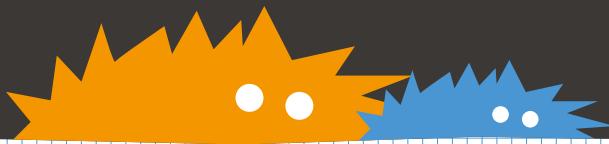




A guide to Web Development 101 using WebMatrix



CONTENTS

Part 1	Introduction	2
–	What is WebMatrix	3
–	Installing the program	4
Part 2	Web Development Basics	5
–	Introduction to HTML	5
–	HTML Basics	5
–	Understanding the Markup	6
–	Creating your first page	8
Part 3	ASP Web pages	12
–	An introduction to Razor	12
–	Your first Web page	17
–	The Web Server	21
–	The Page Title	22
–	The Page Body	23
Part 4	Publishing	24
–	Deploying your site	24
–	Setting Up a Web Hosting Account	27
–	Configuring WebMatrix for Publishing	29
–	Checking for Publishing Compatibility	33
Part 5	Building a WordPress website	35
–	Introduction to WordPress	35
–	How to make Blogging on WordPress even Easier	35
–	Setup Instructions	36
Part 6	Building an Umbraco website	38
–	Introduction to Umbraco	38
–	Umbraco: Changing your style	38



PART 1

Introduction

I remember when I first started developing for the Web back in 1998; I pressed View-source on a web site and saw a whole load of HTML characters and junk staring back at me. It looked like a mystical language, a secret code that was only understood by a select few. Try not to let that initial complication put you off, the fact is, building web sites is a great deal easier than it first looks, and with tools like WebMatrix it's a whole lot simpler than it was back in 1998.

Web development changed my life, the skills I have acquired over the years have taken me out of a small sleepy town in the middle of England, through numerous large companies, across countries and continents and finally onto the doorstep of Microsoft - the world's largest software company. If you're just beginning web development then WebMatrix is a fantastic place to start, but be warned...web development may change your life forever.

Martin Beeby – Developer Evangelist, Microsoft

When WebMatrix was launched in January 2011 I was over the moon – a tool that allows both new developers to get up and running with websites quickly, and let those who have a ton of experience churn out high quality projects on a frequent basis – available totally free of charge. The fact it pulls the latest products, features, and apps to your fingertips without having to set up the stack yourself lowers the barrier to new developers, which means we'd hope to see even more creativity on the web in the coming years – people can focus on the site, the app, the design itself and less time on what holds it all in place.

There's plenty of tutorial content out there about WebMatrix – from download to deploy to deep technical guidance on how to make world-class websites. In the spirit of the product itself, we wanted to pick the best of the bunch and bring it to your fingertips. So this book is a collection of what we feel is the best out there, from some of the best technical minds – and structured in a way that lets you get familiar with the product and produce your own sites easily, quickly and to a higher quality than you might have imagined.

Andy Robb – Developer Evangelist, Microsoft

To get more information on WebMatrix visit <http://www.ubelly.com> and <http://www.microsoft.com/web>

What is WebMatrix?

Microsoft WebMatrix is a web development tool that makes it easy for anyone to create a new web site. Users can start from any of the built-in or online templates or any of the free popular open source ASP.NET or PHP web applications, such as WordPress, Joomla!, DotNetNuke or Umbraco, located in the online Web Applications Gallery or use a pre-built website template. Users can customize their site using the code and database editor within WebMatrix. When a user is ready to publish their site, WebMatrix provides access to offers from our hosting partners within the hosting gallery makes it easy to choose the right hosting partner for the app being created with a selection of hosting packages. Once a hosting package is selected and configured, WebMatrix then seamlessly publishes the site and database to the Web.

Who is it most useful for?

As web development evolves, web site developers are looking for new ways to build web sites faster and make their job easier. WebMatrix meets this by providing a free set of tools that gives them the ability to create new sites, by either writing the code themselves, or by using existing free open source software applications. They can customize their sites by editing their code, data and much more, and they can publish their sites easily to the Internet. When it comes to choosing where and how to publish, WebMatrix includes a gallery of hosting providers that are compatible with the site the developer is building.

WebMatrix is for anyone who wants to build web sites, but it is particularly useful for:

- **Web Agencies** who want to use free open source web applications. WebMatrix provides a way to download and install these quickly, with tools to configure and optimize their code, along with integrated deployment tools to publish the finished site.
- **Developers** who want to build powerful data driven web pages quickly and easily with simple, but powerful programming tools.
- **Students** and new developers who want a free, simple, solution that helps them learn how to build web pages using either open source applications, or by writing all the code themselves, with an integrated deployment experience that allows them to easily publish their work.

What is actually included in WebMatrix?

