Hands-On Lab

Using Page Inspector in Visual Studio 11

Lab version: 1.2.0

Last updated: 3/9/2012

* + 1. 

Contents

[Overview 3](#_Toc319078699)

[Exercise 1: Using Page Inspector in ASP.NET MVC Projects 4](#_Toc319078700)

[Task 1 – Exploring Page Inspector 4](#_Toc319078701)

[Task 2 – Finding and Fixing Style Issues in the Photo Gallery 14](#_Toc319078702)

[Exercise 2: Using Page Inspector in WebForms Projects 19](#_Toc319078703)

[Task 1 – Exploring Page Inspector 19](#_Toc319078704)

[Task 2 – Finding and Fixing Style Issues in the Photo Gallery 29](#_Toc319078705)

[Summary 34](#_Toc319078706)

1. To give feedback please write to [VSKitFdbk@Microsoft.com](mailto:VSKitFdbk@Microsoft.com).
2. Copyright © 2012 by Microsoft Corporation. All rights reserved.

Overview

* 1. In this Hands-on Lab, you will discover a new tool to find and fix web page issues in Visual Studio - the Page Inspector.
  2. Page Inspector is a new tool that brings browser diagnostics tools to Visual Studio and provides an integrated experience among the browser, ASP.NET, and source code. It renders a web page (HTML, Web Forms, ASP.NET MVC, or Web Pages) directly within the Visual Studio IDE and lets you examine both the source code and the resulting output. Page Inspector enables you to easily decompose a website, rapidly build pages from the ground up, and quickly diagnose issues.
  3. Nowadays we have a number of Web frameworks that create flexible and scalable websites in a timely manner, such as ASP.NET MVC and WebForms. On the other hand, it gets harder to find issues on the pages because the IDE does not support the designer view in template-based pages and dynamic content. Therefore, these websites have to be opened in a browser to see how they appear to a user.
  4. Web developers use external tools to find issues that regularly run in the browser. Then, they return to the IDE and start fixing. This back and forth activity among the IDE, the browser and the profiling tools can be inefficient, and sometimes requires a fresh deployment and cache cleaning each time you want to reproduce an issue.
  5. Page Inspector bridges a gap in Web development between the client (browser tools) and the server (ASP.NET and source code) by bringing together the best of both worlds using a combined set of features.
  6. Using Page Inspector, you can see which elements in the source files (including server-side code) have produced the HTML markup to be rendered in the browser. Page Inspector also lets you modify CSS properties and DOM element attributes to see the changes reflected immediately in the browser.
  7. This hands-on lab will walk you through the Page Inspector features and show you how you can use them to fix issues in Web applications. **This lab contains two exercises using similar flows but targeting different technologies. If you are an ASP.NET MVC Developer, follow exercise one; if you are a WebForms developer follow exercise two**.

# Objectives

* 1. In this hands-on lab, you will learn how to:
  + Decompose a Web Site using Page Inspector
  + Inspect and preview CSS styles changes with Page Inspector
  + Detect and fix issues in your web pages using Page Inspector

# Prerequisites

* + [Microsoft Visual Studio 11 Ultimate Beta](http://go.microsoft.com/fwlink/?LinkId=240160)
  + Page Inspector (included in the Microsoft Visual Studio 11 Beta installation)
  + Internet Explorer 9
  + ASP.NET MVC 4 (only if you are performing exercise 1; included in the Microsoft Visual Studio 11 Beta installation)
  1. **Note:** Page Inspector hosts the current version of Internet Explorer in Visual Studio. For full functionality, you must have Internet Explorer 9 installed on your computer. You do not need to have Internet Explorer Platform Preview (also called Internet Explorer 10 Test Drive) installed.

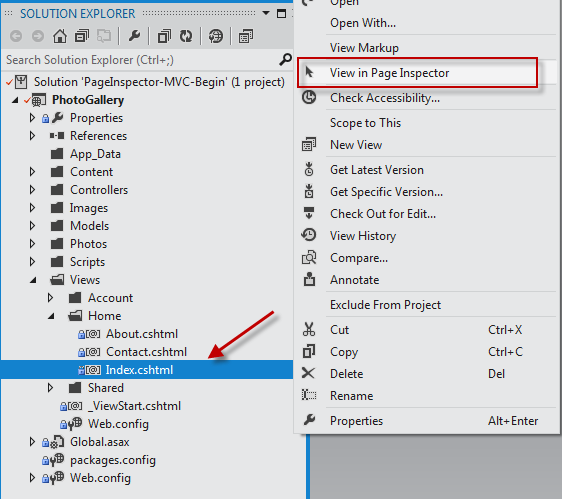
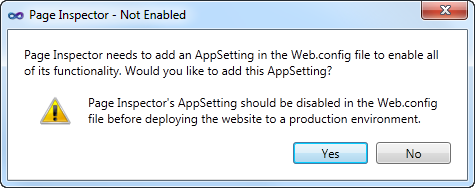
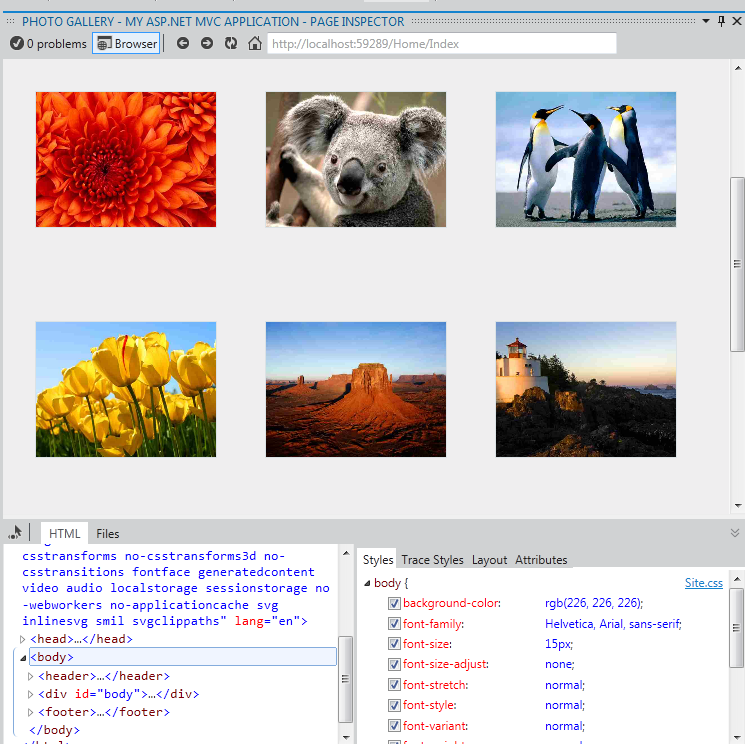
# Exercises

