CSS and BootStrap

We started this last time:

Creating a Simple Data-Entry Application

Setting the Scene

Imagine that a friend has decided to host a New Year's Eve party and that she has asked me to create a web app that allows her invitees to electronically RSVP.

She has asked for four key features:

- A home page that shows information about the party
- A form that can be used to RSVP
- Validation for the RSVP form, which will display a thank-you page
- RSVPs e-mailed to the party host when complete

Today, we will improve the look of the application and add validation to the form.

CSS Intro: Designing Your Webpages Using Cascading Style Sheets

Before CSS:

We used HTML code to specify how we wanted elements to look.

```
<p><font face = "Arial"  color = "blue"  size = "+1">There is some text.</font></p>
```

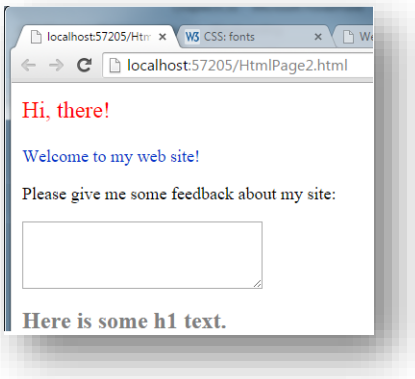
Problems with this:

- We had a limited number of formatting options.
- Presentation code was mixed with content.
- The formatting could not be changed at runtime.
- The formatting code had to be downloaded each time with the HTML code. The page took longer to load.

CSS solved these problems.

With CSS,

- We have a rich set of options.
- Presentation code and content can be separated. We can even specify all formatting in an external file.
- We can give the user access to formatting options.
- A local copy of the style sheet can be saved (cached) the first time that the page is downloaded. When other pages using the same style sheet are visited, only the content needs to be downloaded.



Types of Selectors

Begins with a dot: Class Selector
To style elements in a class

Begins with a #: ID Selector
To style an individual element

Nothing in front: Type Selector
To style elements of a certain type.

```
<style type="text/css">
    .style1 {
        color: #FF0000;
        font-size: x-large;
    }
    .style2 {
        color: #0033CC;
        font-size: large;
    }
    .style3 {
        color: #000000;
        font-size: large;
    }
    #TextArea1 {
        height: 64px;
        width: 244px;
    }
    h1 {
       font-size:x-large;
        color: #808080;
    }
</style>
```

```
<p class="style1">
    Hi, there!</p>
<p class="style2">
    Welcome to my web site!</p>
<p class="style3">
    Please give me some feedback about my site:</p>
<textarea id="TextArea1"></textarea>
<h1> Here is some h1 text.</h1>
```

* is the Universal Selector
Applies to ALL elements

```
*{
}
```

Grouping Selectors

Separated by commas – applies to all listed

```
h1, h2, h3 {
    color: blue;
}
```

Embedded CSS vs. Inline CSS

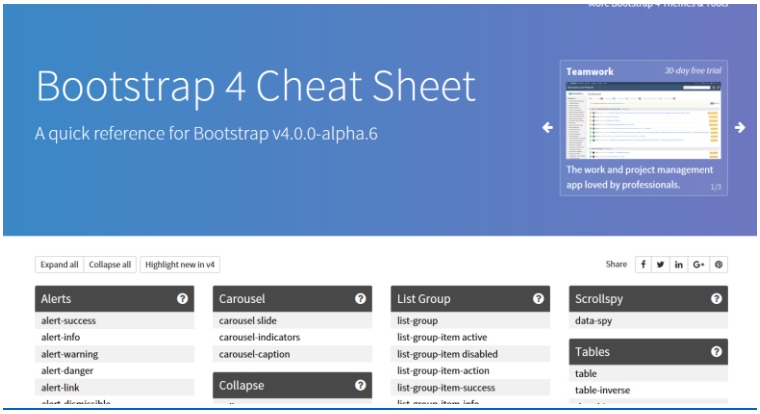
Even Better - External Style Sheets

CSS code can be kept in a file with .css extension instead of in the HTML file. It is a good idea to create a special folder to hold CSS files.

