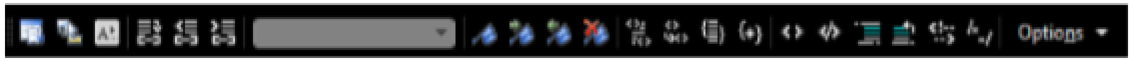
ITEC447
Web Projects

CHAPTER 4 – HTML

Tools in Code View

- As with Design view, Expression Web 4 has an arsenal of tools available to help you work faster and more effectively in Code view.
- These tools can be found on the Code View toolbar, in the Toolbox panel, and inside Code view itself.
- Code view has a dedicated toolbar you can open from the *Toolbar* menu under *View* on the menu bar.
- When opened, it docks under the *Common* toolbar on top of your workspace.



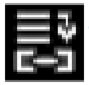
The Buttons of the Code View Toolbar and What They Do

- The *List Members* button tells IntelliSense to provide a shortcut menu for the word or segment your cursor is on. Therefore, if you write <a and click the button, a drop-down menu with all the possible tags starting with the letter a appears.

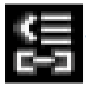
- The *Parameter Info* button opens a list of the valid script parameters for the same segment. (This applies only to script languages such as JavaScript, ASP.NET, and PHP.)

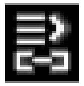
- The *Complete Word* button completes the word you type based on an educated (and surprisingly accurate) guess.

The Buttons of the Code View Toolbar and What They Do Cont.



- The *Follow Code Hyperlink* button lets you navigate to the destination of the selected hyperlink within Expression Web 4.





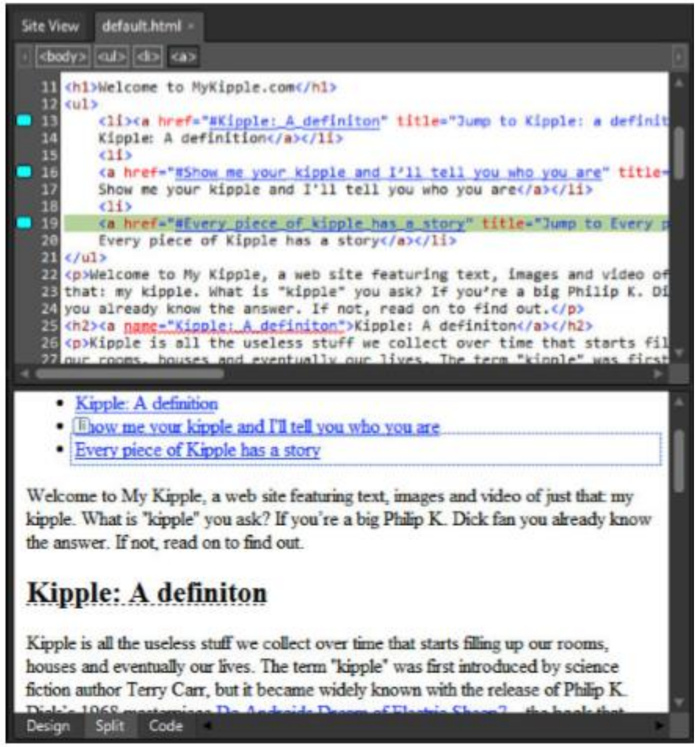
- The *Previous Code Hyperlink* and *Next Code Hyperlink* buttons work in much the same way as the Back and Forward buttons in a browser, taking you back and forth through hyperlinks you have already visited with the Follow Hyperlink button.

- The *Function Lookup* box directs you to the function you select from the drop-down list of the available functions (such as JavaScript or PHP) in your document. This box works only if your document contains functions (small programs) that are present in the code.

The Buttons of the Code View Toolbar and What They Do Cont.

- In Code view, Expression Web 4 lets you insert temporary bookmarks that you can use to quickly navigate between segments of your code.
- The *bookmarks* appear as small blue boxes on the far left side of the panel next to the line numbers.

Code view bookmark



```
<h1>Welcome to MyKipple.com</h1>
<ul>
    <li><a href="#Kipple: A definiton" title="Jump to Kipple: a definit
    Kipple: A definition</a></li>
    <li>
    <a href="#Show me your kipple and I'll tell you who you are" title=
    Show me your kipple and I'll tell you who you are</a></li>
    <li>
    <a href="#Every piece of kipple has a story" title="Jump to Every p
    Every piece of Kipple has a story</a></li>
</ul>
<p>Welcome to My Kipple, a web site featuring text, images and video of
that: my kipple. What is "kipple" you ask? If you're a big Philip K. Di
you already know the answer. If not, read on to find out.</p>
<h2><a name="Kipple: A definiton">Kipple: A definiton</a></h2>
<p>Kipple is all the useless stuff we collect over time that starts fil
our rooms, houses and eventually our lives. The term "kipple" was first
```

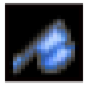
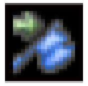
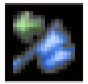
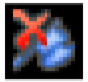
- Kipple: A definition
- Show me your kipple and I'll tell you who you are
- Every piece of Kipple has a story

Welcome to My Kipple, a web site featuring text, images and video of just that: my kipple. What is "kipple" you ask? If you're a big Philip K. Dick fan you already know the answer. If not, read on to find out.

Kipple: A definiton

Kipple is all the useless stuff we collect over time that starts filling up our rooms, houses and eventually our lives. The term "kipple" was first introduced by science fiction author Terry Carr, but it became widely known with the release of Philip K.

The Buttons of the Code View Toolbar and What They Do Cont.

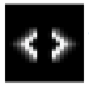





- The *Toggle Bookmark* button inserts a bookmark at the current line of code.
- The *Next Bookmark* and *Previous Bookmark* buttons navigate between the set bookmarks.
- The *Clear Bookmarks* button removes all bookmarks from the document.

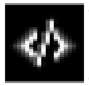
The Buttons of the Code View Toolbar and What They Do Cont.

- Based on the current location of the cursor, the *Select Tag* button selects and highlights the active beginning and end tags and the content between them. This tool is helpful to see what content a certain tag affects.
- The *Find Matching Tag* button finds the beginning tag of the selected content when you first click it and the respective end tag when you click it again. This is an excellent tool if you are confused about where your tags begin or end, and you can use it to clean up code if beginning or end tags are missing.
- The *Select Block* button works the same way as the *Select Tag* button, except it finds and selects code blocks bracketed by braces rather than tags.
- The *Find Matching Brace* button works the same way as the *Find Matching Tag* button, except it looks for braces rather than tags.

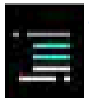
The Buttons of the Code View Toolbar and What They Do Cont.



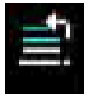
- The *Insert Start Tag* button inserts an empty start tag at the current location.



- The *Insert End Tag* button inserts an empty end tag at the current location.
- You must manually enter the names for these tags.



- The *Comment* and *Uncomment* buttons are used to make it easier to comment out lines or selections. To use the *Comment* option, either place your cursor on the line of code you want to comment out or highlight the section you want to comment out and click the button or use the keyboard shortcut Ctrl+J. This adds the appropriate comment tags before and after the section.



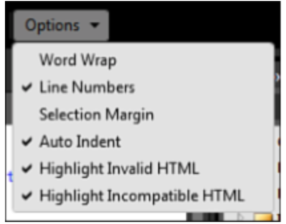
- To use the *Uncomment* option, place your cursor anywhere inside a section that is commented and either click the Uncomment button or use the keyboard shortcut Ctrl+Shift+J. This seemingly simple addition to the Code View toolbar was frequently requested by power users such as myself and can be a huge timesaver, especially if you keep the keyboard shortcuts handy.