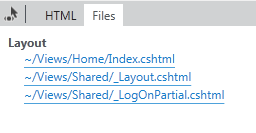
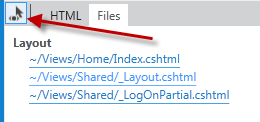
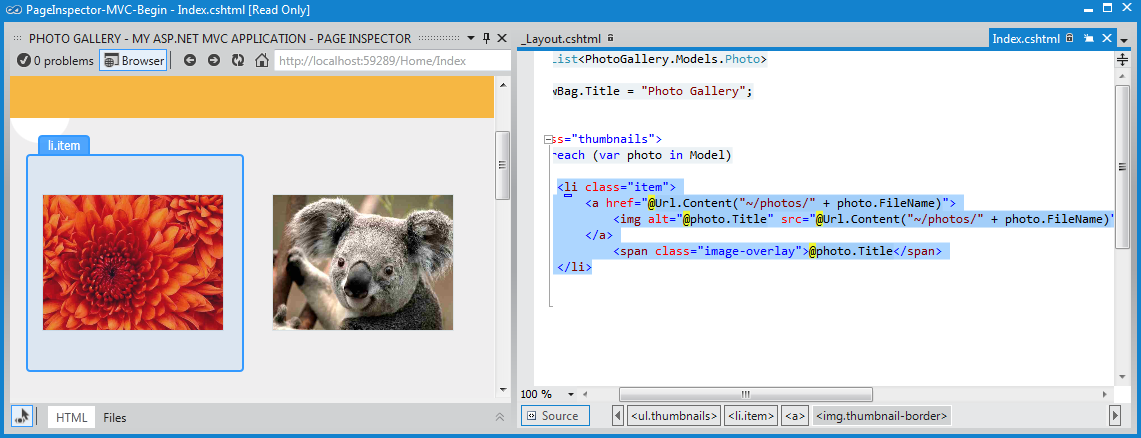
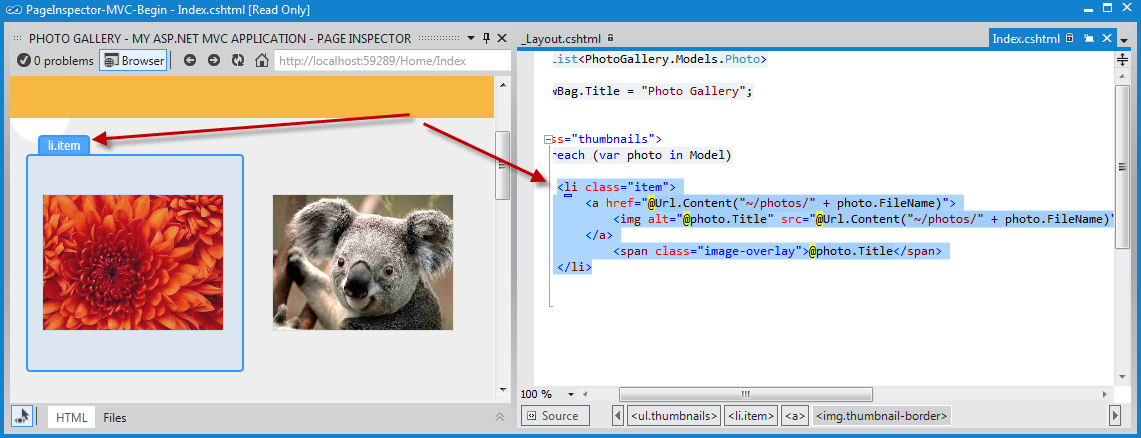
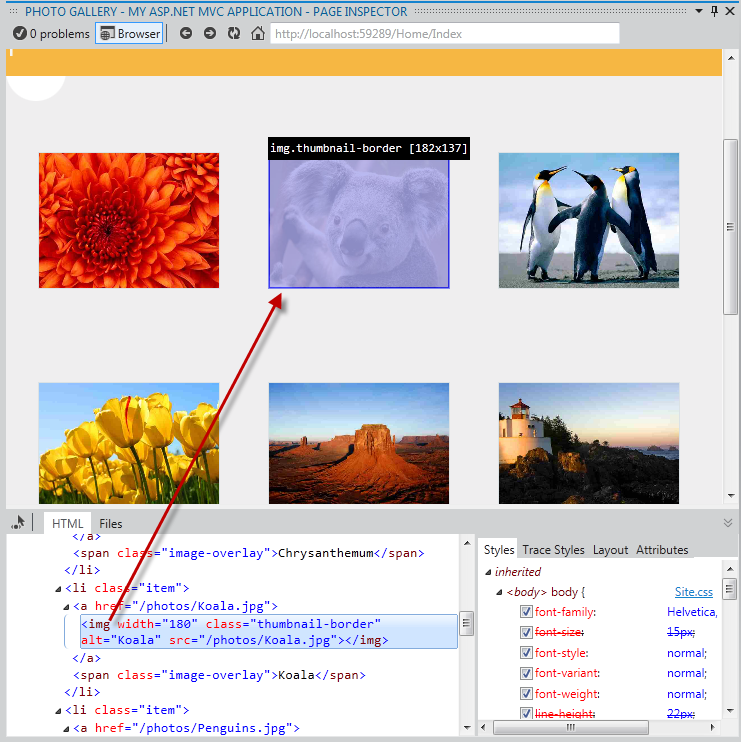
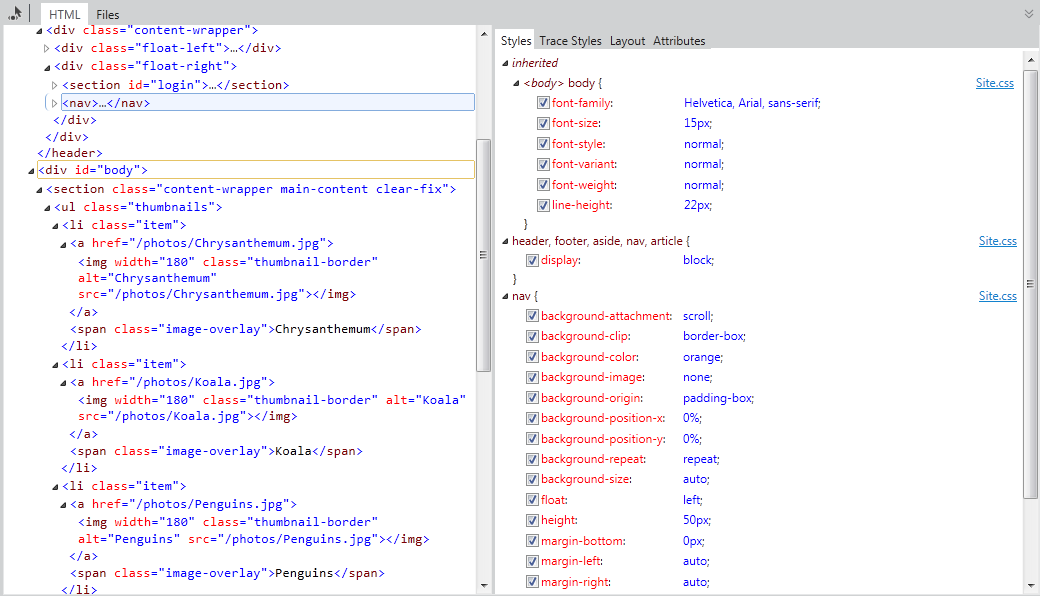
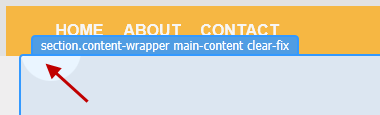
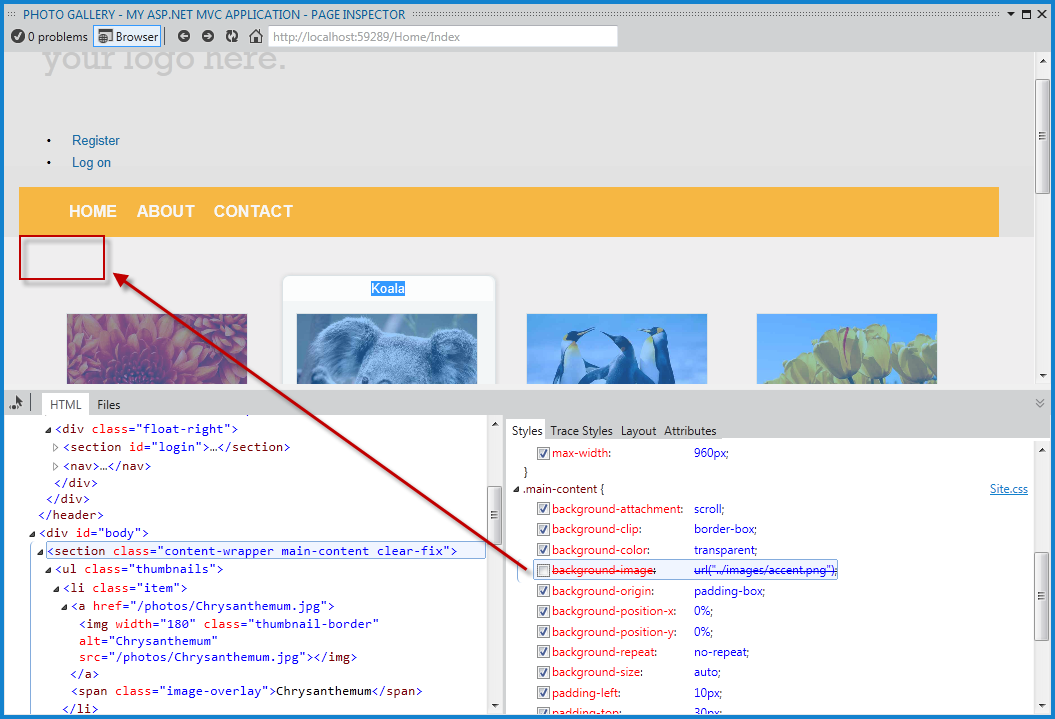
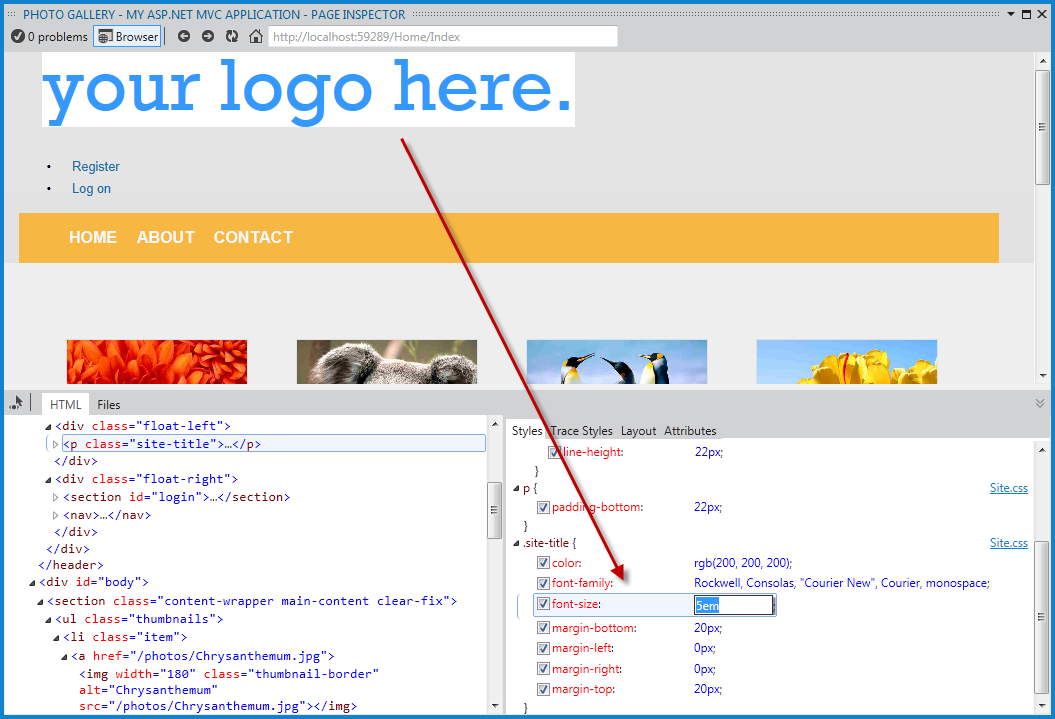
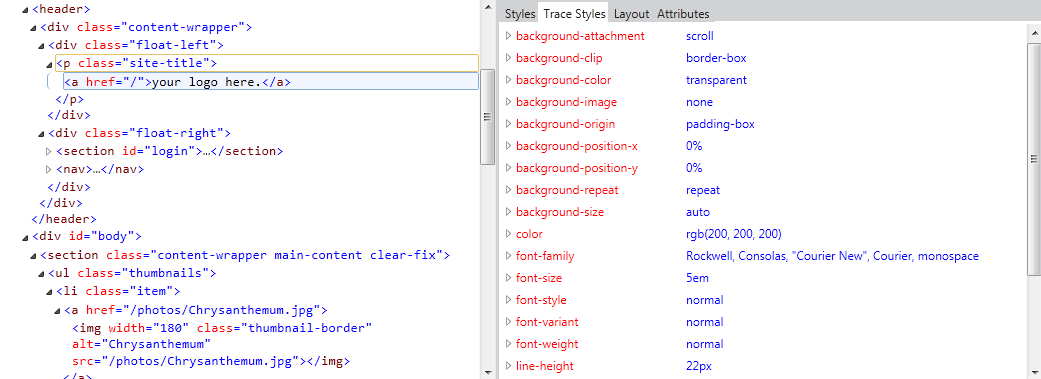
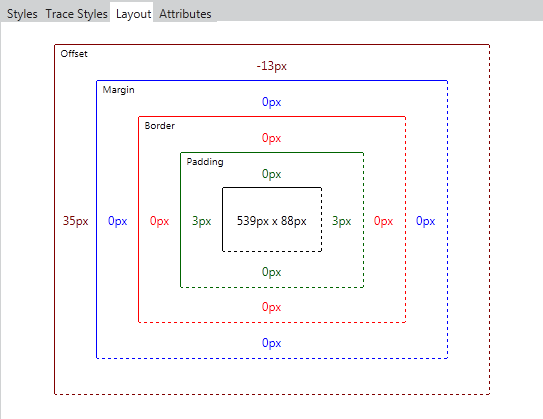
* 1. This hands-on lab includes the following exercises:
  2. Using Page Inspector in ASP.NET MVC Projects
  3. Using Page Inspector in WebForms Projects
  4. **Note:** Each exercise is accompanied by a starting solution—located in the Begin folder of the exercise—that allows you to follow each exercise independently of the others. Inside the source code for an exercise, you will also find an End folder containing a Visual Studio solution with the code that results from completing the steps in the corresponding exercise. You can use these solutions as guidance if you need additional help as you work through this hands-on lab.

