

ITEC447

Web Projects

CHAPTER 8 – DYNAMIC WEB TEMPLATES

Dynamic Web Templates

- Most websites consist of more than a single page.
- The whole idea behind creating the World Wide Web was the ability to make numerous documents available and then link them together rather than presenting them all at the same time.
- But this causes a problem: If you have a website with multiple pages and you want to make a design change to all these pages, you have to update each page individually.
- If your site has only a few pages this is not a problem, but what if it has tens or even hundreds of pages?
- A DWT is a special type of file built using HTML and CSS to define which areas of a page a publisher might edit and which areas are off limits to regular page building.
- After a DWT exists, you can use it to build new pages in which all you need to do is input new content in the predefined areas without having to worry about all the common elements present in every page.
- More important, when you have multiple pages built using a DWT, you can change the composition and layout of all the pages by making changes to only the DWT file.
- In other words, using DWTs makes global alterations to a website a snap.

Dynamic Web Templates Cont.

- When you design a website, you should always consider the following question.
- How do I make sure it is easy to update the look and functionality of every page within the site?
- Depending on the scale of the project and the kind of content you are presenting, the answer to this question is different.
- If you are creating a small-scale site with only a few pages (fewer than ten) and multiple different layouts and designs, you can go with straightforward HTML pages with one or several style sheets attached.
- If your site is (or could become) larger and you have one or two layouts to implement sitewide, creating pages based on DWTs is an effective solution.
- If your site has a high number of pages or constantly updated dynamic content (think an online paper or magazine, a forum, a blog, or a site with multiple authors), the best solution is to use a Content Management System (CMS) that generates pages dynamically with server-side script and a database.
- For the large majority of sites, the second and third options are the best choice because they provide easy page construction and quick sitewide design changes.

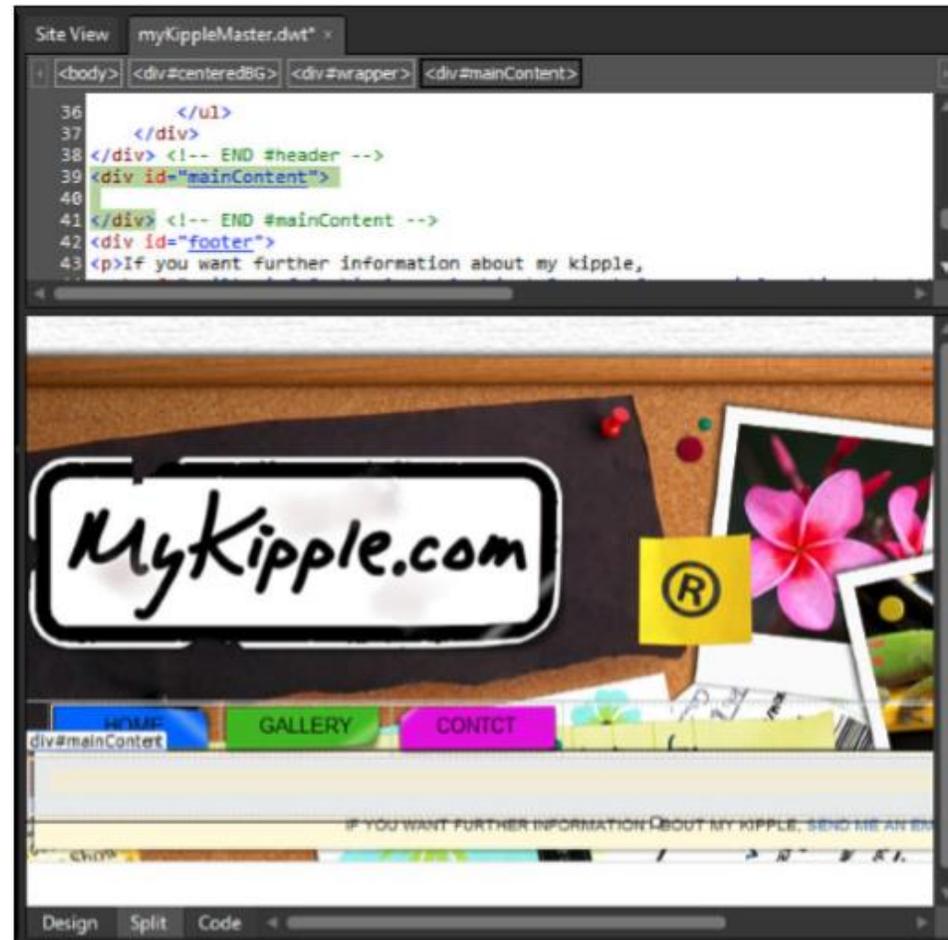
Creating a Dynamic Web Template

- To get a firmer grasp on what a DWT is and how it works, you build a DWT for the *MyKipple.com* website based on the *default.html* page.
- Before you begin, replace the current *default.html*, *kippleStyles.css*, and *layout.css* files in your site with the new ones found in the lesson files for hour 19.
- The new files include more styling to make the overall look and feel of the pages more polished.
- All the changes to the files are based on the lessons you have already completed.
- If you take a closer look at the code in the *kippleStyles.css* file, you see that you can recognize most of the different styling elements.

Creating a Dynamic Web Template Cont.

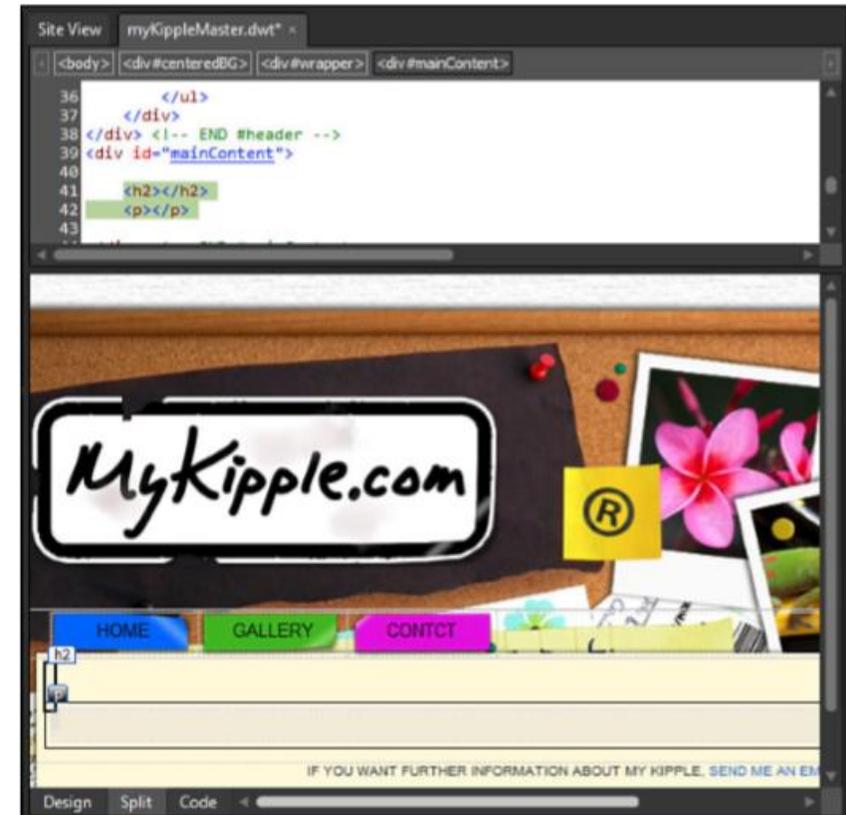
- With the new *default.html* page open in *Design* view, click *File, Save As* on the menu bar.
- In the *Save As* dialog, change the *file type* to *Dynamic Web Template (.dwt)* and name the new file *mykippleMaster.dwt*.
- This creates a new DWT.
- With the *mykippleMaster.dwt* page open in *Split* view, delete all the text content in the *#mainContent* div.
- You can do this by highlighting the text and deleting it in *Design* view, or by highlighting all the content between the beginning and end `<div id="mainContent">` tags and deleting it in *Code* view.
- Make sure you leave the *#centeredBG*, *#wrapper*, *#header*, *#mainMenu*, *#content*, and *#footer* divs intact.