- Create a style sheet.
- Add a link to your html that will link to the CSS.
  <link href="StyleSheet.css" rel="stylesheet" type="text/css" />

Adding Bootstrap

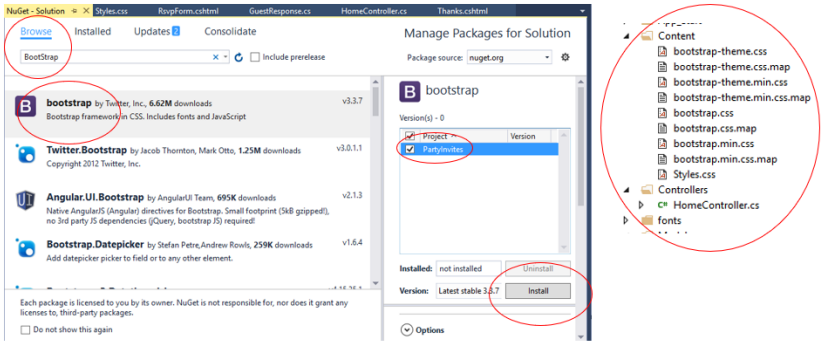
https://hackerthemes.com/bootstrap-cheatsheet/



Download PartyInvitesBeforeStyling.zip from Canvas.

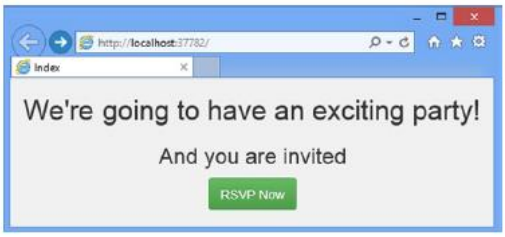
Styling the Content

Tools > NuGet Package Manager > Manage NuGet Packages for Solution.



Now add the bold lines in Index.cshtml.

```
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <title>Index</title>
    <style>
        .btn a {
            color: white;
            text-decoration: none;
        }
        body {
            background-color: #F1F1F1;
        }
    </style>
</head>
<body>
    <div class="text-center">
            <h2>
                We're going to have an exciting party.
            </h2>
                  <h3>
                              And you are invited.
            </h>
        <div class="btn btn-success">
            @Html.ActionLink("RSVP Now", "RsvpForm")
        </div>
    </div>
</body>
</html>
```



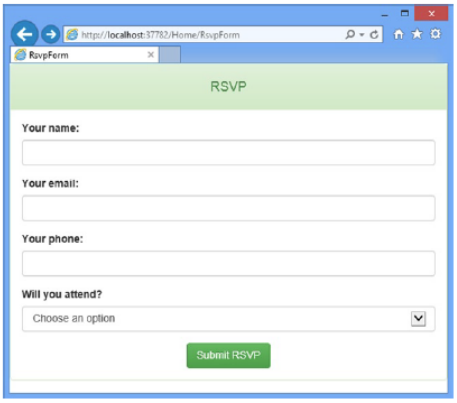
Note! This is the easiest way but it results in a button with a link inside. The link itself has to be clicked and clicking in the button but outside the link will not work.
Do this instead: Get rid of the div and send the class into the ActionLink method as a parameter.

```
@Html.ActionLink("RSVP Now", "RsvpForm", null, new {Class = "btn btn-success" })
```

Now let's change RsvpForm.cshtml to look like this:

```
@model PartyInvites.Models.GuestResponse
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>RsvpForm</title>
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
</head>
<body>
    <div class="text-center"><h4>RSVP</h4></div>
    <div class = "m-3">
        @using (Html.BeginForm())
        {
            <div class="form-group">
                <label>Your name:</label>
                @Html.TextBoxFor(x => x.Name, new { @class = "form-control" })
            </div>
            <div class="form-group">
                <label>Your email:</label>
                @Html.TextBoxFor(x => x.Email, new { @class = "form-control" })
            </div>
            <div class="form-group">
                <label>Your phone:</label>
                @Html.TextBoxFor(x => x.Phone, new { @class = "form-control" })
            </div>
            <div class="form-group">
                <label>Will you attend?</label>
                @Html.DropDownListFor(x => x.WillAttend, new[] {
                    new SelectListItem() {Text = "Yes, I'll be there",
                    Value = bool.TrueString},
                    new SelectListItem() {Text = "No, I can't come",
                    Value = bool.FalseString}
                    }, "Choose an option", new { @class = "form-control" })
            </div>
            <div class="text-center">
                <input class="btn btn-success" type="submit" value="Submit RSVP" />
            </div>
        }
    </div>
</body>

</html>
```



You can download RSVPBodyToCopy.txt and copy the text out of there for the body.

Then add the link tag for Bootstrap.

https://getbootstrap.com/docs/4.0/components/forms/