Exercise 1: Using Page Inspector in ASP.NET MVC Projects

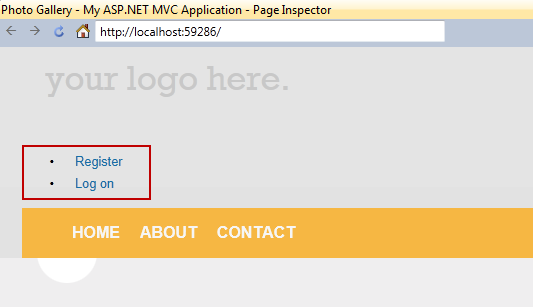
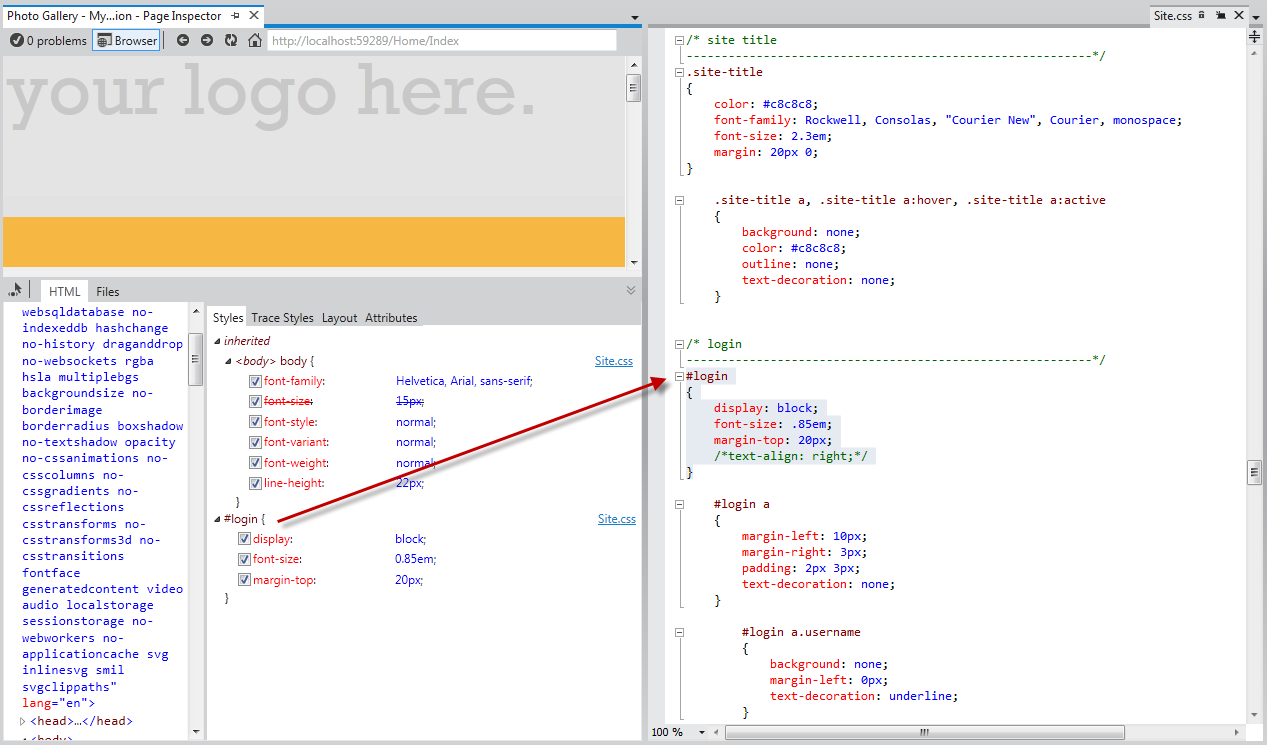
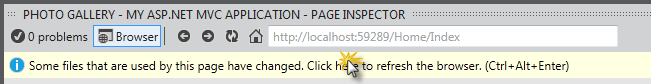
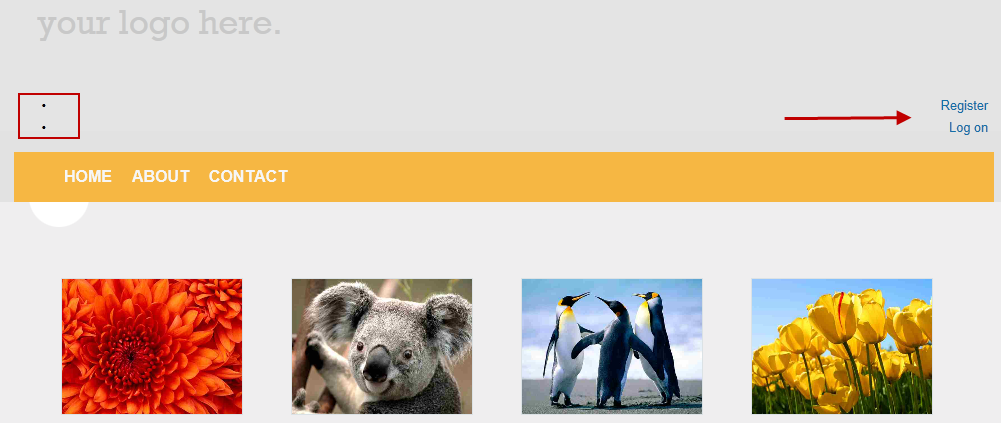
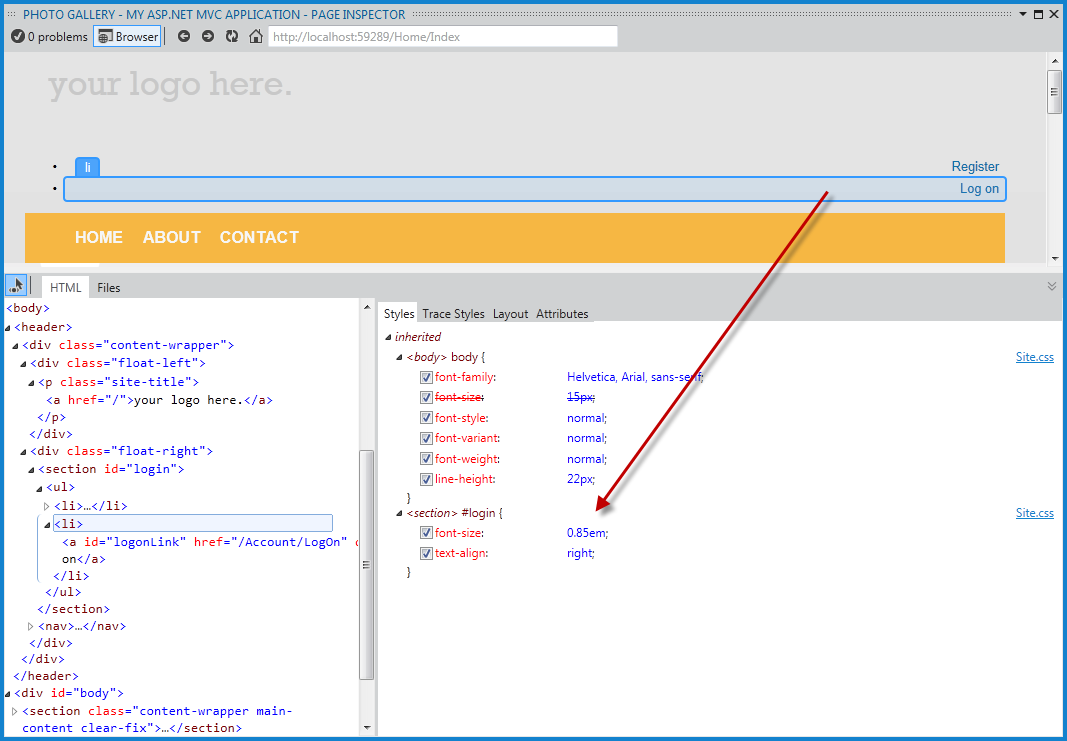
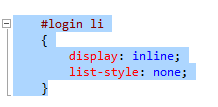
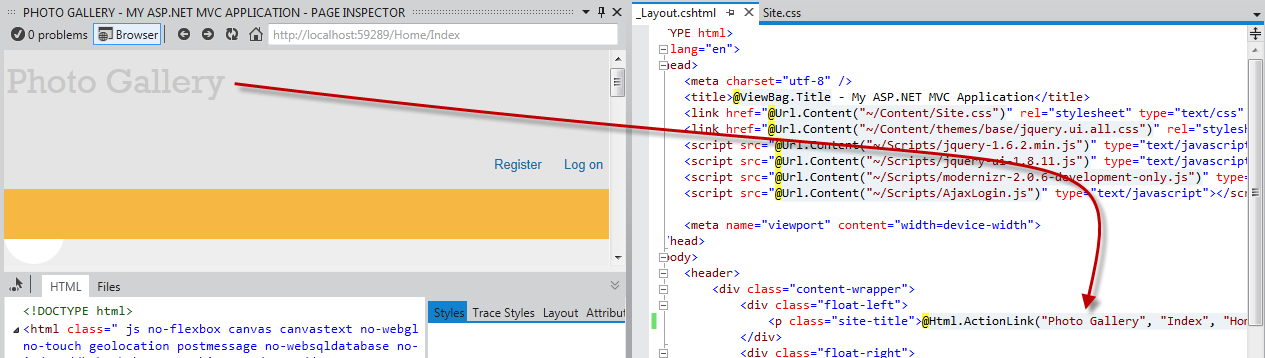
* 1. In this exercise, you will learn how to preview and debug an **ASP.NET MVC 4** solution using **Page Inspector**. First, you will perform a brief lap around the tool to learn the features that facilitate the Web debugging process. Then, you will work in a web page that contains styling issues. You will learn how to use Page Inspector to find the source code that generates the issue and fix it.