Creating a Dynamic Web Template Cont.



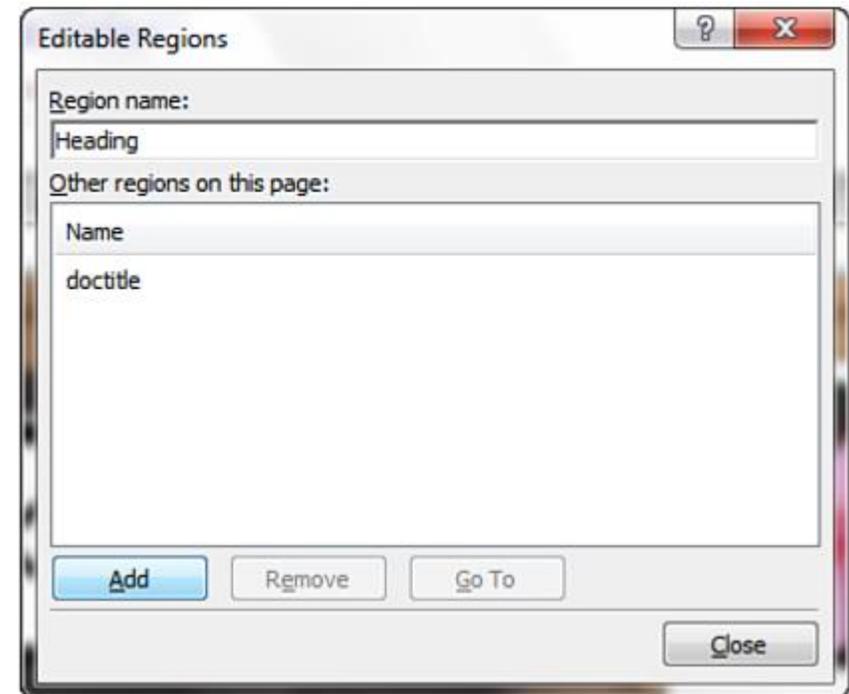
Creating a Dynamic Web Template Cont.

- Place the cursor on the first line inside the `#mainContent` div, and use the *Style* drop-down menu from the *Common* toolbar to set the style of the line to *Heading 2 (h2)*.
- Press *Enter* to create a new paragraph underneath.



Creating a Dynamic Web Template Cont.

- Place the cursor inside the first line (h2) and from the menu bar select *Format, Dynamic Web Templates, Manage Editable Regions*.
- The *Editable Regions* dialog lets you add and remove editable regions within your DWT.
- Under *Region Name*, type *heading* and click the *Add* button.
- This creates a new editable region called heading.
- Click *Close* to apply the changes.



Creating a Dynamic Web Template Cont.

- Place your cursor inside the paragraph on the next line, and click the `<p>` tag in the *Quick Tag Selector* to select the entire line, including the tags.
- From the menu bar, select *Format, Dynamic Web Template, Manage Editable Regions* to open the *Editable Regions* dialog again.
- Create a new editable region called *content* and click *Close* to apply the changes.
- You now have two regions within the page, outlined in orange in *Design* view.

Creating a Dynamic Web Template Cont.

The screenshot displays a web design tool interface for a file named 'myKippleMaster.dwt'. The top section shows the 'Site View' with a breadcrumb trail: <body> <div #centeredBG> <div #wrapper> <div #mainContent> <DWT:editable> <p>. Below this, the code editor shows the following HTML structure:

```
37 </div>
38 </div> <!-- END #header -->
39 <div id="mainContent">
40
41 <h2><!-- #BeginEditable "Heading" -->(Heading)<!-- #EndEditable --></h2>
42 <!-- #BeginEditable "Content" -->
43 <p>(Content)</p>
44 <!-- #EndEditable -->
```

The design view below the code shows a corkboard-themed layout. A large white rounded rectangle contains the text 'MyKipple.com' in a handwritten font, with a yellow registered trademark symbol (®) to its right. Below the logo is a navigation bar with three buttons: 'HOME' (blue), 'GALLERY' (green), and 'CONTACT' (purple). The main content area is a light yellow box containing a heading placeholder '(HEADING)' and a content placeholder '(Content)'. A line from the label 'Heading editable region' points to the heading placeholder. Another line from the label 'Content editable region' points to the content placeholder. At the bottom of the design view, there is a small text snippet: 'IF YOU WANT FURTHER INFORMATION ABOUT MY KIPPLE, SEND ME AN EM'. The tool's bottom status bar shows 'Design Split Code'.

How Dynamic Web Templates Work

- Now that you have created a DWT, let's look at how it works.
- Looking at the page in *Code* view, you see that it looks pretty much the same as the original file with the exception of a few new lines of commented-out code.
- The new elements tell Expression Web 4 that a specific DWT controls the page.
- The elements are commented out because they are not HTML code but rather custom script designed specifically to work with Expression Web 4.
- As a result, they have no actual function when a browser displays the page.
- Also, like other commented content, the code is ignored.
- But when the application opens the page, they link the DWT and its children together.

How Dynamic Web Templates Work

Cont.

- In the body of the document are several code segments that tell Expression Web 4 which areas are locked and which areas are editable.
- When you create a DWT from scratch, you get a standard HTML page with two editable regions: *doctitle* and *body*.
- As the names suggest, the *doctitle* region holds the `<title>` tag in the head portion of the page, and the *body* region holds the *body* of the page.
- The same thing happened when you converted the *default.html* page to a DWT.
- Expression inserted the *doctitle* region into your page at the end of the `<head>` tag.

How Dynamic Web Templates Work

Cont.

- The beginning and end codes are commented out because they are not HTML and their sole purpose is to help Expression Web 4 to define the editable regions.
- For example, the *doctitle* editable region looks like this in *Code* view:

```
<!-- #BeginEditable "doctitle" -->  
<!-- #EndEditable -->
```

How Dynamic Web Templates Work Cont.

- Expression Web 4 considers anything within these two code snippets editable.
- Because code that is invisible to browsers defines the editable regions, you can choose how much you want to micromanage the content within them.
- The two editable regions you inserted in the *mykippleMaster.dwt* page serve as good examples of this.
- If you select the *heading* region in *Design* view and look at the code, you see that the editable region is contained within the *h2* tags:

```
<h2><!-- #BeginEditable "heading" -->(heading)<!-- #EndEditable --></h2>
```

How Dynamic Web Templates Work

Cont.