The Buttons of the Code View Toolbar and What They Do Cont.

- <!-- --> The *Insert HTML Comment* button inserts the beginning and end tags for nonfunctional HTML code comments. These comments can be inserted anywhere inside the HTML code. A code comment is text that is visible only in the code itself and has no function other than as a descriptive tool for the developer or anyone else looking at the code. Designers commonly use comments to keep track of changes or explain how or why certain parts of code work.
- /* */ The *Insert CSS Comment* button inserts the beginning and end tags for nonfunctional CSS code comments. These comments can be inserted anywhere inside the CSS code, typically in a style sheet.

The Buttons of the Code View Toolbar and What They Do Cont.

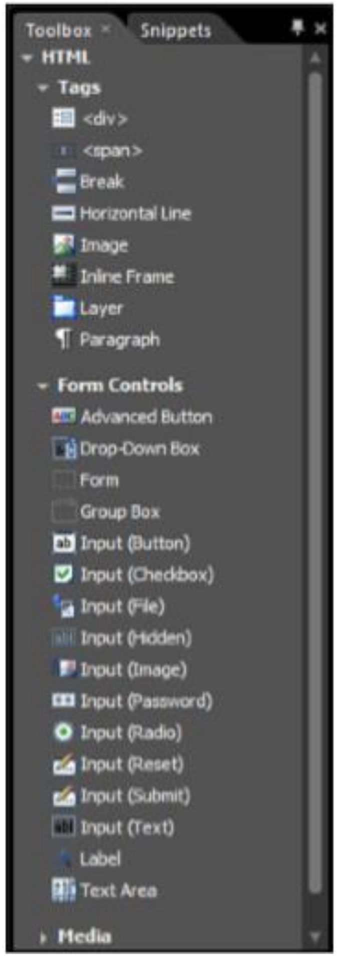
- The *Options* button gives you direct access to the code-viewing options that define how code is displayed in Code view.
- You can also access these options by opening the *Page Editor* Options from the *Tools* menu and selecting the *General* tab.
- The options are *Word Wrap*, which wraps the lines of code to fit inside the window (removing the need to scroll sideways to read all the code);
- *Line Numbers* (self-explanatory);
- *Selection Margin*, which adds a space between the line numbers and the content so that you can click and drag in the margin with the mouse to select multiple lines of code;
- *Auto Indent*, which keeps the indentation from the current line when a new line is created by pressing Enter;
- *Highlight Invalid HTML*, which displays invalid HTML code as red font with a yellow background; and
- *Highlight Incompatible HTML*, which displays incompatible HTML with a wavy underline like the one commonly used to highlight typos.



Word Wrap
✓ Line Numbers
Selection Margin
✓ Auto Indent
✓ Highlight Invalid HTML
✓ Highlight Incompatible HTML

Common HTML Tags in the Toolbox

- To help in the coding process, Expression Web 4 provides a source of commonly used code through the *Toolbox* panel.
- Under HTML, you find commonly used tags, form controls, and media-insertion tools.
- To use any of these, simply place your cursor where you want to insert the code in *Code view* and double-click the desired element.
- You can also drag and drop the elements directly into the code from the panel.
- If your Toolbox panel is closed, you can access the same commonly used code elements along with all other available elements under *Insert* on the menu bar.



Using Code View as a Learning Tool

- Web pages are a bit unusual in that after they are published, the code (or blueprint) is available for all to see.
- That means if you find something you like on a website, you can look at the code to see what is going on and learn from it.
- In Expression Web 4, you can even open websites from the Internet without downloading them first.
- All you have to do is click *Open* from the *File* menu and insert the URL for the site you want to take a closer look at.
- When it is open in *Split* view, you have full access to all the code and other elements that make up the page, and you can see how everything fits together.

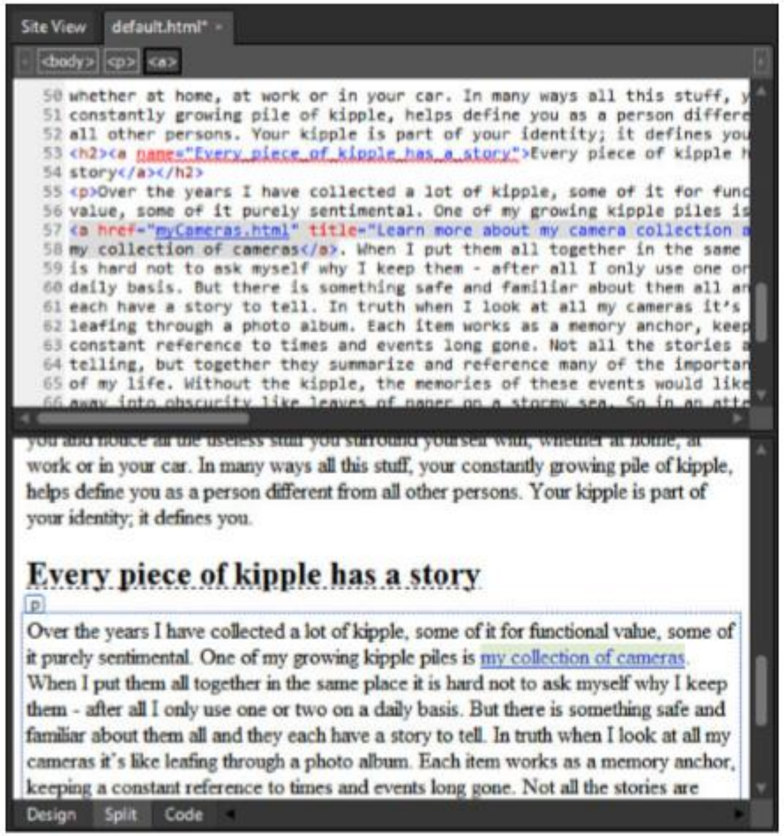
Using Code View as a Learning Tool Cont.

- To start, open the *default.html* file in *Split* view.

Dissecting Hyperlinks in Code View

- In Hour 5, you added several hyperlinks to the page.
- Now you can use *Code* view to see how they work.
- Click the hyperlink with the words “*all the kipple on my desk*.”
- This highlights the same text in *Code* view.
- When you click the highlighted text in *Code* view and click the *Select Tag* button on the *Code* View toolbar, all the code within this particular tag is highlighted.
- In this case, that is the hyperlink or <a> tag:
- *<a href="myDesk.html" title="Learn more">all the kipple on my desk</a>*

Dissecting Hyperlinks in Code View Cont.



```
whether at home, at work or in your car. In many ways all this stuff, y
constantly growing pile of kipple, helps define you as a person differe
all other persons. Your kipple is part of your identity; it defines you
<h2><a name="Every piece of kipple has a story">Every piece of kipple h
story</a></h2>
<p>Over the years I have collected a lot of kipple, some of it for func
value, some of it purely sentimental. One of my growing kipple piles is
<a href="myCameras.html" title="Learn more about my camera collection a
my collection of cameras</a>. When I put them all together in the same
is hard not to ask myself why I keep them - after all I only use one or
daily basis. But there is something safe and familiar about them all an
each have a story to tell. In truth when I look at all my cameras it's
leafing through a photo album. Each item works as a memory anchor, keep
constant reference to times and events long gone. Not all the stories a
telling, but together they summarize and reference many of the importan
of my life. Without the kipple, the memories of these events would like
away into obscurity like leaves of paper on a stormy sea. So in an atte
```

you and notice all the useless stuff you surround yourself with, whether at home, at work or in your car. In many ways all this stuff, your constantly growing pile of kipple, helps define you as a person different from all other persons. Your kipple is part of your identity; it defines you.