Task 1 – Exploring Page Inspector

* 1. In this task, you will learn how to use the Page Inspector in the context of an ASP.NET MVC 4 project that shows a photo gallery.
  2. Open **Visual Studio 11** and open the **PageInspector-MVC-Begin.sln** solution located in **Source\Ex1-MVC4\Begin** from this lab’s folder. You will use this solution for exploring Page Inspector.
  3. In the Solution Explorer, locate **Index.cshtml** view under the **/Views/Home** project folder, right-click it and select **View in Page Inspector**.
     1. 
     2. Figure 1
     3. Selecting a file to preview in Page Inspector
  4. You will notice that Visual Studio will prompt for permission before enabling the tool in your project. Click **YES.**
     1. 
     2. Figure 2
     3. Enabling Page Inspector in a Visual Studio project
     4. To generate the metadata that is required for some of its features, Page Inspector must instrument the project by adding the following setting to the <appSetting> element in the **Web.config** file. You can use Page Inspector even if this setting is disabled or it is missing. However, some features like the source selection mapping will not work under these conditions.
     5. Web.config
     6. <add key="VisualStudioDesignTime:Enabled" value="true" />
  5. The Page Inspector window will show the /Home/Index URL mapped to the source View you selected.
     1. 
     2. Figure 3
     3. The first contact with Page Inspector

1. The Page Inspector tool is integrated in your Visual Studio environment. The inspector contains an embedded browser, together with a powerful HTML profiler. Notice that you do not have to run the solution to see how your pages look.
   * 1. **Note:** When the width of Page Inspector browser is less than the width of the opened page, you will not see the page properly. If that happens, adjust the width of the Page Inspector.
   1. Click the **Files** tab in Page Inspector.
      1. You will see all the source files that are composing the Index page. This feature helps to identify all the elements at a glance, especially when you are working with partial views and templates. Notice that you can also open each of the files if you click the links.
      2. 
      3. Figure 4
      4. The Files tab
   2. Click the **Toggle Inspection Mode** button, located at the left of the tabs.
      1. This tool will let you select any element of the page and see its HTML and Razor code.
      2. 
      3. Figure 5
      4. Toggle Inspection Mode button
   3. In the Page Inspector browser, move the mouse pointer over the page elements. While you move the mouse pointer over any part of the rendered page, the element type is displayed and the corresponding source markup or code is highlighted in the Visual Studio editor.
      1. 
      2. Figure 6
      3. Inspection mode in action
      4. **Note:** Do not maximize the Page Inspector window or you will not be able to see the preview tab showing the source code. If you click the element in Page Inspector when it is maximized, the source code of the selection will appear but it will hide the Page Inspector window.
      5. If you pay attention to the **Index.cshtml** file, you will notice that the portion of source code that generates the selected element is highlighted. This feature facilitates the editing of long source files, providing a direct and fast way to access the code.
      6. 
      7. Figure 7
      8. Inspecting elements
   4. Click the **Toggle Inspection Mode** button () to disable the cursor.
   5. Select the **HTML** tab to display the HTML code rendered in the Page Inspector browser.
   6. In the HTML markup, locate the list item with the Koala link and select it.
      1. Notice that when you select the code, the corresponding output is automatically highlighted in the browser. This feature is useful to see how an HTML block is rendered on the page.
      2. 
      3. Figure 8
      4. Selecting HTML element in the page
   7. Click the **Toggle Inspection Mode** button to enable *Inspection Mode* and click the navigation bar. On the right of the HTML code, in the Styles pane, you will see a list with the CSS styles applied to the selected element.
      1. **Note:** Since the header is a part of the site layout, Page Inspector will also open \_Layout.cshtml file and highlight the segment of code affected.
      2. 
      3. Figure 9
      4. Discovering styles and source files of a selected element
   8. With the toggle inspection pointer enabled, move the mouse pointer below the menu bar and click the blank half circle.
      1. 
      2. Figure 10
      3. Selecting an element
   9. In the Styles pane, locate the **background-image** item under the **.main-content** group. **Uncheck** the **background-image** and see what happens. You will notice that the browser will reflect the changes immediately and the circle is hidden.
      1. **Note:** The changes you apply on the Page Inspector Styles tab do not affect the original stylesheet. You can uncheck styles or change their values as many times as you want, but they will be restored after refreshing the page.
      2. 
      3. Figure 11
      4. Enabling and disabling CSS styles
   10. Now, click the’ **your logo here’** text on the header using the inspection mode.
   11. In the **Styles** tab, locate the **font-size** CSS attribute under the **.site-title** group. Double-click the attribute value and replace the 2.3em value with **3em**, and then press **ENTER**. Notice that the title looks bigger.
       1. 
       2. Figure 12
       3. Changing CSS values in the Page Inspector
   12. Click the **Trace Styles** tab, located in the right pane of Page Inspector. This is an alternative way to see all the styles applied to the selection, ordered by attribute name.
       1. 
       2. Figure 13
       3. CSS styles tracing of the selected element
   13. Another feature of Page Inspector is the Layout pane. Using the inspection mode, select the navigation bar and then click the **Layout** tab on the right pane. You will see the exact size of the selected element, as well as its offset, margin, padding and border size. Notice that you can also modify the values from this view.
       1. 
       2. Figure 14
       3. Element layout in Page Inspector

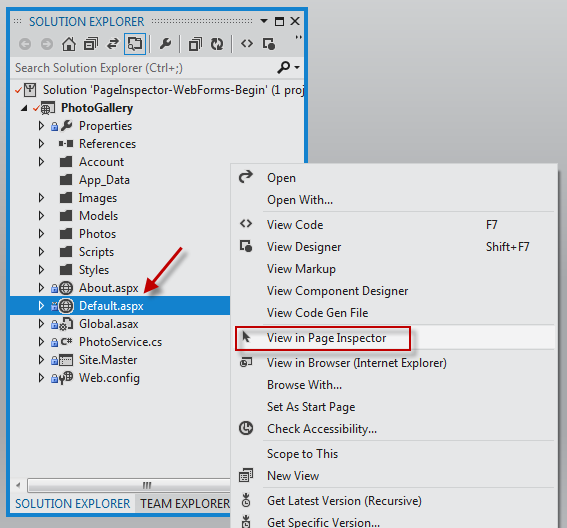
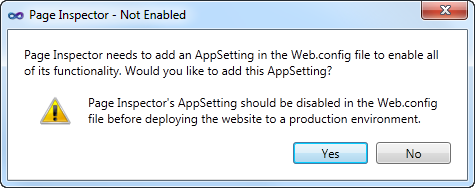
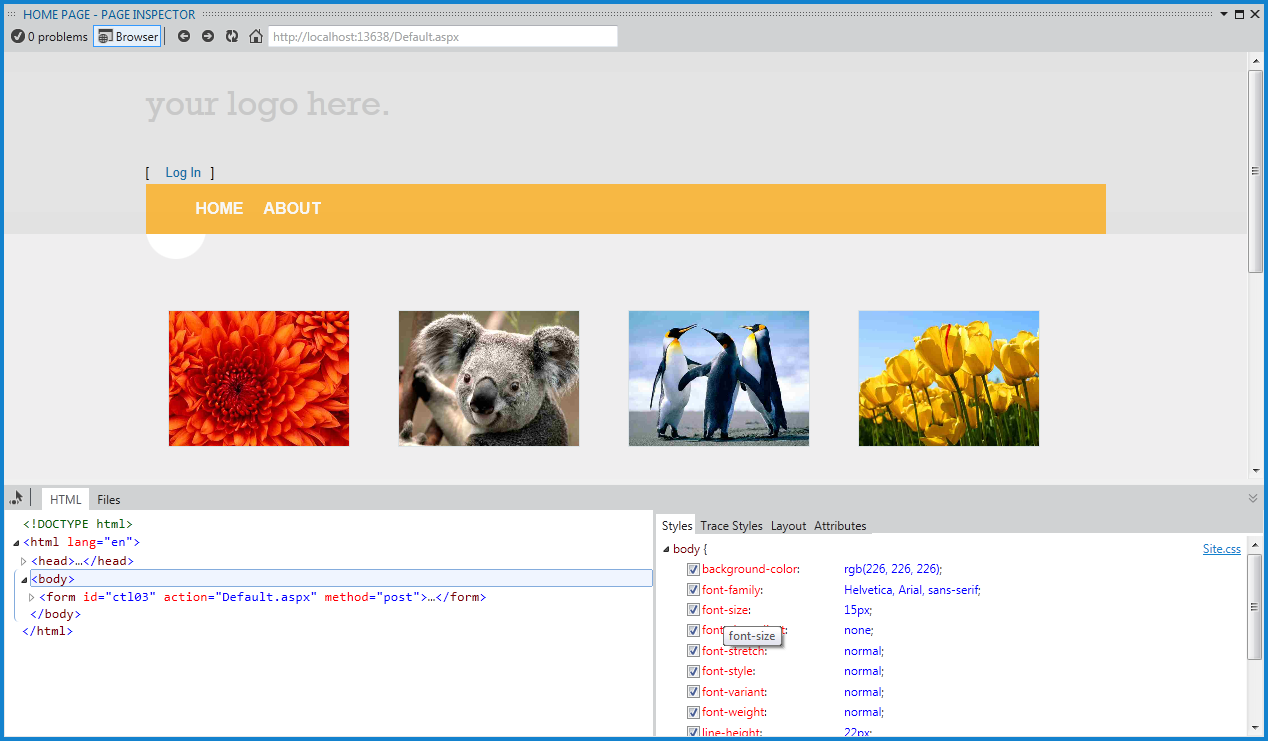
Task 2 – Finding and Fixing Style Issues in the Photo Gallery