- That means whatever content you place inside the editable region will be styled with the *h2* style.
- As a result, when building a page based on this DWT, the designer cannot change the style of this content.
- Expression Web 4 placed the editable region inside the `<h2>` tags because you placed the cursor inside the *h1* area before inserting it.
- With the paragraph, the content region is placed on the outside of the style tags, like this:

```
<!-- #BeginEditable "content" --><p>(content)</p><!-- #EndEditable -->
```

How Dynamic Web Templates Work

Cont.

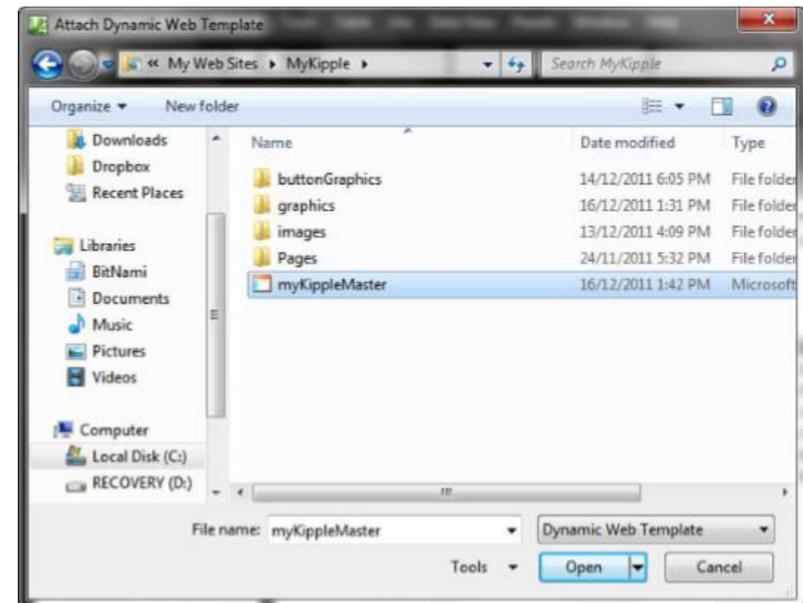
- As a result, the content within the region is not yet styled and the designer can apply other tags and styles at will.
- Understanding the difference between these two methods of inserting editable regions means that the designer of the DWT has almost unlimited control of the output that comes from pages created with the template.

Create a New Page from a DWT

- When you have a DWT, creating new pages for your project becomes much easier.
- The DWT contains all the common elements that all pages should feature, and all you need to do is insert the page-specific content.
- Because HTML is the basis for the DWT, all the CSS styling you attach to the DWT is available in your new page.

Create a New Page from a DWT Cont.

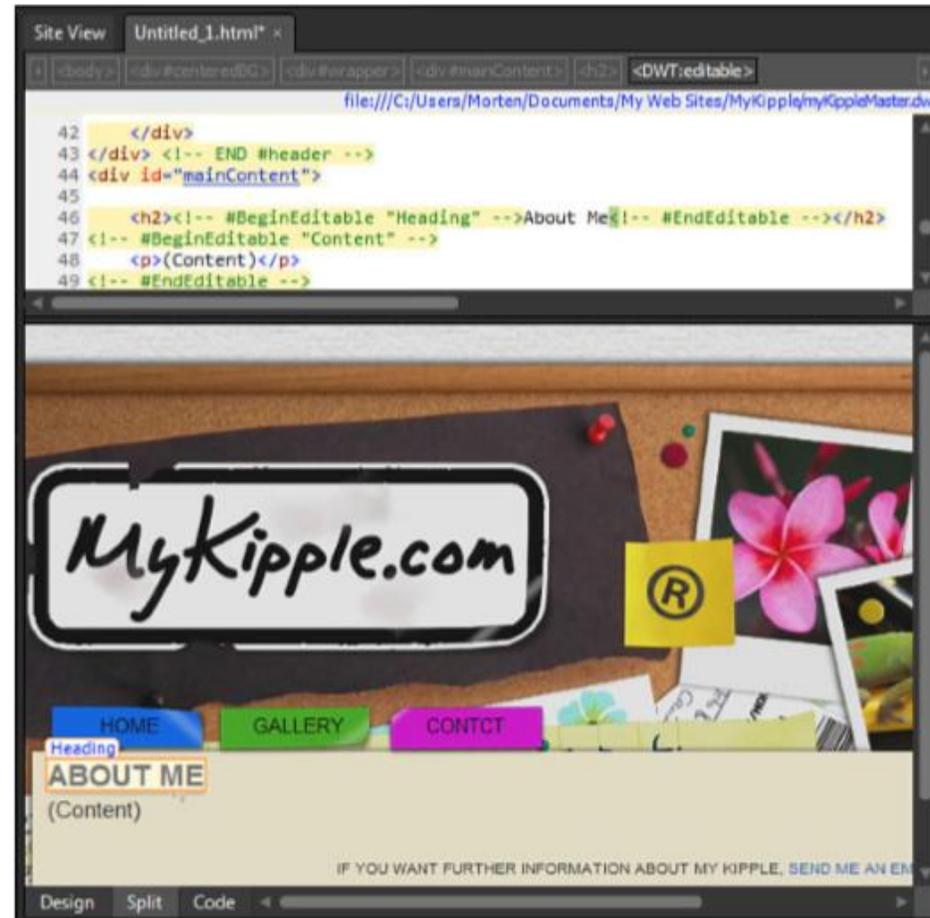
- To create a new page from a DWT, select *File, New, Create from Dynamic Web Template* from the menu bar.
- When you create a page from a DWT, the *Attach Dynamic Web Template* dialog opens.
- Select the *myKippleMaster.dwt* file and click *Open*.
- An information box opens to report that a number of files updated; this refers to the content of the DWT populating the new file.



Create a New Page from a DWT Cont.

- With the new page open in *Design* view, you now have only two clickable areas in the page: the *heading* and the *content*.
- If you move your cursor anywhere else, all you get is a stop sign.
- Place your cursor in the heading area, remove the existing text, and type *About Me*.
- Expression Web 4 automatically applies the *h2* style.
- If you press *Enter* to create a new line, a warning pops up to tell you that you can't make the change because a DWT is locking the code.