Every piece of kipple has a story

Over the years I have collected a lot of kipple, some of it for functional value, some of it purely sentimental. One of my growing kipple piles is my collection of cameras. When I put them all together in the same place it is hard not to ask myself why I keep them - after all I only use one or two on a daily basis. But there is something safe and familiar about them all and they each have a story to tell. In truth when I look at all my cameras it's like leafing through a photo album. Each item works as a memory anchor, keeping a constant reference to times and events long gone. Not all the stories are

Highlighted code

Selected hyperlink

Dissecting Hyperlinks in Code View Cont.

- The highlighted code shows the basic elements of a hyperlink:
- The visible hyperlink text is contained between the *<a>* and *</a>* tags.
- The brackets of the first wrapper contain all the attributes relating to this specific link, separated by spaces.
- For example;
- *<a attributeOne=”attribute value” attributeTwo=”attribute value” attributeThree=”attribute value”>.*
- The destination address is set using the *href* attribute.
- The *ScreenTip* is set using the *title* attribute.

Dissecting Hyperlinks in Code View Cont.

- If you look at the code for the link to the book on *Amazon.com* you created earlier in the page, you see that an added attribute, named target, is set to _*blank*:
- *<a href=”http://www.amazon.com/Do-Androids-Dream-Electric-Sheep/(...)” target=”_blank” title=”Do Androids Dream of Electric Sheep? is available from Amazon.com”>Do Androids Dream of Electric Sheep?</a>*
- This tells the browser that the target of the link (that is, what window it should open in) is a new, empty (blank) window.
- If no target is set, the link opens in the current window. You can also force the target to be the current window by setting it to _*self*.

Dissecting Hyperlinks in Code View Cont.

- At the bottom of the page, you inserted a hyperlink to open the default email program on the computer:
- *<a href=”mailto:kippleinfo@pinkandyellow.com?subject=Email generated from the MyKipple website” title=”Send all your questions about this site here”>send me an email</a>*
- The *href* attribute in this hyperlink contains a code rather than an address.
- This code has two sections:
- *mailto:* tells the browser that the following is a recipient email address for insertion into the default email program.
- *?subject=* tells the browser that the following text is the subject line to insert into the email.

Dissecting Hyperlinks in Code View Cont.

- You also created several bookmarks within default.html.
- They are the headings with the dotted underlines in *Design* view.
- If you click one of these headings and look at them in *Code* view, you see that the *<a>* tag wraps them, too.
- However, unlike the hyperlinks, they don’t have an *href* attribute.
- Instead, they have a name attribute that matches the bookmark name you set in the *Bookmark Properties* dialog:
- *<h2><a name=”Kipple:_A_definition”>Kipple: A definition</a></h2>*

Dissecting Hyperlinks in Code View Cont.

- Hyperlink code can wrap around many elements, not just text.
- In the *myDesk.html* file, you created a hyperlink attached to an image.
- To see an example of this, open *myDesk.html* in *Split* view and click the thumbnail of the detail photo you created using *Auto Thumbnail*.
- By highlighting the relevant code using the *Select Tag* function in *Code* view, you can see that the only difference between this hyperlink and the previous examples is that an image has replaced the content between the <*a*> and <*/a*> tags:
- *<a href="Images/KennyOriginal.jpg"> <img alt="Kenny squeeze toy" class="style2" src="Images/KennyOriginal_small.jpg" style="float: left" height="147" width="100"> <!--MSComment="autothumbnail" xthumbnail-orig-image="file:///C:/Documents and Settings/Morten/My Documents/My Web Sites/myKipple/Images/KennyOriginal.jpg" --> </a>*

Dissecting Hyperlinks in Code View Cont.

- Note the commented section, which is shown in gray in *Code* view.
- Expression Web 4 inserted this nonfunctional code to explain the origins of the thumbnail.
- The comment is wrapped inside a special tag that starts with <*!*-- and ends with -->.
- *<!-- MSComment="autothumbnail" xthumbnail-orig-image="file:///C:/Documents and Settings/Morten/My Documents/My Web Sites/myKipple/Images/KennyOriginal.jpg" -->*

Images in Code View

- When you inserted images into the page, you didn’t convert them into code first.
- And when you click them and inspect them in *Code* view, all you get is what looks like a hyperlink.
- Images in web pages belong to a group of elements known as replaced elements.
- A replaced element is a segment of code that has no content in it, but instead points to external content that the browser displays in its place.
- So when you insert an image into your web page in *Design* view, you are inserting a small piece of code that tells the browser to find a specific image file, insert it in place of the code, and apply a set of attributes to it.

Images in Code View Cont.

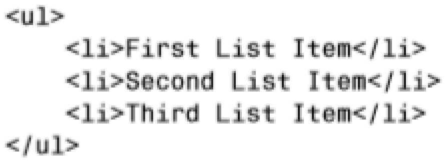
- Select the Kenny detail thumbnail in *Design* view and look at the section of code highlighted in *Code* view:
- *<img alt="Kenny squeeze toy" class="style2" height="147" src="Images/KennyOriginal_small.jpg" style="float: left" width="100" />*
- Unlike the tags you saw earlier, the *<img>* tag doesn't have a corresponding *</img>* tag but rather ends with */>*, telling the browser that it is closed.
- Within the tag, you find all the information you inserted using the *Picture Properties* dialog:
- The *alt* attribute defines the alternative text; the *height* and *width* attributes define the height and width; and the *src* attribute defines the location of the external image file.

Unordered and Ordered Lists in Code View

- Working with lists in *Design* view can get complicated, especially if you have lists within lists.
- Fortunately, lists have an organized code structure that makes *Code* view ideal for advanced list editing.
- There are two types of lists.
- Unordered lists, signified by the *<ul>* tag, are what we usually refer to as bullet lists. When the browser finds a *<ul>* tag, it assumes that each list item should have a bullet in front of it unless otherwise specified.
- Ordered lists, signified by the *<ol>* tag, are what we usually refer to as numbered lists. When the browser finds an *<ol>* tag, it assumes that each list item should have a consecutively higher number in front of it than the previous one.

Unordered and Ordered Lists in Code View Cont.

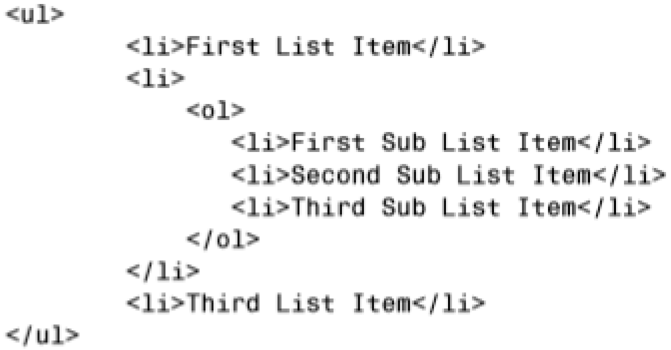
- The HTML list structure, regardless of whether it is an unordered or an ordered list, is simple.
- It starts with the *<ul>* or *<ol>* tag, telling the browser what kind of list it is, and then wraps each list item between *<li>* and *</li>* tags.
- The end of the list is indicated by the closing *</ul>* or *</ol>* tag.
- The structure looks like this:



```
<ul>
    <li>First List Item</li>
    <li>Second List Item</li>
    <li>Third List Item</li>
</ul>
```

Unordered and Ordered Lists in Code View Cont.