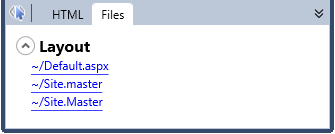
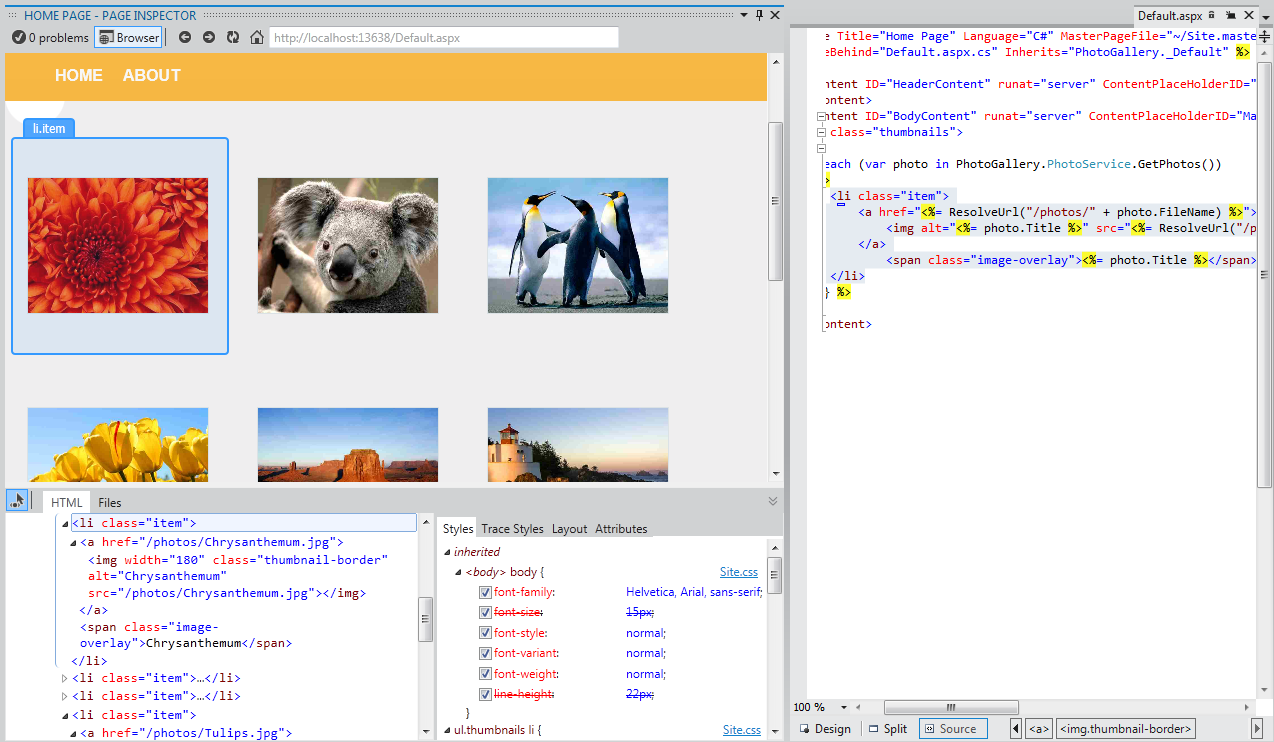
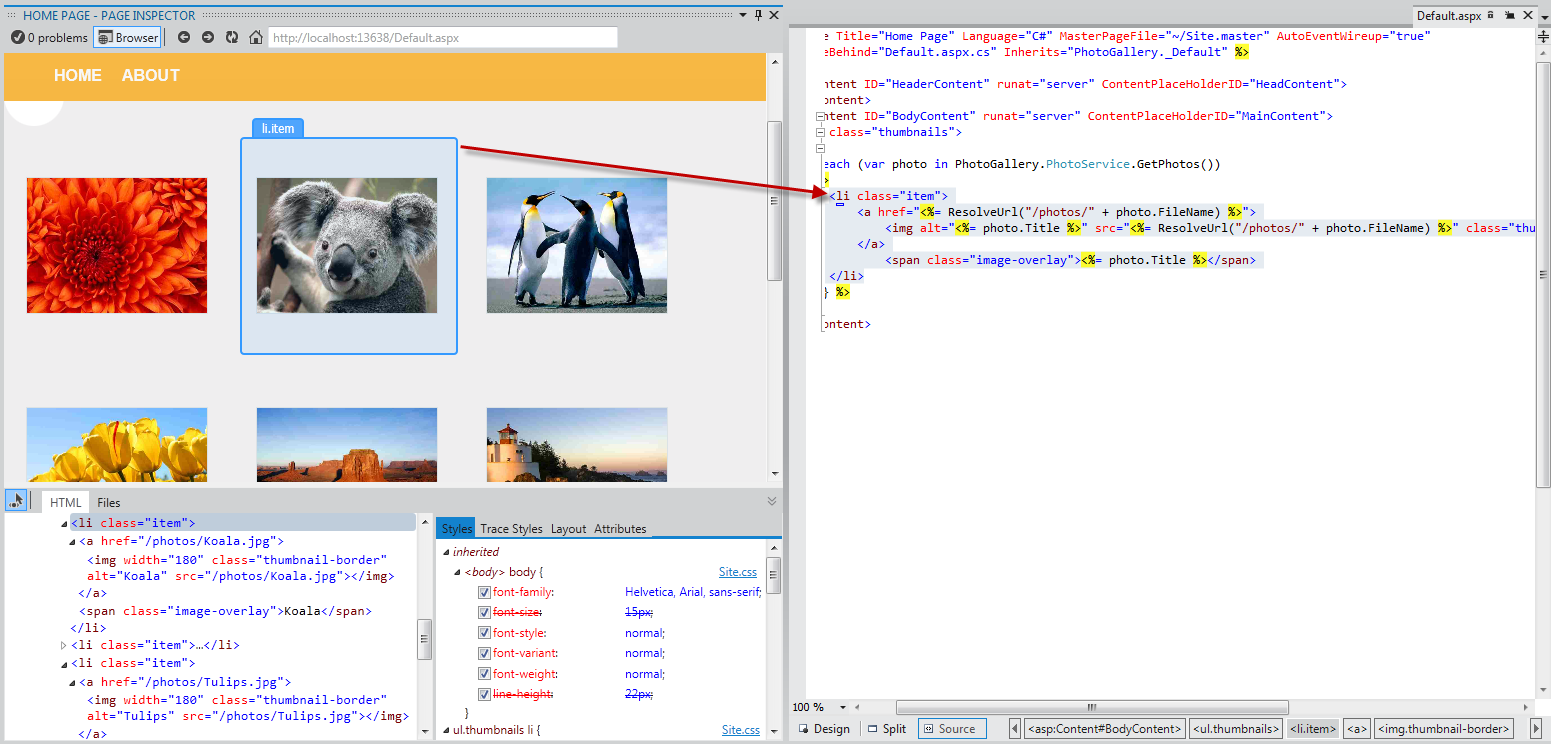
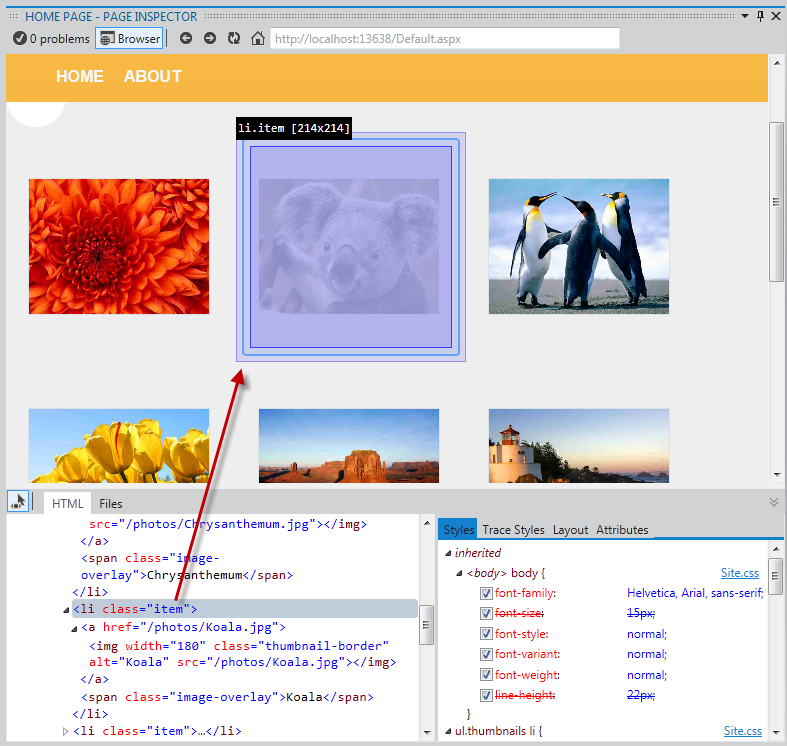
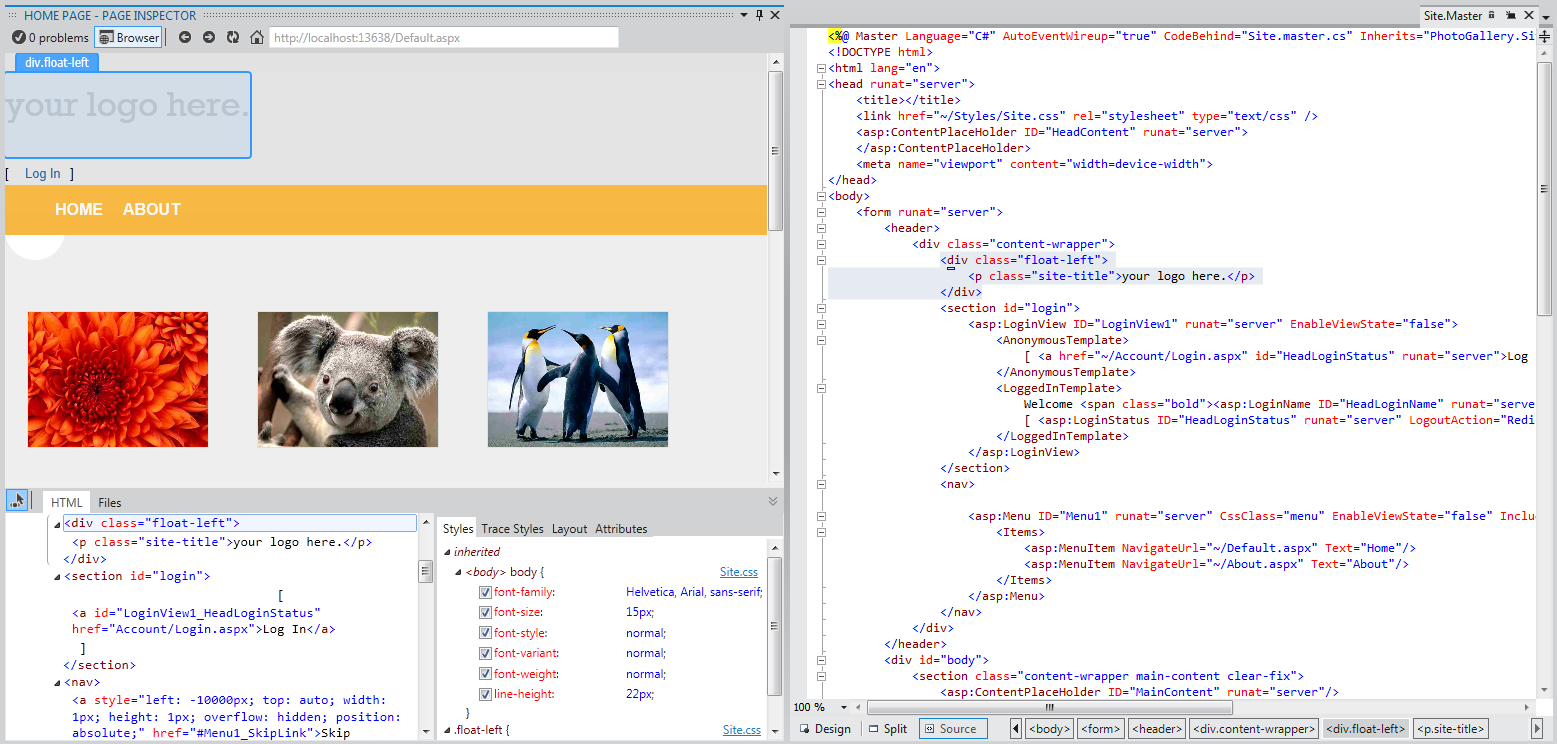
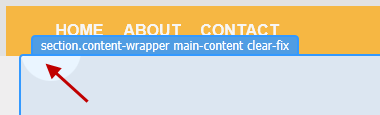
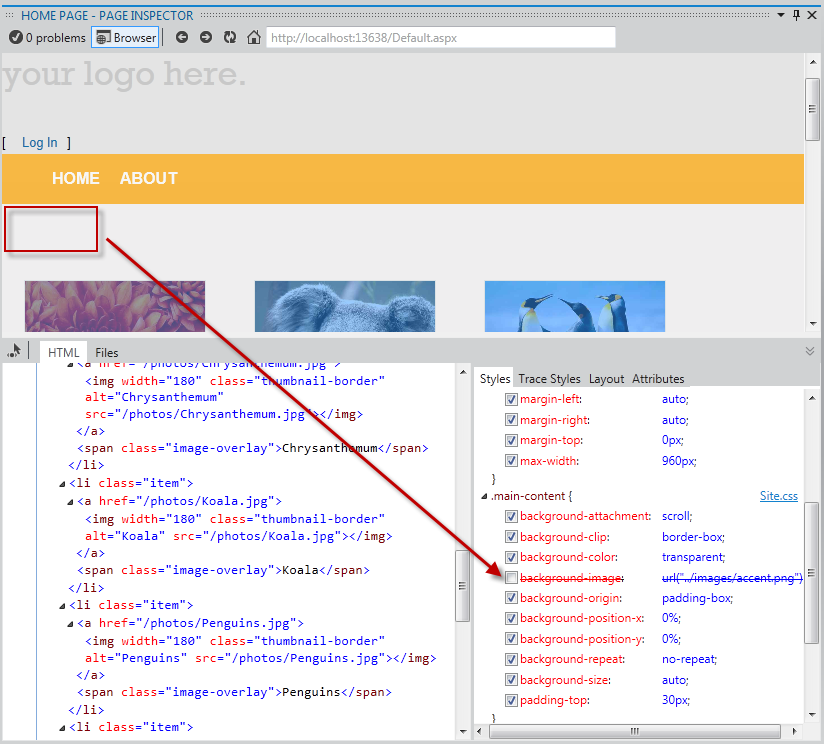
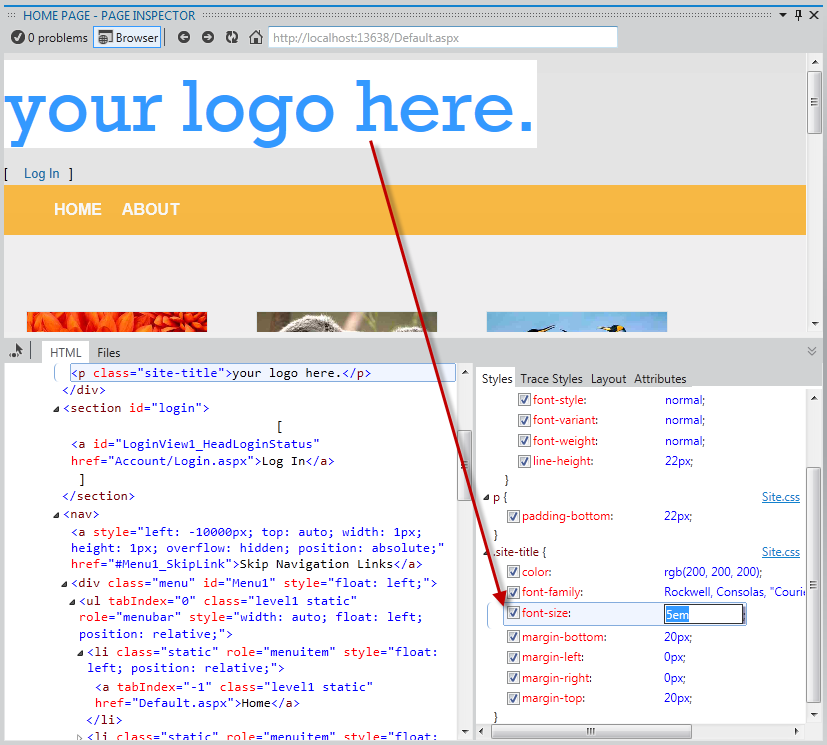
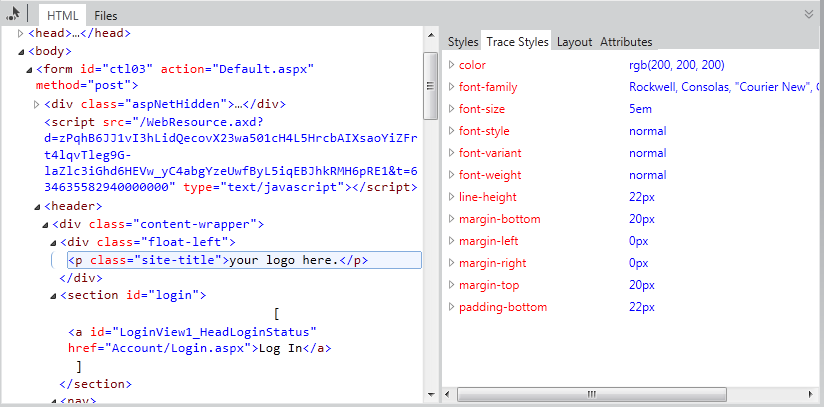
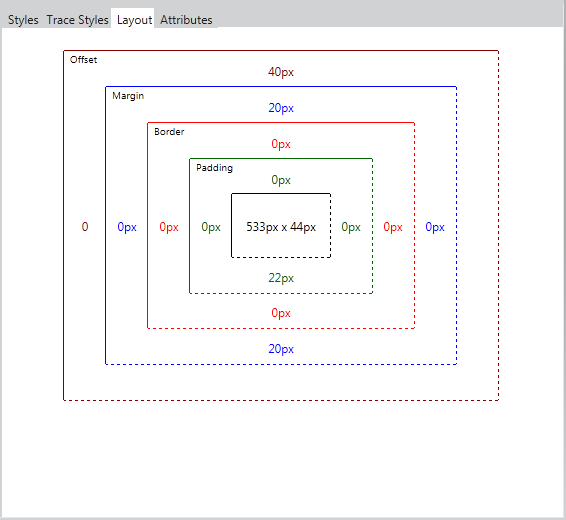
* 1. How would you diagnose Web pages issues with previous versions of Visual Studio? You are likely familiar with web debugging tools that run outside the Visual Studio IDE, like Internet Explorer Developer Tools or Firebug. Browsers only understand HTML, scripting and styles, while an underlying framework generates the HTML that will be rendered. For that reason, you often need to deploy the whole site to see how web pages look like.
  2. You had probably followed these steps when you wanted to detect and fix an issue in your web site:
  3. Run the Solution from Visual Studio, or deploy the page on the web server.
  4. In the browser, open the developer tools you use, or simply open the source code and the styles and try to match the issue.
     + To find the files involved, you would have used the “Search” or “Search in files” features with the name of the style classes.
  5. Once the error is detected, stop the Web browser and the server.
  6. Clear the browser cache.
  7. Return to Visual Studio to apply a fix. Repeat all the steps to test.
  8. As there is no real WYSIWYG in ASP.NET MVC, most of the style issues are detected on a later stage, after running or deploying the web application. Now, with Page Inspector, it is possible to preview any page without running the solution.
  9. In this task, you will use the Page inspector and fix some issues the Photo Gallery application.
  10. Using Page Inspector, locate the **Register** and the **Log on** links at the left side of the header.
      1. Notice that the links are not displayed at the expected place on the right, and they are shown like a bulleted list. You will now align the links to the right and restyle them accordingly.
      2. 
      3. Figure 15
      4. Locating the Register and Log on links
  11. With Toggle Inspection Mode selected, click close to, but not on, the Register link to open its code.