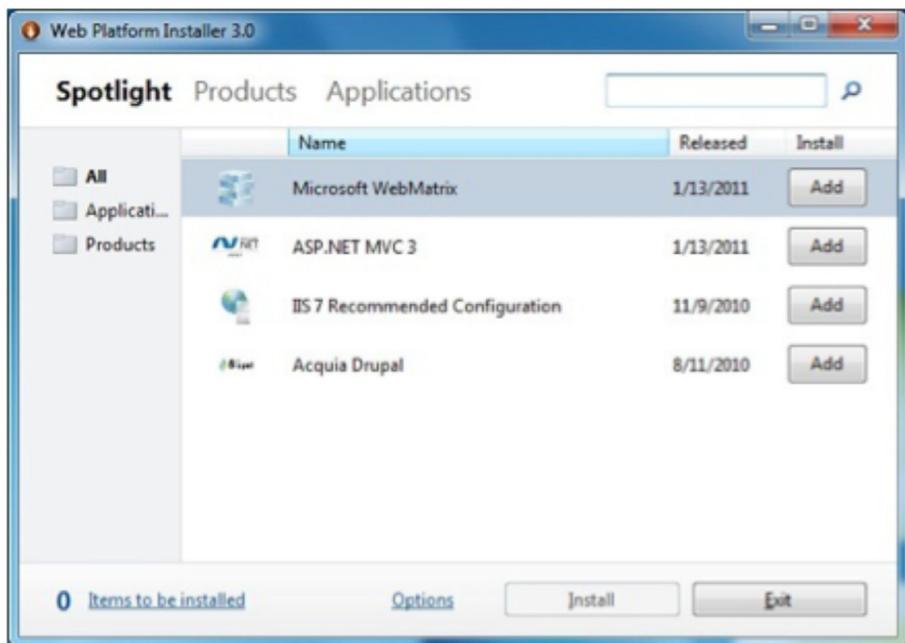
WebMatrix includes IIS Express (a development web server), ASP.NET (a web framework), and SQL Server® Compact (an embedded database). It also includes a simple tool that streamlines website development and makes it easy to start websites from popular open source apps. The skills and code developed with WebMatrix transition seamlessly to Visual Studio® and SQL Server.

Web pages created using WebMatrix can be dynamic; they can alter their content or style based on user input or on other information, such as database information. To program dynamic web pages, a user chooses ASP.NET with the Razor syntax and with the C# or Visual Basic programming languages.

Installing the program

To install WebMatrix, you can use Microsoft's Web Platform Installer, which is a free application that makes it easy to install and configure web-related technologies.

1. If you don't already have the Web Platform Installer, download it from <http://www.microsoft.com/web>
2. Run the Web Platform Installer, select **Spotlight**, and then click **Add** to install WebMatrix.





Web Development Basics

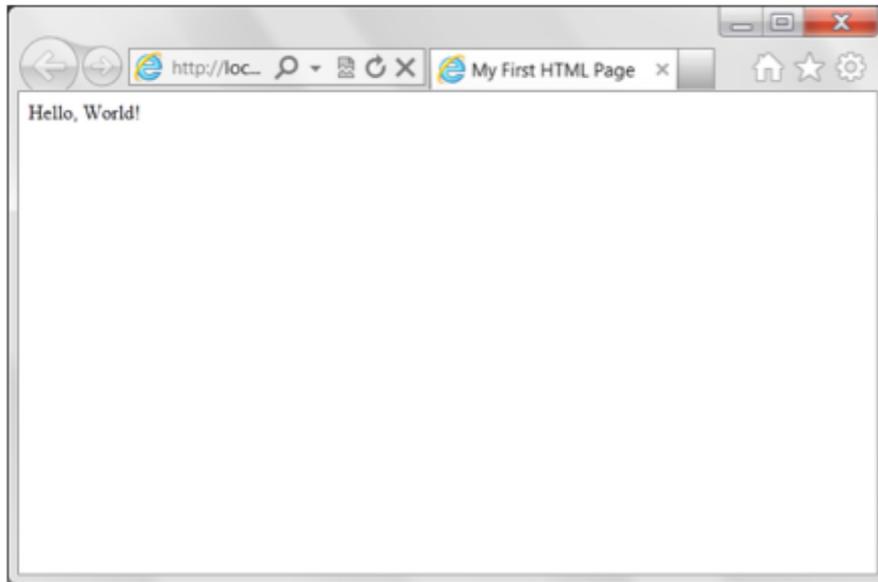
Introduction to HTML

HTML stands for HyperText Markup Language. It's a type of text document, where text is 'marked up' by using special tags that inform a program that reads the text in how to render the text. Typically that program is a Web Browser such as Internet Explorer, FireFox, Opera or Chrome. With the growth of Internet-connected devices such as cell phones, slates and embedded systems (such as TVs), HTML is more important than ever, as it provides a near ubiquitous cross platform way of delivering content.

Despite that, because of HTML's age, it's pretty much taken for granted! It's pretty hard to find basic 'getting started' tutorials that will teach you what it is, and how to use it, from the ground up. Well, that's what we're trying to fix here. Hope you enjoy it!

HTML Basics

Here's an example of a very simple HTML document running within Internet Explorer 9.



You'll notice the content that appears in the main part of the browser. It reads 'Hello World' - this is called the body. Now, let's take a look at the HTML that generates this page.

```
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    Hello, World!
  </body>
</html>
```

Remember earlier where we mentioned that HTML is a special way of marking up text so that the browser can render it. You can see this in the above listing.