- To change this list from an unordered list to an ordered list, all you need to do is change the *<ul>* and *</ul>* tags to *<ol>* and *</ol>*.
- You can build lists within lists just as easily.
- The following is an unordered list with three list items in which the second one is an ordered list with three subitems:



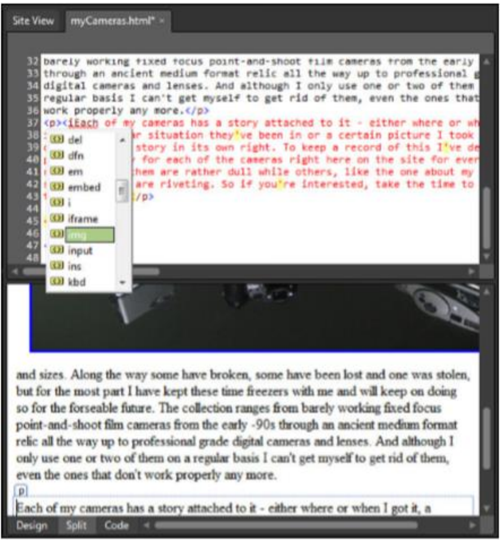
```
<ul>
        <li>First List Item</li>
        <li>
            <ol>
               <li>First Sub List Item</li>
               <li>Second Sub List Item</li>
               <li>Third Sub List Item</li>
            </ol>
        </li>
        <li>Third List Item</li>
</ul>
```

Introducing IntelliSense—Your New Best Friend

- Joining the *Code* view toolbar and the *Toolbox* panel on the list of tools to help you navigate and write code in Expression Web 4 is *IntelliSense*.
- With the addition of HTML5, CSS3, JavaScript, and even jQuery script support, it is now, one of the most powerful IntelliSense offering available.
- *IntelliSense* is a tool that helps you write proper code faster. The best way to understand exactly what it does is to see it in action.

Introducing IntelliSense—Your New Best Friend Cont.

- In this example, you insert an image in *Code* view with the help of *IntelliSense*:
- With the *myDesk.html* file open in *Split* view, find the second paragraph wrapped by *<p>* tags. Place the cursor next to the *<p>* tag before the text.
- Still in *Code* view, type a “less-than” bracket: <. A drop-down menu appears below the bracket—this is *IntelliSense* presenting you with all the available code snippets that start with a less-than bracket. You are inserting an image, so you want the *img* tag. To find this, enter an *i* and *IntelliSense* brings up all the available tags that start with *<i*. Use the arrow keys to skip down to *<img* and press Enter to insert the *<img* tag.



```
barely working fixed focus point-and-shoot film cameras from the early
through an ancient medium format relic all the way up to professional g
digital cameras and lenses. And although I only use one or two of them
regular basis I can't get myself to get rid of them, even the ones that
work properly any more.</p>
<p><iEach of my cameras has a story attached to it - either where or wh
            r situation they've been in or a certain picture I took
            story in its own right. To keep a record of this I've de
            for each of the cameras right here on the site for ever
            hem are rather dull while others, like the one about my
            are riveting. So if you're interested, take the time to
            /p>
```

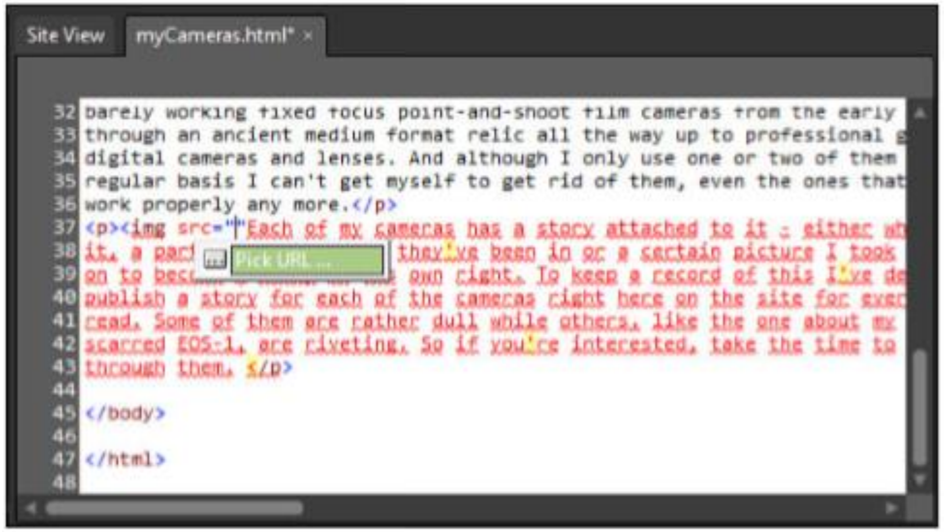
and sizes. Along the way some have broken, some have been lost and one was stolen, but for the most part I have kept these time freezers with me and will keep on doing so for the forseable future. The collection ranges from barely working fixed focus point-and-shoot film cameras from the early -90s through an ancient medium format relic all the way up to professional grade digital cameras and lenses. And although I only use one or two of them on a regular basis I can't get myself to get rid of them, even the ones that don't work properly any more.

p

Each of my cameras has a story attached to it - either where or when I got it, a

Introducing IntelliSense—Your New Best Friend Cont.

- You need a source for your image. Press the *spacebar* and *IntelliSense* pops up with new options. This time, it gives you only available attributes for the *img* tag. As you previously learned, the image location is set with the *src* attribute, so type *s* to narrow the search. *IntelliSense* takes you straight to *src*, and all you need to do is press the Enter key.
- Because the *src* attribute requires a URL (uniform resource locator), *IntelliSense* inserts the necessary code and asks you for the location of the image. Click the *Pick URL* button, and navigate to the desired file (in this case, KennyOriginal_small.jpg, located under the Images folder).



```
barely working fixed focus point-and-shoot film cameras from the early
through an ancient medium format relic all the way up to professional g
digital cameras and lenses. And although I only use one or two of them
regular basis I can't get myself to get rid of them, even the ones that
work properly any more.</p>
<p><img src=""Each of my cameras has a story attached to it - either wh
it, a part                     they've been in or a certain picture I took
on to bec                      own right. To keep a record of this I've de
publish a story for each of the cameras right here on the site for ever
read. Some of them are rather dull while others, like the one about my
scarred EOS-1, are riveting. So if you're interested, take the time to
through them. </p>

</body>

</html>
```

Introducing IntelliSense—Your New Best Friend Cont.

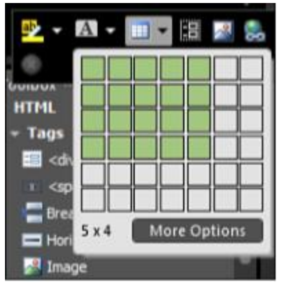
- Before you close the tag, remember to add the alternative text using the alt attribute. This is done by following the same steps you used previously, but starting with the letter *a*. If you want to, you can add more attributes to your image using IntelliSense.
- To close the *img* tag, simply type */>*.
- The final code looks like this:
- *<img src="Images/KennyOriginal_small.jpg" alt="Another thumbnail" />*

Introducing IntelliSense—Your New Best Friend Cont.

- If you click outside the active text or accidentally press *Backspace*, *IntelliSense* closes.
- To open it again, use the *Ctrl+L* shortcut.
- *IntelliSense* helps you write a number of different code languages, including HTML, CSS, PHP, JavaScript, ASP.NET, and even Visual Basic.
- More important, it keeps tabs on your code to prevent you from making errors. For instance, if you insert a beginning tag, *IntelliSense* automatically inserts a matching end tag. The same goes for braces and quotes. The code output of *IntelliSense* matches the code settings for the current document.
- The status bar at the bottom of the work area tells you what code language and version you are working in.
- If you want to change the *IntelliSense* settings, open the *Page Editor Options* under *Tools* on the menu bar and select the *IntelliSense* tab.