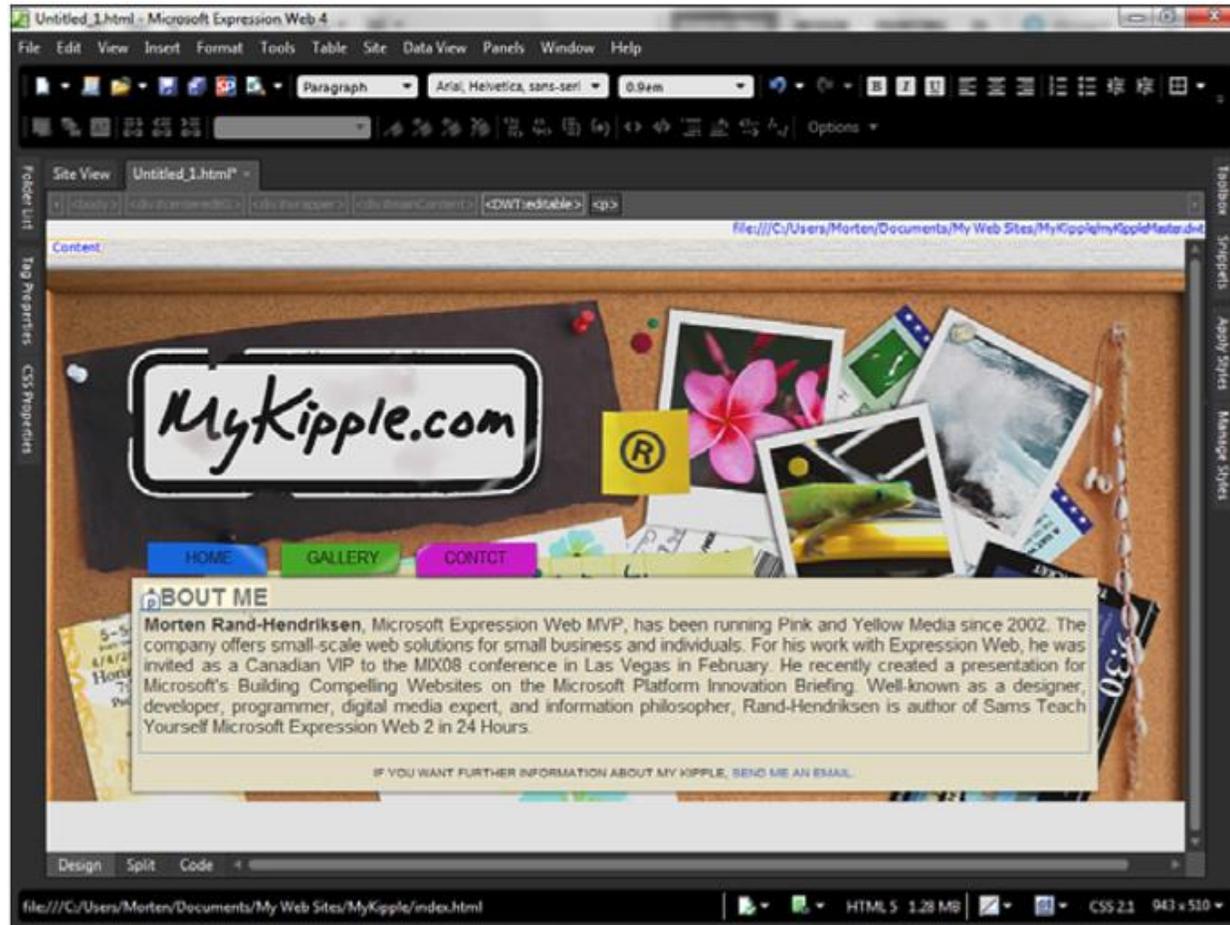
Create a New Page from a DWT Cont.



Create a New Page from a DWT Cont.

- Place your cursor in the content region, remove the existing text, and write a short bio about yourself or insert some other content of your choice.
- Because this region is not contained within a tag, it is not constrained in the same way the heading region is.
- Therefore, you can add several paragraphs, images, or any other HTML content as you place and style it using the styles and classes in the attached style sheet.
- When you finish inserting content in the page, press *Ctrl+S* to save the file.
- In the *Save As* dialog, go to the *Pages* folder and give the new page the name *about.html*.

Create a New Page from a DWT Cont.

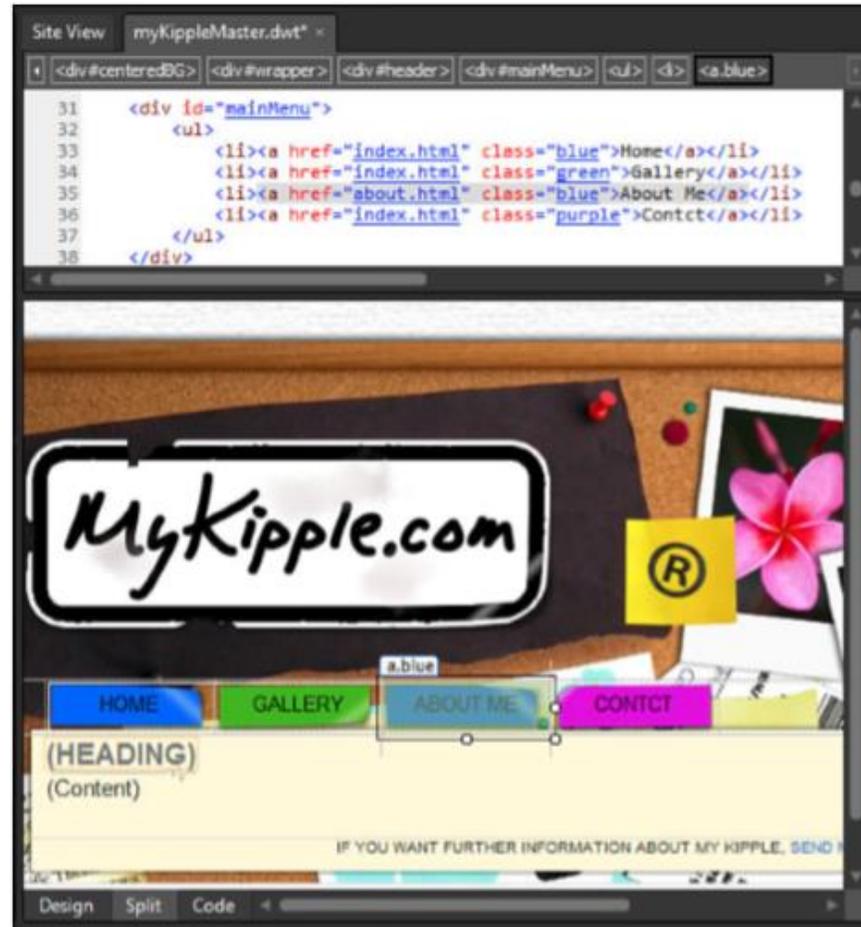


By previewing the new page in your browser, you see that, although all you did was insert the heading and main content, the page looks and works just like the *default.html* page, because all the common components are the same.

Update Your Site by Editing the DWT

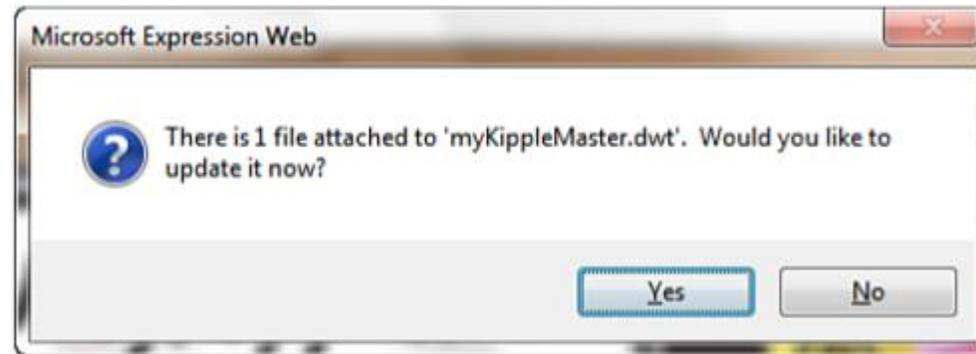
- The reason the editable region's code remains in the page is to give you the ability to change the DWT and by doing so to change all the pages created from it.
- Open the *mykippleMaster.dwt* page in *Split* view.
- Find the unordered *#mainMenu* list in *Code* view.
- Create a new list item before *Contact* to create a new button.
- Type *About Me* as the text, and create a hyperlink that points to the new *about.html* page you just created.
- Like the three other buttons, attach a color class to the new button.