      1. Notice that the source code of the links is located in the **\_LogOnPartial.cshtml** file, not the Index.cshtml nor the \_Layout.cshtml, which are the places you might look in first place. You have been placed directly in the correct source file.
  12. In the **Styles** tab, locate and click the **<section> #login** item, which is the HTML container for these links.
      1. Notice that the **#login** style is automatically located in **Site.css** after you click. Moreover, the code is now highlighted.
      2. 
      3. Figure 16
      4. Selecting the CSS styles
  13. Uncomment the **text-align** attribute in the highlighted code by removing the opening and closing characters (/\* \*/) and save the **Site.css** file.
      1. Page Inspector is aware of all the different files that compose the current page, and it can detect when any of these files change. It alerts you whenever the current page in browser is not in sync with the source files.
  14. In the Page Inspector browser, click the bar located below the address bar to reload the page.
      1. 
      2. Figure 17
      3. Reloading the page
      4. The links are now at the right, but they still look like a bulleted list. Now, you will remove the bullets and align the links horizontally.
      5. 
      6. Figure 18
      7. Reloading the page
  15. Using the inspection mode, select any of the **<li>** items that contain the “Register” and “Log On” links. Then, click the **<section> #login** item to access Styles.css code.
      1. 
      2. Figure 19
      3. Finding the style
  16. In **Style.css**, uncomment the code for **#login li** items. The style you are adding will hide the bullet and display the items horizontally.
      1. 
      2. Figure 20
      3. Restyling the login links
  17. Save **Style.css** file and click once on the bar located below the address to reload the page. Notice that the links appear correctly.
      1. 
      2. Figure 21
      3. Links aligned to the right side
  18. Finally, you will change the header title. Use the inspection mode to click **your logo here** text and get to the source code that generates it.
  19. Now you are in **\_Layout.cshtml**, replace ‘**your logo here**’ text with ‘**Photo Gallery**’. Save and update the Page Inspector browser.
      1. 
      2. Figure 22
      3. Assigning a new title
      4. 
      5. Figure 23
      6. Photo Gallery page updated
  20. Finally press **F5** to run the app. Check out all the changes work as expected.