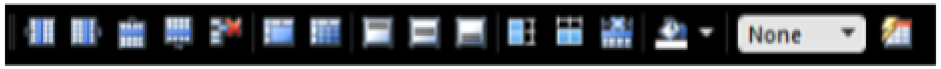
Creating a Table from Scratch

- The following steps show you how to create a table in *Design* view:
- Create and html file named *myWallet.html.*
- In Design view, open the *myWallet.html.*
- Make a new *h1* subheading with the text ’’*The Kipple in My Wallet*’’.
- Make a new paragraph under the heading and click the *Insert Table* button on the *Common* toolbar.
- This opens a drop-down panel consisting of empty boxes.
- By sliding your mouse over the boxes, you can highlight the number of rows and columns you want.
- In this example, you want 4 rows and 3 columns.
- Because the drop-down panel allows only for tables up to 6 rows and 7 columns, you need to click the *More Options* button to open the *Insert Table* dialog and set the rows and columns manually.



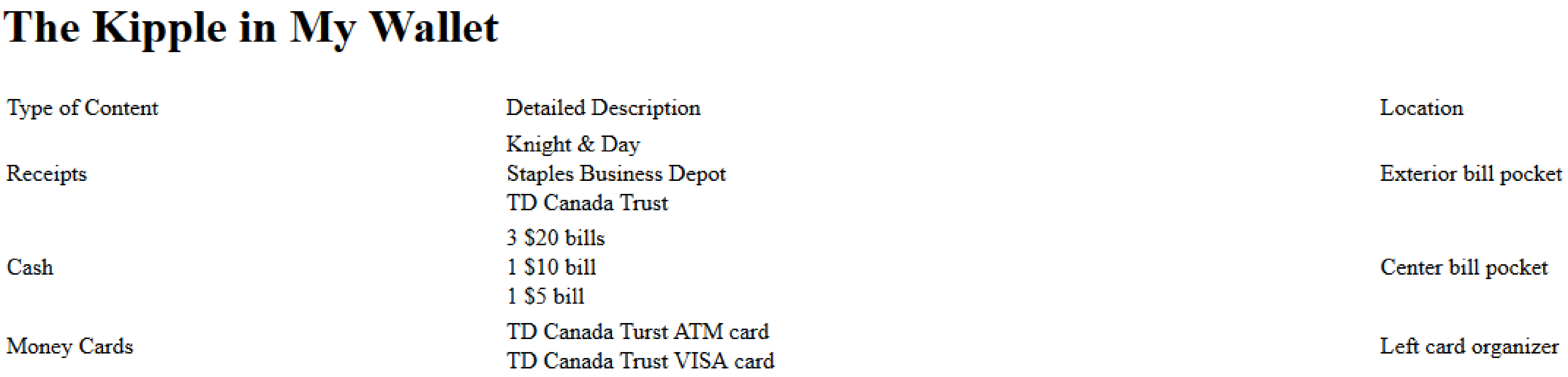
Creating a Table from Scratch Cont.

- You now have a 4-by-3 table in your page.
- By default, HTML tables are invisible, meaning they have no color and no borders.
- They simply separate the content within them, as indicated by the dotted lines.
- You can control how tables appear in your page by using the functions found under *Table* on the menu bar or by right-clicking the table and accessing the options from there.
- However, the easiest way to edit your tables is by using the *Tables* toolbar found under *View*, *Toolbars.*



Creating a Table from Scratch Cont.

- Before going any further, the table needs some content.
- Click inside the first cell and start typing.

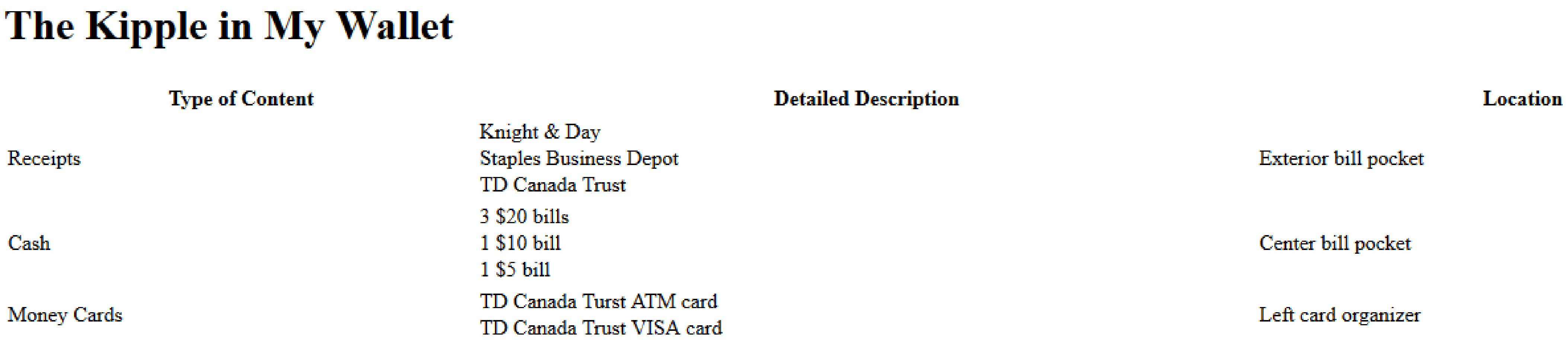


The Kipple in My Wallet

Type of Content	Detailed Description	Location
Receipts	Knight & Day Staples Business Depot TD Canada Trust	Exterior bill pocket
Cash	3 $20 bills 1 $10 bill 1 $5 bill	Center bill pocket
Money Cards	TD Canada Turst ATM card TD Canada Trust VISA card	Left card organizer

Creating Table Headings

- The first row of your table contains the headings that explain the content below.
- The markup should reflect this by using the *<th>* (table heading) tags for the cells instead of the regular *<td>* (table data) tags.
- You can do this either manually in *Code View* or by using the *Quick Tag Editor*.
- When the table heading tag is applied, the text appears centered in the cell and the font is bold.

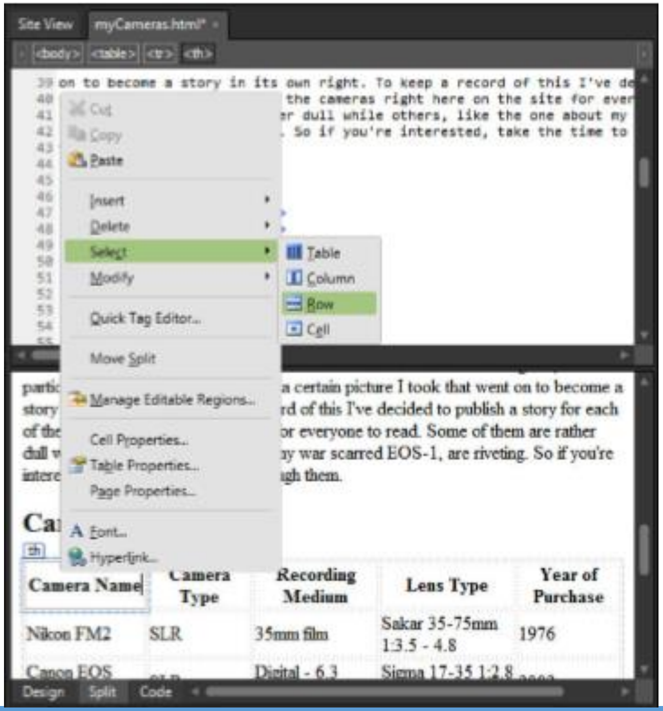


The Kipple in My Wallet

Type of Content	Detailed Description	Location
Receipts	Knight & Day Staples Business Depot TD Canada Trust	Exterior bill pocket
Cash	3 $20 bills 1 $10 bill 1 $5 bill	Center bill pocket
Money Cards	TD Canada Turst ATM card TD Canada Trust VISA card	Left card organizer

Creating Table Headings Cont.