Update Your Site by Editing the DWT Cont.



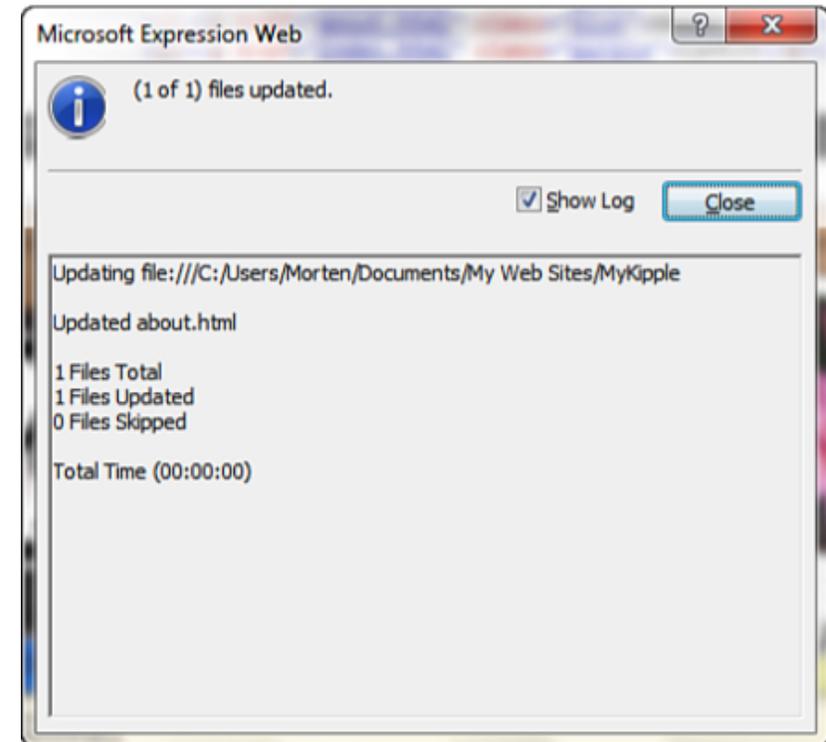
Update Your Site by Editing the DWT Cont.

- Save the *mykippleMaster.dwt* page.
- An information dialog opens to tell you that there is one page attached to *mykippleMaster.dwt* and asks whether you want to update it now.
- Click *Yes*.
- Expression Web 4 now updates all the files built based on the DWT.



Update Your Site by Editing the DWT Cont.

- When it finishes, another information dialog opens to tell you how many pages Expression Web 4 updated.
- If you check the *Show Log* box, the dialog expands to provide a more detailed log of the update process, including a list of what pages the application updated.



Update Your Site by Editing the DWT Cont.

- If you test the *about.html* page in your browser, you see that the main menu now has an *About Me* button in addition to the three original ones, even though you didn't make any changes to the page.
- This example showcases the true power of DWTs.
- By making a website where a DWT is the basis for all the pages, you can make consistent changes to the common content in every page by editing just one page.

Attach a DWT to an Existing Page

- Sometimes, you want to add the new DWT designs to old pages that you already built.
- Your initial hunch might be to copy and paste the contents from this page into a new one, but that is not necessary.
- Just as you attached a DWT to a new page, you can also attach a DWT to an existing page.
- In Hour 9, you created a page called *myWallet.html* that contained a table listing content of my wallet.
- Now, you want this page to match the rest of the pages in the *MyKipple* site.

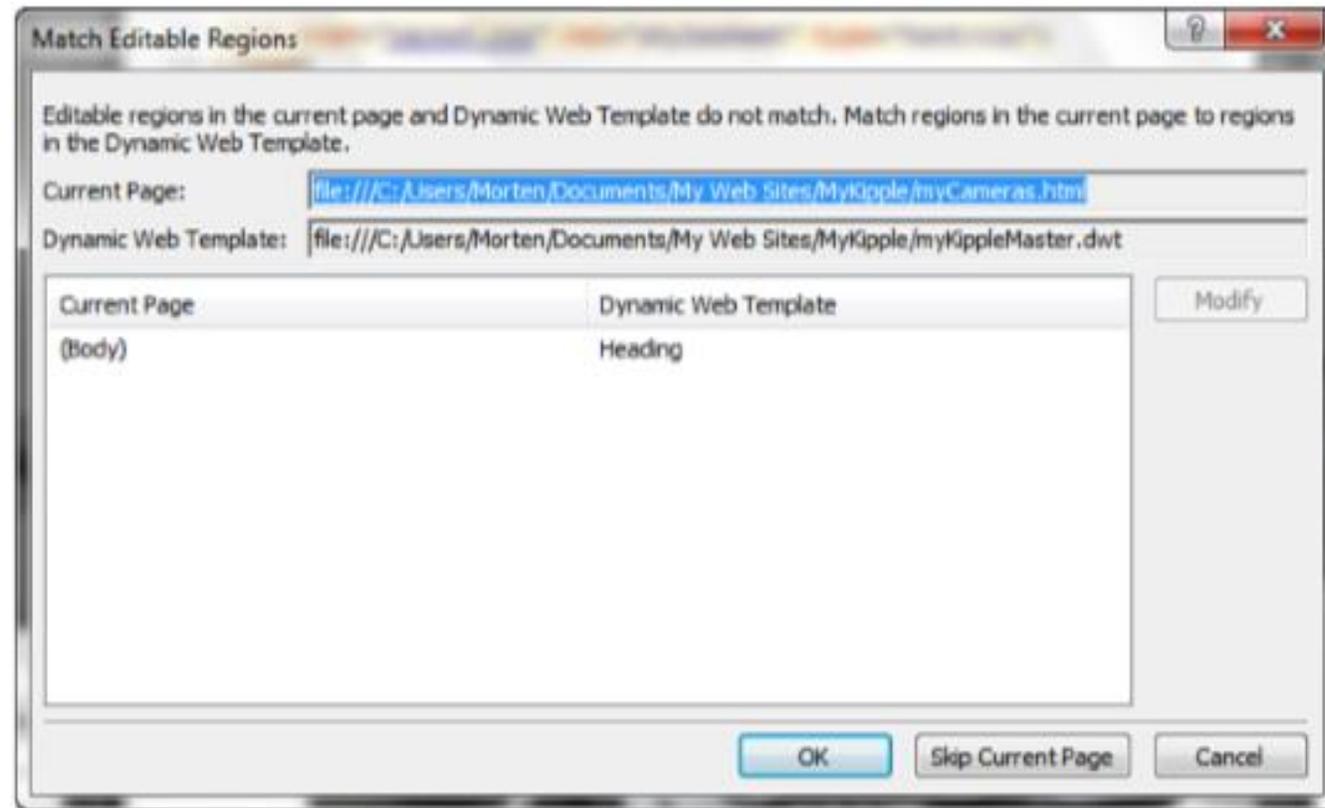
Attach a DWT to an Existing Page Cont.