Exercise 2: Using Page Inspector in WebForms Projects

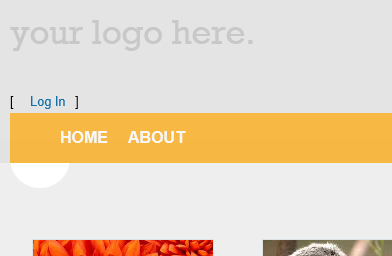
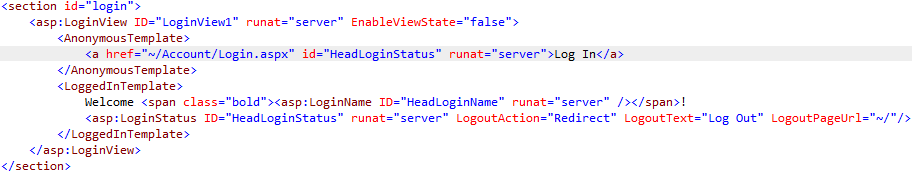
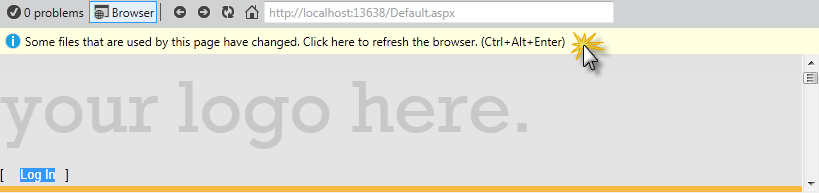
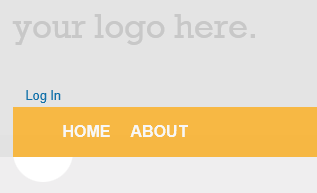
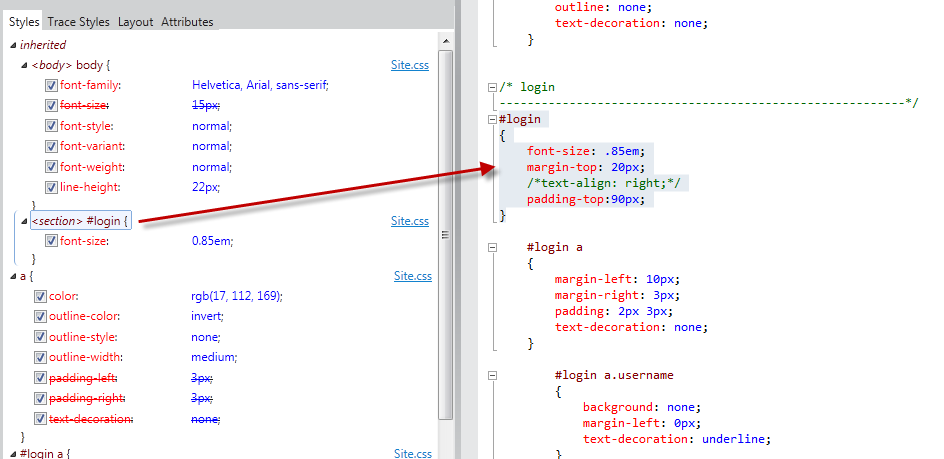
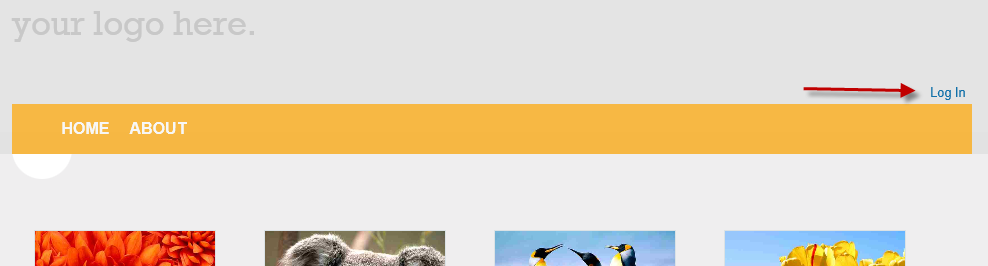
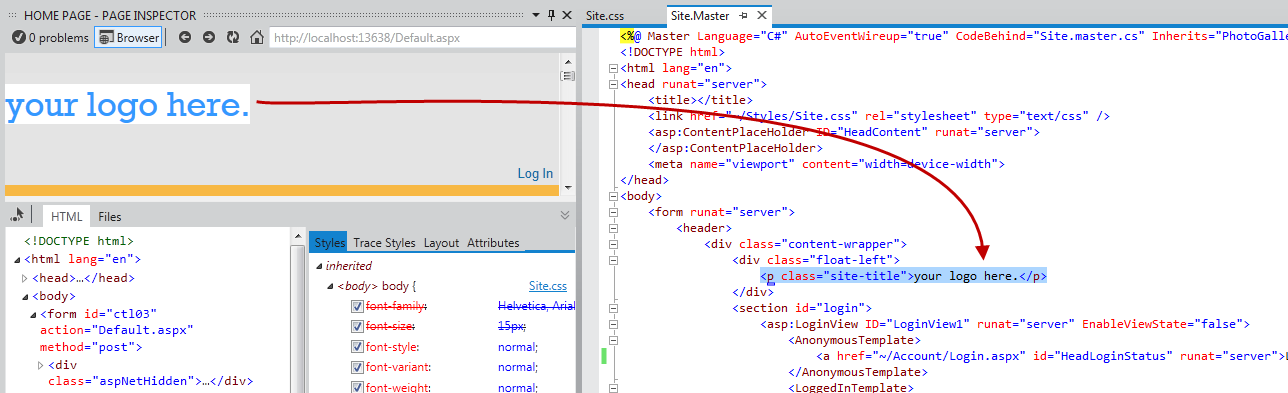
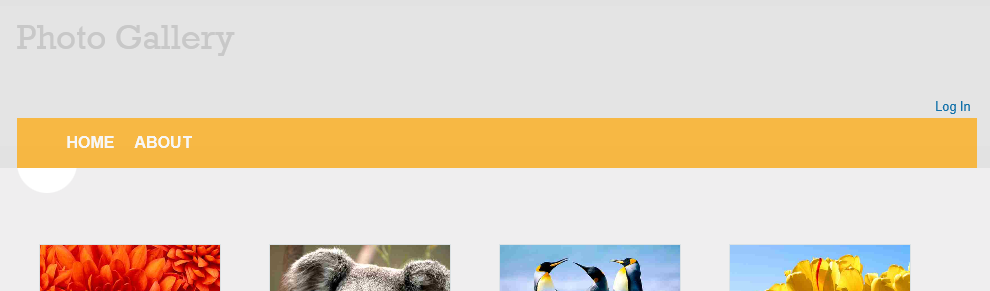
* 1. In this exercise, you will learn how to preview and debug a WebFormssolution using PageInspector. You will first perform a brief lap around the tool to learn the Page Inspector features that facilitate the Web debugging process. Then, you will work in a web page that contains styling issues. You will learn how to use Page Inspector to find the source code that generates the issue and fix it.