- If you don’t like this appearance and want to change the heading row, you first need to select all the cells in the row.
- You can do this in two ways: Click and hold inside the first cell and drag your mouse over the others. When the mouse pointer is out of the first cell, it changes to a + sign indicating that you are selecting more than one cell.
- You can use this technique to select groups of cells from rows, columns, or both.
- Alternatively, you can right-click any of the cells in the row and click *Select*, *Row.*



```
on to become a story in its own right. To keep a record of this I've de
                                 the cameras right here on the site for ever
                                 r dull while others, like the one about my
                                 . So if you're interested, take the time to
```

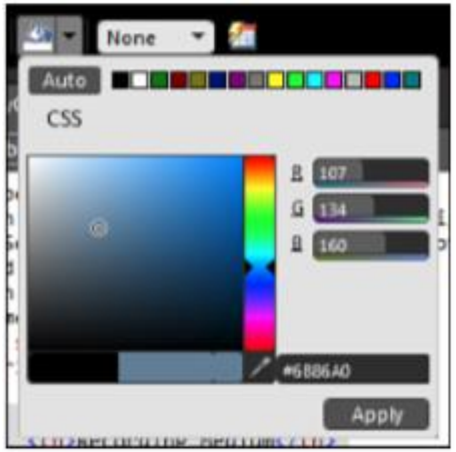
partic a certain picture I took that went on to become a story rd of this I've decided to publish a story for each of the or everyone to read. Some of them are rather dull w ny war scarred EOS-1, are riveting. So if you're intere gh them.

Ca

Camera Name	Camera Type	Recording Medium	Lens Type	Year of Purchase
Nikon FM2	SLR	35mm film	Sakar 35-75mm 1:3.5 - 4.8	1976
Canon EOS	SLR	Digital - 6.3	Sigma 17-35 1:2.8	2003

Creating Table Headings Cont.

- With the row selected, you can change the appearance of the entire row of table headings.
- For example, you can change the background color of the cells by clicking the *Fill Color* button on the *Tables* toolbar and selecting a new color.



R 107
G 134
B 160
#6B86A0

Changing the Vertical Alignment of Cell Content

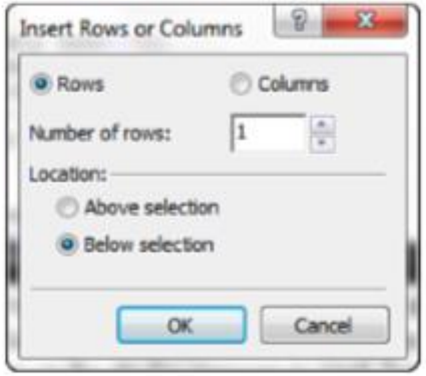
- By default, all the content in a table centers vertically in the cells.
- This looks strange when some cells have one line of text and others have several.
- To change the vertical alignment of the cells, select the remaining cells using the click-and-hold technique and click the *Align Top* button on the *Tables* toolbar.
- You can choose to align text to the *top*, *center*, or *bottom* of the cells.

Adding Rows and Columns to the Table

- When you create a table, you often find that you need to add more rows or columns.
- If you want to add one or two, the quickest way is to use the *Tables* toolbar.
- To add a new row below the last row of your table, select any of the bottom cells and click the *Row Below* button on the *Tables* toolbar.
- You can also add a row above the selected row by clicking the *Row Above* button.
- To add a column, you click either the *Column to the Left* or *Column to the Right* button.

Adding Rows and Columns to the Table Cont.

- If you want to add several rows or columns, select the cell in which you want to create new rows or columns next to and choose *Table*, *Insert*, *Rows* and *Columns* from the menu bar, or right-click and select *Insert*, *Rows* and *Columns* from the context menu.
- This opens the *Insert Rows or Columns* dialog, where you can set how many rows or columns you want and whether to insert them above, below, to the left, or to the right of your selection.



Insert Rows or Columns

● Rows ○ Columns

Number of rows: 1

Location:

○ Above selection

● Below selection

Merging and Splitting Cells

- Sometimes it is necessary to merge cells to display certain types of content within the table.
- Let’s say you want to insert a subheading in the table to go underneath the heading and across the whole table.
- If you simply insert a new row underneath the heading, you still have three columns to work with when you need only one.
- To insert a subheading in your table, first insert a new row underneath the heading using the *Row Below* button.
- Note that if you chose a fill color for the heading row, the styling (color) is automatically copied down to the new row.
- After inserting the new row, select all the cells in that row and click the *Merge Cells* button on the *Tables* toolbar.
- Doing so merges the three cells into one. Now, you can insert a subheading that goes across the entire table.
- Please create a new row under the heading of the table and enter the text ‘’*This list is comprised of the contents of my wallet on Saturday May 24th, 2008*’’.

Merging and Splitting Cells Cont.

- Likewise, you can split a cell into several new rows or columns using the *Split Cell* button.
- After you select a cell and click the button, Expression Web 4 asks how many rows or columns you want to create within that cell.
- You can also merge and split cells by highlighting them, right-clicking, and selecting *Modify* and *Merge Cells* or *Split Cells* from the pop-up menu.

Deleting Cells, Rows, and Columns from the Table

- To delete a cell, right-click the cell and select *Delete*, *Cell* from the context menu or click the *Delete Cells* button on the *Tables* toolbar.
- Both actions delete the cell in question, and the remaining cells shift to the left to accommodate the deletion.
- Deleting a cell leaves an empty space filled with the background color.
- To delete a row or column, right-click inside a cell on the row or column you want to delete and then select *Delete*, *Row* or *Column* or select all the cells in the row or column you want to delete and click the *Delete Cells* button on the *Tables* toolbar.
- Expression Web 4 deletes the row or column, and the remaining rows or columns realign themselves to fill the void.
- To delete an entire table, you can right-click anywhere inside the table and select *Delete Table*; Expression Web 4 deletes the table.

Changing Row and Column Sizes

- When you first entered text into the cells in your table, you noticed that the width of the columns changed accordingly.
- The same thing happened with the height of the rows when you entered several lines of text in one of the cells.
- This is because both row height and column width are set to *AutoFit* by default.
- In most cases, you want to manually control the height and width of your rows and columns to achieve a certain look.
- You can do this in several ways.

Changing Row and Column Sizes Cont.

- The first method uses the *Distribute Rows Evenly* and *Distribute Columns Evenly* buttons on the *Tables* toolbar to make changes to the whole table.
- To use one of these buttons, simply place your cursor anywhere inside the table and click the button.
- The program gives the rows or columns an even distribution based on their content and gives them a fixed height or width.
- To revert to AutoFit, simply click the *AutoFit to Contents* button.

Changing Row and Column Sizes Cont.

- The second method is a more direct approach in which you manually set the width and height of each row or column by grabbing the margins of the table cells by clicking and holding them with the mouse and pulling them up, down, left, or right.
- When you let go, the width and height are permanent just like in the first method, but they are now set to your specifications rather than evenly distributed.
- You can also use the second method to resize the entire table.
- By default, the table width setting is 100% so that the table fills whatever area it is in (no area is yet defined, so the table fills the entire width of the screen), but you can change this by grabbing the sides of the table and resizing it.

Using Table Properties to Change the Appearance of Your Table

- If you want to make changes to the size or appearance of the entire table, you need to open *Table Properties* by right-clicking anywhere inside the table and selecting *Table Properties* or selecting *Table*, *Table Properties* on the menu bar.
- From here, you can change all the main aspects of your table.
- The *Size* setting lets you define how many rows and columns the table should have. The *Layout* setting sets the alignment and float of the entire table in relation to the page, the width and height of the table in *pixels* or *percentages*, and the *cell padding* and *cell spacing*.

