type="text/javascript">
    $(document).ready(function () {
      if (typeof skip != 'boolean')
         $(".accordion").accordion({ header: "h3", collapsible: true, active: false,
            autoHeight: false, navigation: true });

      $('#slide1').cycle({
         fx: 'fade', speed: 2000, timeout: 1000
      });
      $('#slide2').cycle({
         fx: 'fade', speed: 2000, timeout: 1100
      });
      $("ul.sf-menu").superfish();
   });
</script>

<script type="text/javascript" charset="utf-8">
   // This code creates a two second timeout that adds a custom overlay icon
   // and triggers a notification.
   window.setTimeout(function () {
   try {
      window.external.msSiteModeSetIconOverlay("/favicon.ico", "Pin Southwest Fox");
      window.external.msSiteModeActivate();
   }
   catch (e) { }
   }, 2000);
</script>
```

New: the script section is what we've seen for other Bootstrap pages plus it loads jQueryUI which is used for automation of scrolling on the sessions pages:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">
   </script>
<script src=https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js
   integrity="sha384-0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9a
   J7xS" crossorigin="anonymous"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js">
   </script>
```

The other page we'll look at is one of the session pages. Both 2013 (**Figure 50**) and the new site (**Figure 51**) use an accordion to collapse session descriptions so only one is displayed at a time.



Keynote

In Code We Trust — Christof Wollenhaupt

Presenter: Christof Wollenhaupt

Software is the foundation of today's world. The future of software development is bright; or is it? Christof will challenge everything you believed to be true about software development. And then you'll get to decide...

Pre-Conference

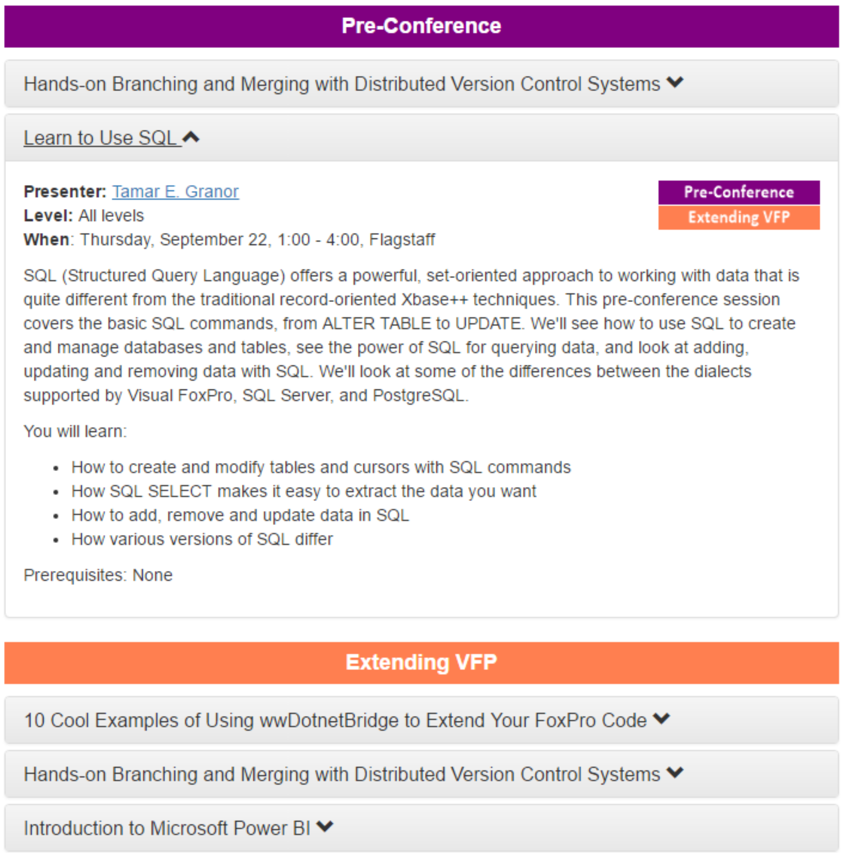
HTML5 is the Future of the Web — David Giard

Moving Forward from FoxPro: Tools and Technologies for Tomorrow — Andrew MacNeill

VFP and HTML: Running Enterprise Apps on the Web with FoxInCloud — Tuvia Vinitsky

Workshop on Migrating Applications to Lianja — Hank Fay

Figure 50. The 2013 site uses an accordion to display sessions.



Pre-Conference

Hands-on Branching and Merging with Distributed Version Control Systems

Learn to Use SQL

Presenter: Tamar E. Granor
Level: All levels
When: Thursday, September 22, 1:00 - 4:00, Flagstaff

Pre-Conference
Extending VFP

SQL (Structured Query Language) offers a powerful, set-oriented approach to working with data that is quite different from the traditional record-oriented Xbase++ techniques. This pre-conference session covers the basic SQL commands, from ALTER TABLE to UPDATE. We'll see how to use SQL to create and manage databases and tables, see the power of SQL for querying data, and look at adding, updating and removing data with SQL. We'll look at some of the differences between the dialects supported by Visual FoxPro, SQL Server, and PostgreSQL.