Task 1 – Exploring Page Inspector

* 1. In this task, you will learn how to use the Page Inspector features in the context of a WebForms project that shows a photo gallery.
  2. Open **Visual Studio 11** and open the **PageInspector-WebForms-Begin.sln** solution located in **Source\Ex2-WebForms\Begin** from this lab’s folder.
  3. In the Solution Explorer, locate **Default.aspx** page, right-click it and select **View in Page Inspector**.
     1. 
     2. Figure 24
     3. Selecting a file to preview in Page Inspector
  4. You will notice that Visual Studio will prompt for permission before enabling the tool in your project. Click **YES.**
     1. 
     2. Figure 25
     3. Enabling Page Inspector in a Visual Studio project
     4. To generate the metadata that is required for some of its features, Page Inspector must instrument the project by adding the following setting to the <appSetting> element in Web.config file. You can use Page Inspector even with this setting disabled or missing. However, some features will not work, such as source selection mapping.
     5. Web.config
     6. <add key="VisualStudioDesignTime:Enabled" value="true" />
  5. The Page Inspector window will show Default.aspx.
     1. 
     2. Figure 26
     3. The first contact with Page Inspector

1. The Page Inspector tool is integrated in your Visual Studio environment. The inspector contains an embedded browser, together with a powerful HTML profiler that will show the selected code. Notice that you do not have to run the solution to see how your pages look.
   * 1. **Note:** When the width of Page Inspector browser is less than the width of the opened page, you will not see the page properly. If that happens, adjust the width of the Page Inspector.
   1. Click the **Files** tab in Page Inspector.
      1. You will see all the source files that are composing the rendered Default page. This is a useful feature to identify all the elements at a glance, especially when you are working with User Controls and Master Pages. Notice that you can also navigate to each of the files.
      2. 
      3. Figure 27
      4. The Files tab
   2. Click the **Toggle Inspection Mode** button, located at the left of the tabs.
      1. This tool will let you select any element of the page and see its HTML code and .aspx source.
      2. 
      3. Figure 28
      4. Toggle Inspection Mode button
   3. In the Page Inspector browser, move the mouse over the page elements. While you move the mouse pointer over any part of the rendered page, the element type is displayed and the corresponding source markup or code is highlighted in the Visual Studio editor.
      1. 
      2. Figure 29
      3. Inspection mode in action
      4. **Note:** Do not maximize the Page Inspector window or you will not be able to see the preview tab showing the source code. If you click the element in Page Inspector when it is maximized, the source code of the selection will appear but it will hide the Page Inspector window.
      5. If you pay attention to **Default.aspx** file, you will notice that the portion of source code that generates the selected element is highlighted. This feature facilitates the edition of long source files, providing a direct and fast way to access the code.
      6. 
      7. Figure 30
      8. Inspecting elements
   4. Click the **Toggle Inspection Mode** button (), located in Page Inspector tabs, to disable the cursor.
   5. Select the **HTML** tab to display the HTML code rendered in the Page Inspector browser.
   6. In the HTML code, locate the list item with the Koala link and select it.
      1. Notice that when you select the code, the corresponding output is automatically highlighted browser. This feature is useful to see how an HTML block is rendered on the page.
      2. 
      3. Figure 31
      4. Selecting an HTML element in the page
   7. Click the **Toggle Inspection Mode** button to enable *Inspection Mode* and click the navigation bar. On the right of the HTML code, in the Styles pane, you will see a list with the CSS styles applied to the selected element.
      1. **Note:** since the header is a part of the site layout, Page Inspector will also open Site.Master file and highlight the segment of code affected.
      2. 
      3. Figure 32
      4. Discovering styles and source files of a selected element
   8. With the toggle inspection pointer enabled, move the mouse pointer below the menu bar and click the blank half circle.
      1. 
      2. Figure 33
      3. Selecting an element
   9. In the Styles pane, locate the **background-image** item under the **.main-content** group. **Uncheck** the **background-image** and see what happens. You will notice that the browser will reflect the changes immediately and the circle is hidden.
      1. **Note:** The changes you apply on the Page Inspector Styles tab do not affect the original stylesheet. You can uncheck styles or change their values as many times as you want, but they will be restored after refreshing the page.
      2. 
      3. Figure 34
      4. Enabling and disabling CSS styles
   10. Now, click the ‘**your** **logo here’** text on the header using the inspection mode.
   11. In the **Styles** tab, locate the **font-size** CSS attribute under the **.site-title** group. Click the attribute once to edit its value. Replace the 2.3em value with **5em**, and then press ENTER. Notice that the title looks bigger.
       1. 
       2. Figure 35
       3. Changing CSS values in the Page Inspector
   12. Click the **Trace Styles** tab, located in the right pane of Page Inspector. This is an alternative way to see all the styles applied to the selection, ordered by attribute name.
       1. 
       2. Figure 36
       3. CSS styles tracing of the selected element
   13. Another feature of Page Inspector is the Layout pane. Using the inspection mode, select the navigation bar and then click the **Layout** tab on the right pane. You will see the exact size of the selected element, as well as its offset, margin, padding and border size. Notice that you can also modify the values from this view.
       1. 
       2. Figure 37
       3. Element layout in Page Inspector

Task 2 – Finding and Fixing Style Issues in the Photo Gallery

* 1. How would you diagnose Web pages issues with previous versions of Visual Studio? You are likely familiar with web debugging tools that run outside the Visual Studio IDE, like Internet Explorer Developer Tools or Firebug. Browsers only understand HTML, scripting and styles, while an underlying framework generates the HTML that will be rendered. For that reason, you often need to deploy the whole site to see how web pages look like.
  2. You had probably followed these steps when you wanted to detect and fix an issue in your web site:
  3. Run the Solution from Visual Studio, or deploy the page on the web server.
  4. In the browser, open the developer tools you use, or simply open the source code and the styles and try to match the issue.
     + To find the files involved, you’d have used the “Search” or “Search in files” features with the name of the style classes.
  5. Once the error is detected, stop the Web browser and the server.
  6. Clear the browser cache.
  7. Return to Visual Studio to apply a fix. Repeat all the steps to test.
  8. As there is no real WYSIWYG in ASP.NET WebForms, some style issues are detected on a later stage, after running or deploying. Now, with Page Inspector, it is possible to preview any page without running the solution.
  9. In this task, you will use the Page inspector for fixing some issues the Photo Gallery application. In the following steps, you will detect and quickly fix some simple styling issue in the header.
  10. Using Page Inspection, locate the **Log In** link at the left side of the header.
      1. Notice that the link is not displayed at the expected place on the right. You will now align the link to the right and restyle it accordingly.
      2. 
      3. Figure 38
      4. Log on link positioned on the left
  11. With Toggle Inspection Mode selected, select the Log In link to open its code.
      1. Notice that the link source code is located in the **Site.Master** file, not in the Default.aspx page which is the place you might look in first place; you have been placed directly in the correct source file.
  12. The page is using the LoginView control in the same way it is used by default in the Web Application project template. To make it look nicer, remove the brackets between the Log In and Log Out controls.
      1. 
      2. Figure 39
      3. Brackets Removed
      4. Page Inspector is aware of all the different files that compose the current page, and it can detect when any of these files change. It alerts you whenever the current page in browser is not in sync with the source files.
  13. In the Page Inspector browser, click the bar located below the address bar to save the changes and reload the page.
      1. 
      2. Figure 40
      3. Reloading the page
      4. The Log In link looks nicer without the brackets, now, you will move it to the right side. To do this, you will locate the styles being applied to the Login section and update it to align its content to the right.
      5. 
      6. Figure 41
      7. Reloading the page
  14. In the **Styles** tab, locate and click the **<section> #login** item, which is the HTML container for the link.
      1. Notice that **Site.css** opened and the corresponding style is selected.
      2. 
      3. Figure 42
      4. Selecting the CSS styles
  15. Uncomment the **text-align** attribute in the highlighted code by removing the opening and closing characters (/\* \*/) and save the **Site.css** file.
      1. Notice that the page inspector will prompt again for showing and refreshing.
  16. In the Page Inspector browser, click once on the bar located below the address bar to reload the page.
      1. 
      2. Figure 43
      3. Moving the Log In link to the right
  17. Finally, you will change the header title. Instead of searching for the ‘**your logo here’** text in all the files, use the inspection mode to click the text and get to source code that generates it.
      1. 
      2. Figure 44
      3. Finding the site title
  18. Now you are in **Site.Master**, replace the ‘**your logo here**’ text with ‘**Photo Gallery**’. Save and update the Page Inspector browser.
      1. 
      2. Figure 45
      3. Photo Gallery page updated
  19. Finally press **F5** to run the app the check out all the changes work as expected.

Summary

* 1. By completing this Hands-On Lab, you have learnt how to use Page Inspector to preview your Web application without having to rebuild and run the Web site in a browser. In addition, you have learnt how to quickly find and fix bugs by accessing directly from the rendered output to the source code.