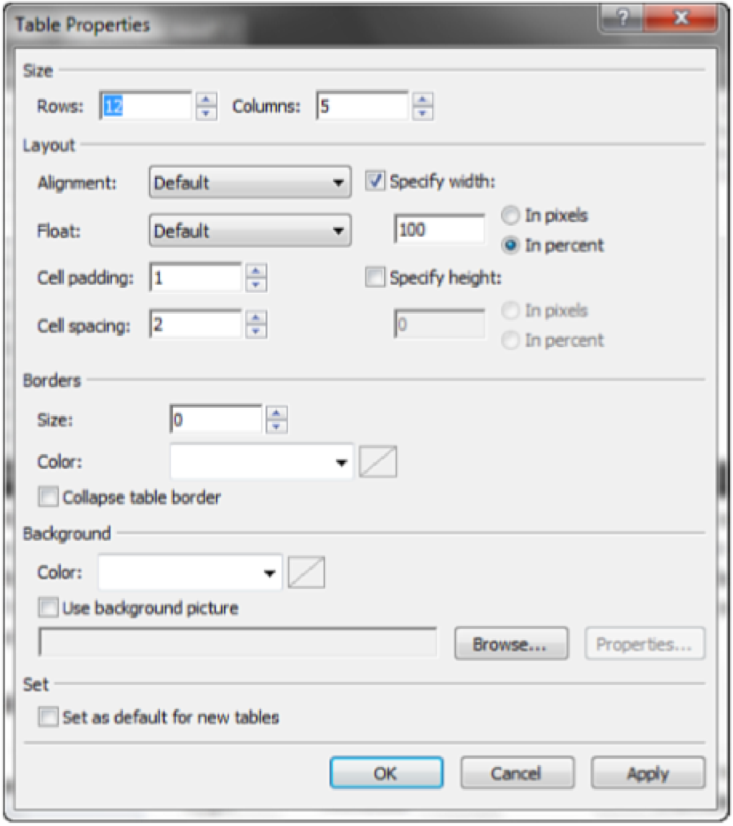


Table Properties

Size

Rows: 12 Columns: 5

Layout

Alignment: Default

Float: Default

Cell padding: 1

Cell spacing: 2

☑ Specify width: 100 ○ In pixels ● In percent

□ Specify height: 0 ○ In pixels ○ In percent

Borders

Size: 0

Color:

□ Collapse table border

Background

Color:

□ Use background picture

Set

□ Set as default for new tables

Using Table Properties to Change the Appearance of Your Table Cont.

- *Cell padding* is the space between the walls of the cell and the content within it.
- *Cell padding* of 0 pixels means that the content of the cell is touching the four walls, whereas cell padding of 5 pixels leaves an empty 5-pixel “buffer zone” between the walls and the content.
- *Cell spacing* is the space between the cells that separate them from each other.
- *Cell spacing* of 0 pixels means that the cells touch each other, whereas cell spacing of 5 pixels means there is a 5-pixel gap between each cell.
- *Cell spacing* is especially useful if you want to create a table with clearly defined borders between the cells because the cell spacing area is transparent and shows the background color of the table.

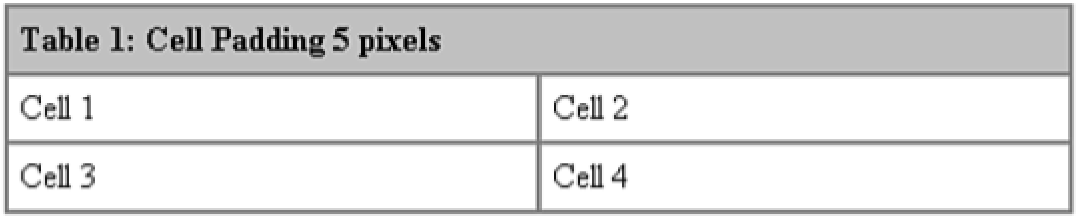


Table 1: Cell Padding 5 pixels	
Cell 1	Cell 2
Cell 3	Cell 4

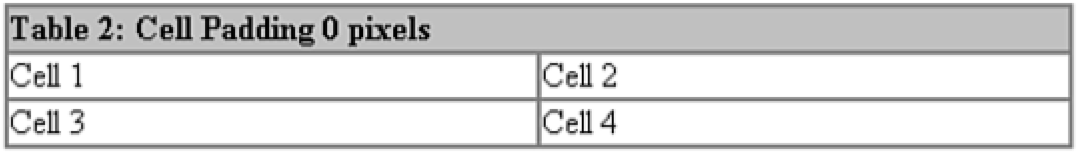


Table 2: Cell Padding 0 pixels	
Cell 1	Cell 2
Cell 3	Cell 4

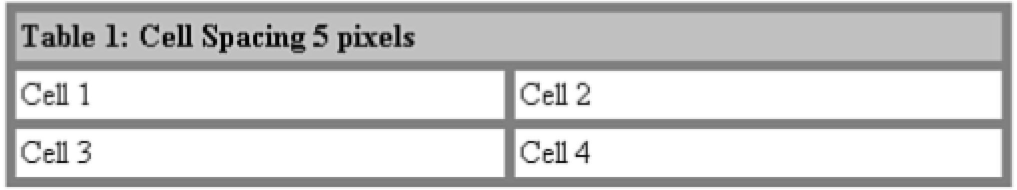


Table 1: Cell Spacing 5 pixels	
Cell 1	Cell 2
Cell 3	Cell 4

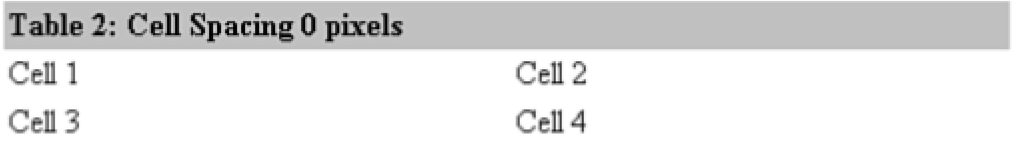


Table 2: Cell Spacing 0 pixels	
Cell 1	Cell 2
Cell 3	Cell 4

Using Table Properties to Change the Appearance of Your Table Cont.

- The *Borders* setting values define the thickness and color of the border of the table.
- The border is a box that surrounds the entire table.
- By default, the border is invisible (0 pixels).
- Depending on the settings, the border can be unique in that it has a 3D look to it: The top and left sides are slightly lighter in color than the bottom and right sides.
- The *Collapse Table Border* check box changes the relationship between the table border and the outer cell spacing.