You will learn:

- How to create and modify tables and cursors with SQL commands
- How SQL SELECT makes it easy to extract the data you want
- How to add, remove and update data in SQL
- How various versions of SQL differ

Prerequisites: None

Extending VFP

10 Cool Examples of Using wwDotnetBridge to Extend Your FoxPro Code

Hands-on Branching and Merging with Distributed Version Control Systems

Introduction to Microsoft Power BI

Figure 51. The new site also uses an accordion.

The HTML for 2013 uses jQueryUI for the accordion. The sessions under each track are in a <div class="accordion"> and each session is in its own <div>. JavaScript at the end of the page defines <h3> as the element for the header, which displays the session title and speaker. Clicking the header expands and collapses the <div> containing the session information. Here's an example of one session:

```
<div class="accordion">
   <div>
      <h3>
         <a name="In_Code_We_Trust" href="#In_Code_We_Trust">In Code We Trust
            <span style="float:right">Christof Wollenhaupt</span></a>
      </h3>
      <div>
         <p><strong>Presenter:</strong>
            <a href="Speakers.aspx#ChristofWollenhaupt">
               Christof Wollenhaupt</a><br/>
         </p>
         <p>Software is the foundation of today's world...</p>
      </div>
   </div>
</div>
```

New: the entire sessions area of the sessions page uses a <div class="panel-group" id="accordion">. Each session consists of a panel with the heading being the title of the session and the body being the session information. For direct comparison, here's the same session from 2013 if it was on the sessions page for the new site:

```
<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">
         <a id="In_Code_We_Trust_Title"
            href="#In_Code_We_Trust" data-toggle="collapse"
            data-parent="#accordion">Keynote Presentation: In Code We Trust
         </a>
      </h3>
   </div>
   <div id="In_Code_We_Trust" class="panel-collapse collapse">
      <div class="panel-body">
            <a href="Speakers.aspx#ChristofWollenhaupt">
               Christof Wollenhaupt</a><br/>
            <br/>
         <p>Software is the foundation of today's world...</p>
      </div>
   </div>
</div>
```

Resources

GetBootstrap.com is the resource I used the most when learning Bootstrap. It has lots of documentation and examples for all aspects of Bootstrap. W3Schools (http://www.w3schools.com/bootstrap) also has lots of tutorials, including an interactive "try it yourself" where you can edit the HTML to see the effect, although it doesn't go beyond what GetBootstrap.com has. Syncfusion (http://tinyurl.com/ktw69sz) has a lot of free e-books including a couple on Bootstrap.

Summary

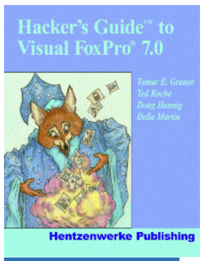
Bootstrap makes it much easier to create attractive, mobile-ready web sites than writing HTML from scratch. As you've seen, it's easy to get started with Bootstrap so hopefully you'll try it yourself the next time you need to create a web site or revamp an existing one.

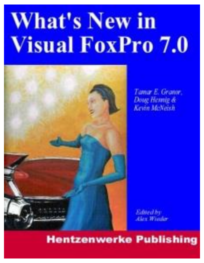
Biography

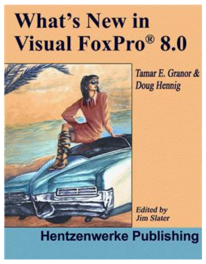
Doug Hennig is a partner with Stonefield Software Inc. He is the author of the award-winning Stonefield Database Toolkit (SDT); the award-winning Stonefield Query; the MemberData Editor, Anchor Editor, and CursorAdapter and DataEnvironment builders that come with Microsoft Visual FoxPro; and the My namespace and updated Upsizing Wizard in Sedna.

Doug is co-author of *VFPX: Open Source Treasure for the VFP Developer*, *Making Sense of Sedna and SP2*, the *What's New in Visual FoxPro* series, *Visual FoxPro Best Practices For The Next Ten Years*, and *The Hacker's Guide to Visual FoxPro 7.0*. He was the technical editor of *The Hacker's Guide to Visual FoxPro 6.0* and *The Fundamentals*. All of these books are from Hentzenwerke Publishing (http://www.hentzenwerke.com). He wrote over 100 articles in 10 years for FoxTalk and has written numerous articles in FoxPro Advisor, Advisor Guide to Visual FoxPro, and CoDe. He currently writes for FoxRockX (http://www.foxrockx.com).

Doug spoke at every Microsoft FoxPro Developers Conference (DevCon) starting in 1997 and at user groups and developer conferences all over the world. He is one of the organizers of the annual Southwest Fox and Southwest Xbase++ conferences (http://www.swfox.net). He is one of the administrators for the VFPX VFP community extensions Web site (http://vfpx.codeplex.com). He was a Microsoft Most Valuable Professional (MVP) from 1996 through 2011. Doug was awarded the 2006 FoxPro Community Lifetime Achievement Award (http://tinyurl.com/ygnk73h).






What's New in Visual FoxPro® 8.0
Tamar E. Granor & Doug Hennig
Hentzenwerke Publishing