- Open the *myWallet.html* file in *Design* view.
- From the menu bar, select *Format, Dynamic Web Template, Attach Dynamic Web Template*.
- In the *Attach Dynamic Web Template* dialog, select *mykippleMaster.dwt*.
- A warning appears, telling you that content outside of the `<html>` tag will be deleted and that if you want to preserve it, you need to move it into an editable region or into the `<head>` tag.
- This warning is to alert designers who have attached code, such as a script, outside of the main content of the page.
- If that is the case, you can click *No* and move the script.
- For most situations, this is not necessary, so click *Yes*.

Attach a DWT to an Existing Page Cont.

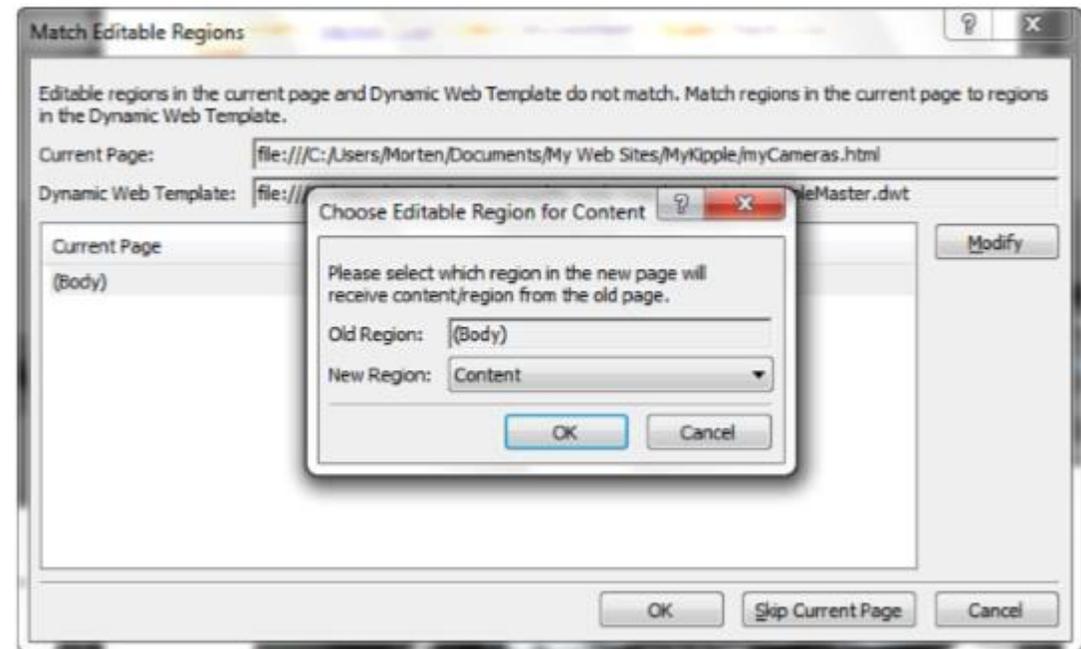
- Because you did not create the *myWallet.html* page using the DWT, there are no editable regions defined within the page.
- Instead of just erasing the content not already defined in the page, Expression Web 4 makes an educated guess as to where you want your content and lets you correct its choices.
- When there are no matching editable regions or there is another conflict, the *Match Editable Regions* dialog opens.
- From here, you can tell Expression Web 4 what content should go in what region.
- In this example, the `<body>` tag contains only one block of content, and Expression Web 4 is guessing that it should go in the heading region.
- To change this, select the list item by clicking it and clicking the *Modify* button.

Attach a DWT to an Existing Page Cont.



Attach a DWT to an Existing Page Cont.

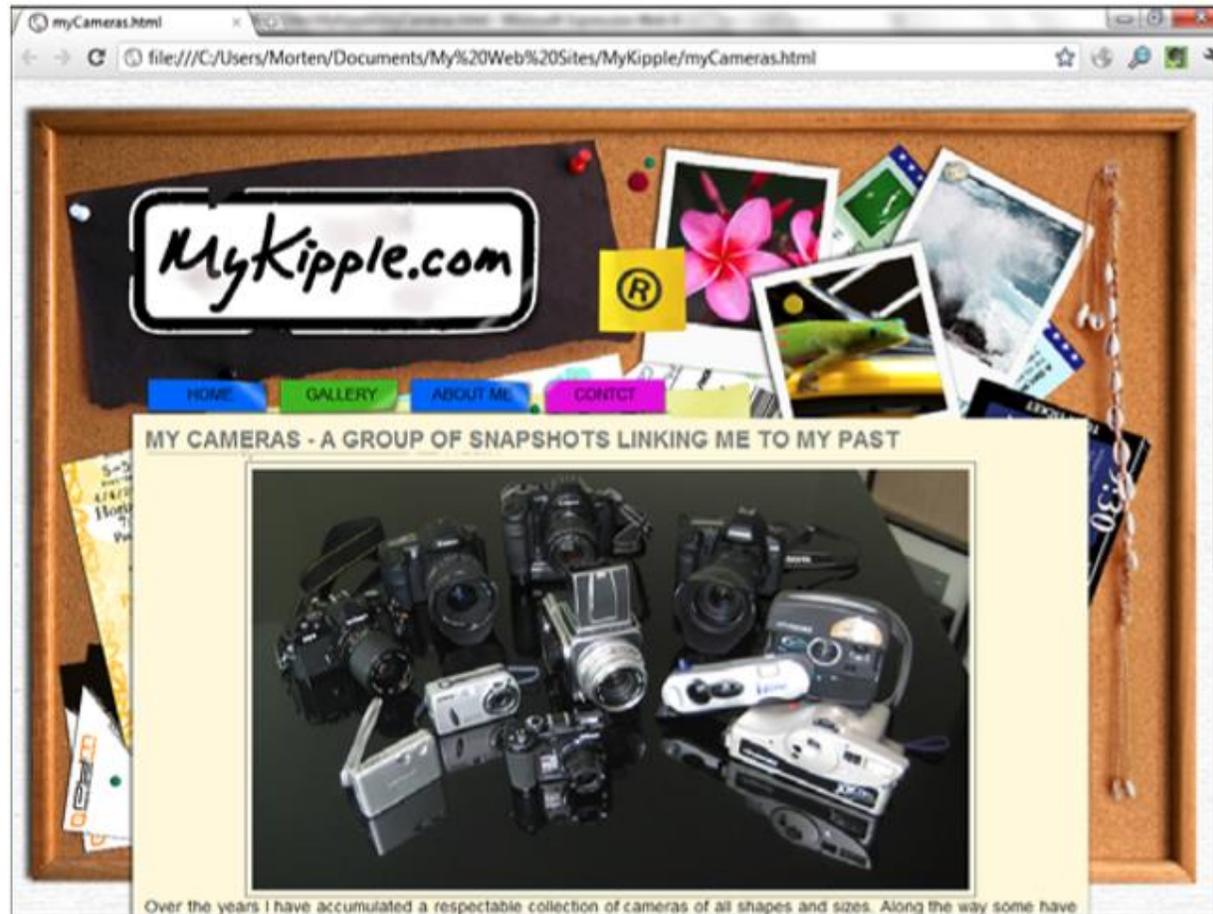
- When you click the *Modify* button, the *Choose Editable Region for Content* dialog appears.
- There, you can define which new region should receive the content from the old region.
- Use the *New Region* drop-down menu to select the content region for the `<body>` content.
- Click *OK* twice to attach the DWT.