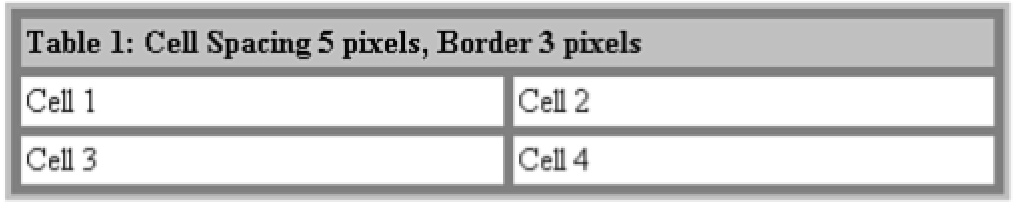


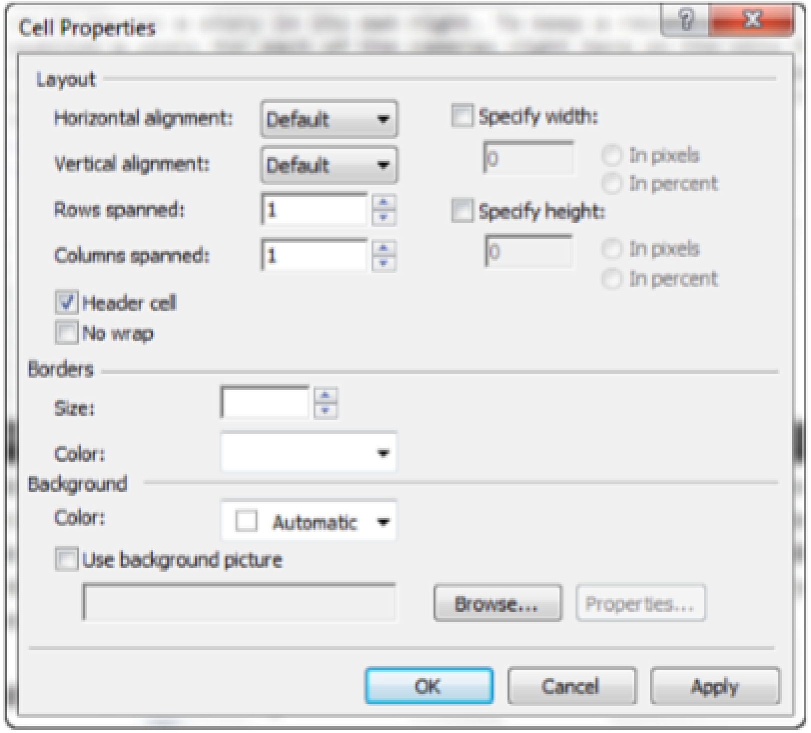
Table 1: Cell Spacing 5 pixels, Border 3 pixels	
Cell 1	Cell 2
Cell 3	Cell 4

Using Table Properties to Change the Appearance of Your Table Cont.

- The *Background* setting indicates the background color or background image of the table.
- By default, tables are transparent, and the background color or image shines through.
- If you define the color of a cell or group of cells, they will no longer be transparent.
- To make a table with white cells and blue lines around each cell, set the table background color to blue and the cell background color to white.
- The cell spacing remains transparent and lets the blue color shine through, and the white cells block the background color.
- The *Set value* lets you define the current table layout as the default layout for future inserted tables.

Using Cell Properties to Change the Appearance of Your Cells

- You can also change the appearance of a single cell, or group of cells, in a table by highlighting the cell and selecting *Cell Properties* from the context menu or choosing *Table*, *Cell Properties* on the menu bar.
- This opens the *Cell Properties* dialog.



Cell Properties

Layout

Horizontal alignment: Default

Vertical alignment: Default

Rows spanned: 1

Columns spanned: 1

☑ Header cell

□ No wrap

□ Specify width: 0 ○ In pixels ○ In percent

□ Specify height: 0 ○ In pixels ○ In percent

Borders

Size:

Color:

Background

Color: □ Automatic

□ Use background picture

Using Cell Properties to Change the Appearance of Your Cells Cont.

- From here, you can change the layout, borders, and background of each cell or a group of cells.
- The *Borders* and *Background* settings work in much the same way as in the *Table Properties* dialog, but there is some new functionality in *Layout*.
- The *Horizontal* and *Vertical Alignment* settings define where the content appears within the cell.
- *Horizontal Alignment* works just like alignment in a word processor (left, right, center, justified).
- *Vertical Alignment* defines whether the content within the cell should be positioned on the top, middle, or bottom of the cell.

Using Cell Properties to Change the Appearance of Your Cells Cont.

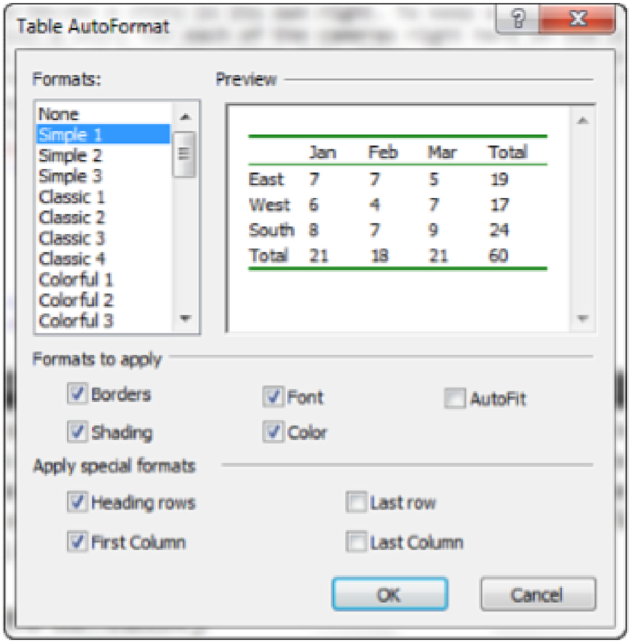
- The *Rows Spanned* and *Columns Spanned* settings change the layout of the table by letting the cells span more than one row or column.
- If you let one cell span two rows, the displaced cells in the row below shift to the right to make room.
- As a result, the rightmost cell protrudes from the right side of the table.
- If you let one cell span two columns, the remaining cells in the row shift to the right, and the leftmost cell protrudes from the right side of the table.
- These functions can easily make your table very confusing, so use them with care.

Using Cell Properties to Change the Appearance of Your Cells Cont.

- The *Specify Width* and *Specify Height* settings give you the ability to specify the width or height of the selected row or column in pixels or percentages.
- The effect is the same as when you used the mouse to drag the width and height of the rows and columns but is more precise because you set the values manually.
- The *Header Cell* setting changes the content of the cells to center-aligned and bold, as is normal in table headers.
- The name *Header Cell* is a bit misleading because you can apply the effect to any cell in a table, not just the header.
- The true purpose of a *Header Cell* is that in the markup the tag changes from *<td>* to *<th>*, which identifies it as a header for styling and accessibility purposes.
- The *No Wrap* attribute tells the cell not to split lines even if they don’t fit within the cell.
- If *No Wrap* is checked, lines of text that are too long to fit on one line do not continue on the next line but instead hide behind the next cell.

Using AutoFormat to Quickly Change the Look of Your Tables

- Making tables more interesting to look at can be an arduous task.
- To help ease the workload, Expression Web 4 has a long list of prepackaged table layouts at your disposal through the *AutoFormat* option.
- To apply one of these layouts, all you have to do is select the table you want to change and click the *Table AutoFormat* button or the *AutoFormat Combo* drop-down menu on the *Tables* toolbar.
- Clicking the *AutoFormat* button opens the *Table AutoFormat* dialog.



	Jan	Feb	Mar	Total
East	7	7	5	19
West	6	4	7	17
South	8	7	9	24
Total	21	18	21	60

Formats to apply

☑ Borders ☑ Font □ AutoFit

☑ Shading ☑ Color

Apply special formats

☑ Heading rows □ Last row

☑ First Column □ Last Column

Using AutoFormat to Quickly Change the Look of Your Tables

- From here, you can preview the various layouts available and make changes to them using the check boxes.
- All these table layouts are different, and the check boxes affect them in different ways depending on the layout, so it’s a good idea to experiment to get that perfect look.
- When you are satisfied with what you see in the preview, click OK, and Expression Web 4 applies the *AutoFormat*.
- If it is not what you want, simply press *Ctrl+Z* or select *Edit*, *Undo* from the menu bar and try again.
- If you already know what *Table AutoFormat* you want, you can select it from the *Table AutoFormat Combo* drop-down menu on the *Tables* toolbar.
- This bypasses the *Table AutoFormat* dialog and applies the *AutoFormat* directly.
- *Table AutoFormat* creates a series of different CSS styles and applies them to the table.