Attach a DWT to an Existing Page Cont.

- As you can see, the conversion to the new look is not flawless.
- Among other things, you now have some stray divs and tags (most important, one called *#wrapper0*) that you need to remove, and you need to move the heading up into the heading area and delete the entire *#sidebar* div.
- All of these fixes can be done in *Split* view, and after some tidying up, the page will look the way it should.
- Now, if you make any changes to the DWT, the *myWallet.html* page changes automatically, too.

Attach a DWT to an Existing Page Cont.



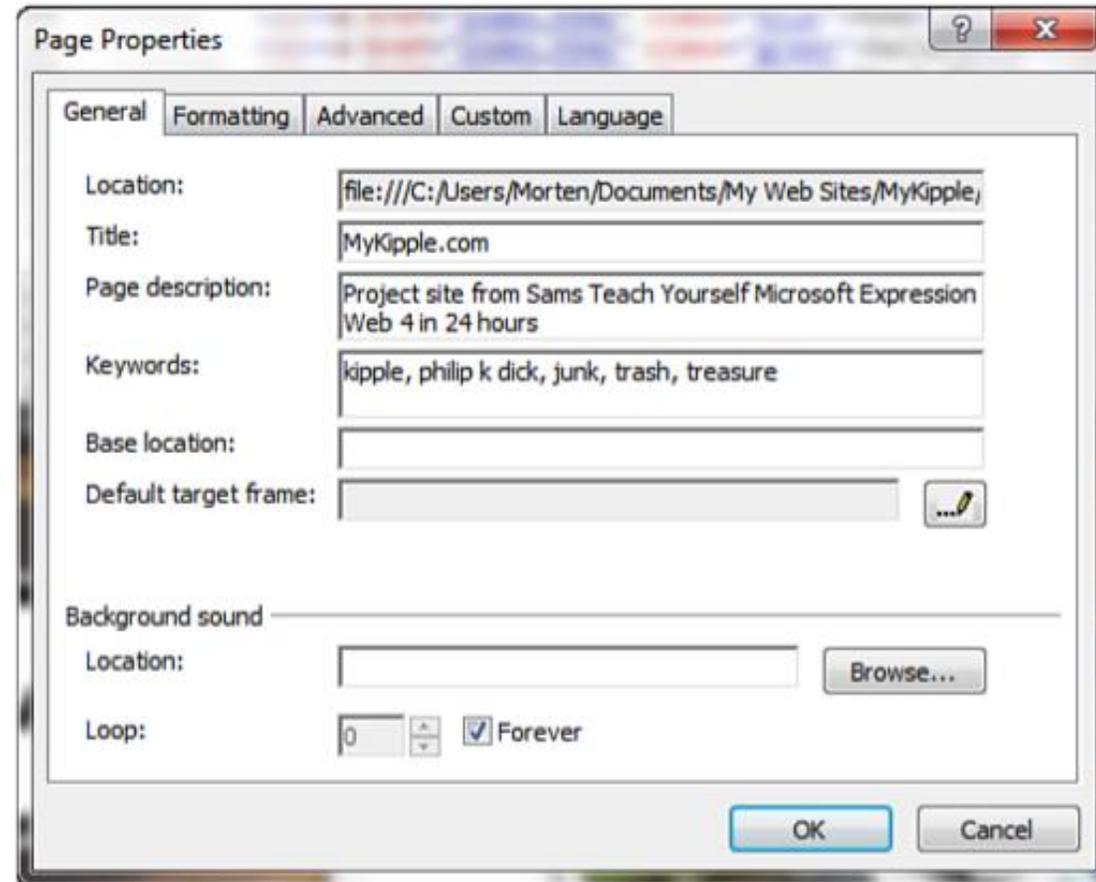
Editable Regions Outside the Body

- As mentioned in the beginning of this hour, Expression Web 4 inserts an editable region called *doctitle* by default in all pages attached to a DWT because the application assumes (as it should) that every page has its own distinctive title.
- Because the `<title>` tag is contained within an editable region by default, you can edit it as you normally would either through the *Page Properties* dialog or directly in *Code* view.
- Using the same technique, you can also predefine other head content (editable or locked) for your pages.
- Moreover, you can use the editable content area in the DWT to give the page predefined common properties that you or other designers can edit, add to, or replace when building new pages.

Create Common Editable Keywords for All Pages

- To make a page easier to find for people who search the Web using search engines, it should always contain keywords and a description that describes the page and its contents.
- With the *mykippleMaster.dwt* page open in *Split* view, right-click in *Design* view and select *Page Properties* from the drop-down menu.
- This opens the *Page Properties* dialog.
- In the *Title* area, enter *MyKipple.com*.
- In the *Page Description* area, enter a short description that is common to all the pages.
- In the *Keywords* area, insert the keywords *kipple, philip k dick, junk, trash, treasure*.
- Click *OK* to apply the changes.

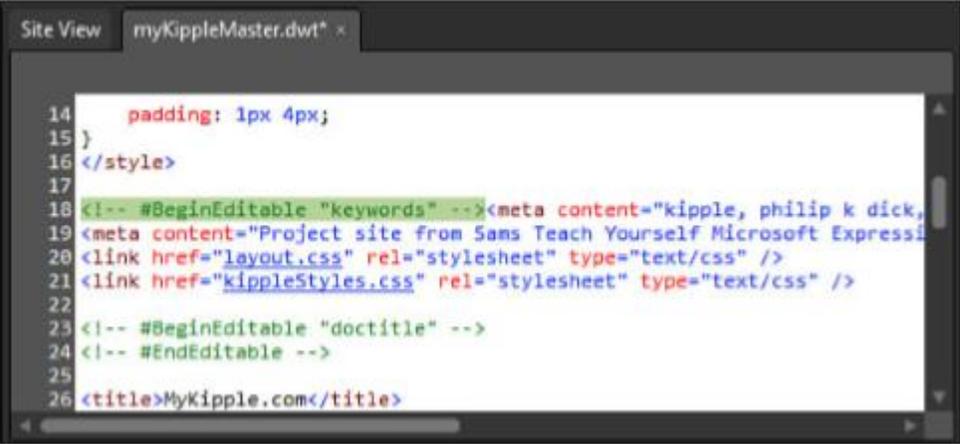
Create Common Editable Keywords for All Pages Cont.



Create Common Editable Keywords for All Pages Cont.

- Toward the top of the page in *Code* view (most likely around line 17) are two new lines of code that start with `<meta content="....`
- These lines define the keywords and description for the page.
- In this example, you want the keywords to be editable, but not the description.
- To do so, place your cursor at the beginning of the first line in *Code* view containing the keywords and manually add the following line of code:

```
<!-- #BeginEditable "keywords" -->
```



```
Site View myKippleMaster.dwt x
14 padding: 1px 4px;
15 }
16 </style>
17
18 <!-- #BeginEditable "keywords" --><meta content="kipple, philip k dick,
19 <meta content="Project site from Sams Teach Yourself Microsoft Express!
20 <link href="layout.css" rel="stylesheet" type="text/css" />
21 <link href="kippleStyles.css" rel="stylesheet" type="text/css" />
22
23 <!-- #BeginEditable "doctitle" -->
24 <!-- #EndEditable -->
25
26 <title>MyKipple.com</title>
```

Create Common Editable Keywords for All Pages Cont.

- Press the *End* key or use the mouse to place the cursor at the end of the line.
- Add the following line of code after the last tag:

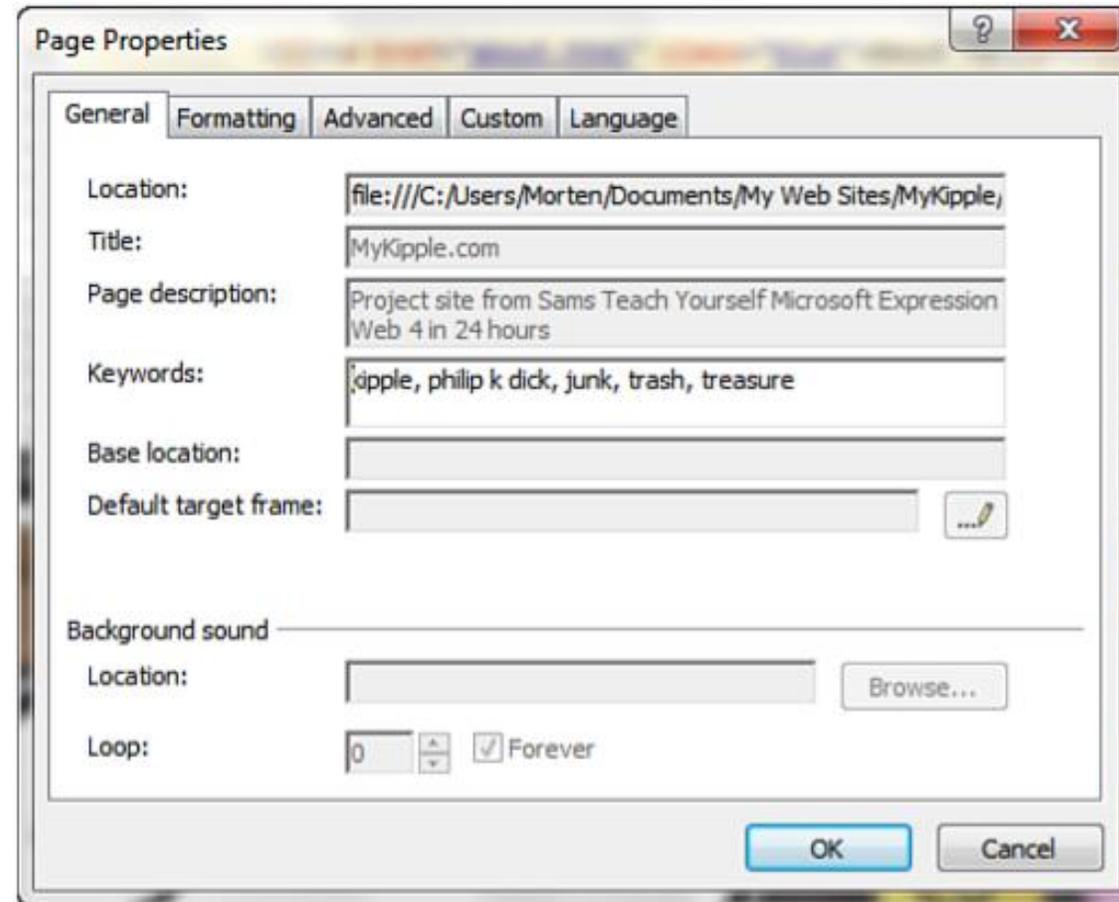
```
<!-- #Editable -->
```

- Save the DWT and click *Yes* when asked whether you want to update the attached pages.

Create Common Editable Keywords for All Pages Cont.

- After applying the changes, open the *about.html* page, right-click, and select *Page Properties* from the context menu.
- In the *Page Properties* dialog, you see that the *Title* and *Keywords* areas are editable, but the *Description* and other areas are grayed out.
- It is also important to note that even though the *Keywords* section remains editable, Expression Web 4 still inserts the words you added in the DWT.
- After the application creates the new page, you can edit, add to, or delete those words without the changes affecting the DWT or the other pages.
- You'll also see that the title is set to *HomeGalleryContact*.
- These are the first words on the site and Expression Web 4 has made the assumption that they are the page title.
- Change the title to *About Me*.

Create Common Editable Keywords for All Pages Cont.

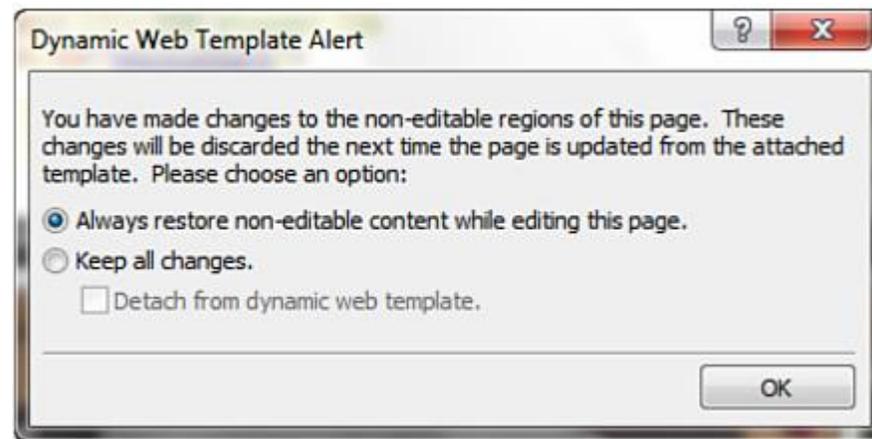


Editing Content Outside the Editable Regions in Individual Pages

- Although it is not recommended, you can manually change the content outside of the editable regions in individual pages.
- While in *Design* view, the only selectable areas are the editable regions.
- However, if you switch to *Code* view, you can modify the code manually.
- In a page created from a DWT, Expression Web 4 highlights all the code defined by the DWT in yellow in *Code* view.
- The highlighting tells you exactly which areas the template controls externally and which areas are open for editing.
- Nevertheless, you can place your cursor anywhere within the code and make all the changes you want.
- However, if you make changes to the highlighted code, Expression Web 4 tells you that you are now changing code defined by the DWT.

Editing Content Outside the Editable Regions in Individual Pages Cont.

- When you save the file or switch back to *Design* view after changing the code outside the editable regions in *Code* view, Expression Web 4 displays an alert dialog with a warning that the changes you have made are in the code defined by the DWT.
- The *Dynamic Web Template Alert* dialog gives you two options: either restore the non-editable content (in other words, discard any changes made to the code defined by the DWT) or keep the changes.
- If you choose the latter, you have the additional option of detaching the page from the DWT.



Editing Content Outside the Editable Regions in Individual Pages Cont.

- The ability to change the code in individual pages comes with a strong warning—if later you choose to make changes to the DWT and update the files attached to it, you permanently erase the changes you made in the individual page code.
- If an area within a page requires individualization for each page, it is always advisable to place this area within an editable region and place the default content in the DWT.
- That way, if you do nothing to the code, it remains the same across every page, but you retain the ability to change the individual pages if you choose to do so.
- The most obvious example of this would be if you wanted to change menu options for some pages, but not others.
- To do so, simply place the menu list items within an editable region called menu, and you now have the ability to change the menu in individual pages if necessary without making those changes sitewide.