Understanding the Markup

First is the **Title**, as you can see the text 'My First HTML Page' is surrounded by `<title>` and `</title>`. This is Markup, where you start a piece of content with `<something>` and end it with `</something>`, so the content that you want to be the Title is marked up using `<title></title>`.

Similarly the **Body** is marked up using `<body></body>`, so because the text 'Hello, World!' is within the `<body>` tags, the browser will render it within the 'main' area that the user reads.

You'll see that there are a couple more tags. The first is the one that wraps the entire document with `<html></html>` which is just specifying that this document is an HTML document. Simple, right?

The other tag, the `<head>` component, is a little more interesting. This is used to contain content that is processed and rendered before the body, and is typically used for things that are independent of the body text, such as the title, and stuff like scripts and styles that you'll learn about later. As the `<title>` is independent of the body (it is rendered on the browser UI, not within the body of the browser), it's held within the head.

PART 2

You might also notice that the structure of HTML is *nested*. This means that tags can be contained within tags. You can see that the `<head>` is within the `<html>` and that the `<title>` is within the `<head>`. Tags are nested by opening them before the previous one is closed, so you can see the structure:

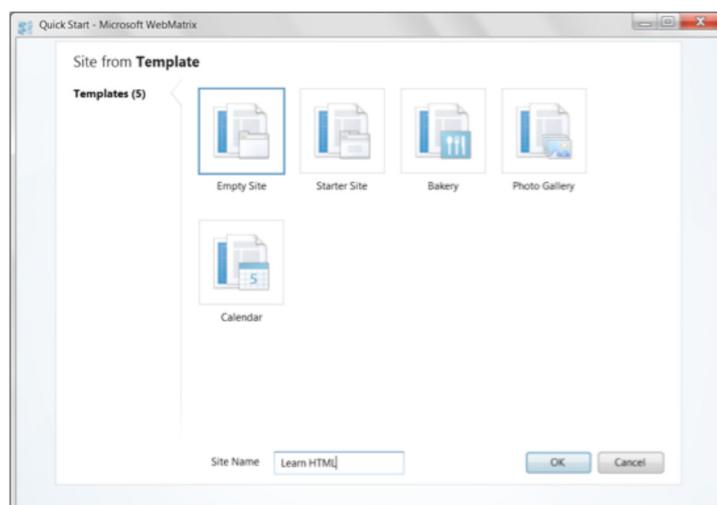
```
<html> - Open HTML  
<head> - Open Head  
<title> - Open Title  
</title> - Close Title  
</head> - Close Head  
<body> - Open Body  
</body> - Close Body  
</html> - Close HTML
```

It's important to understand this structure. Your HTML document will always have these tags and in this order. It's also important to remember how tags are opened and closed. As you write more HTML, you'll soon discover that one of the most common bugs you'll find when your page doesn't render correctly is that nesting is wrong, or tags aren't closed etc. Documents can soon become pretty complex, so it's good to get started on the right foot!

Creating your first page



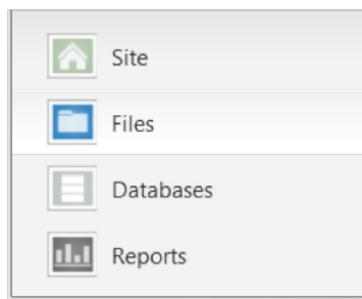
Select the 'Site From Template' link, and you'll see the list of templates that WebMatrix supports. Templates are simple 'web sites-in-a box' that give you lots of useful things such as a simple (but fully functional) eCommerce site for a bakery store. It's pretty easy to customise it for other things, but for now, let's just focus on HTML, and you'll use the 'Empty' template.



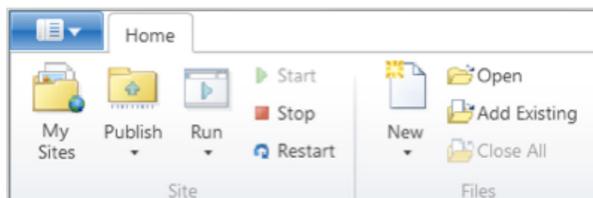
Give it a name and press 'OK'. WebMatrix will create your empty site, and will launch the WebMatrix workbench.

PART 2

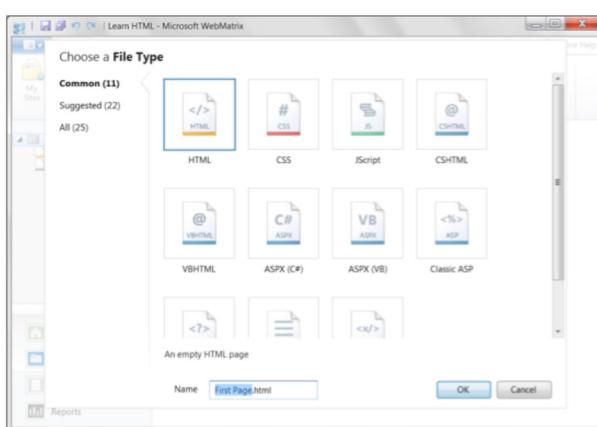
On the left hand side of the screen, you'll see a number of links for the different types of workspace that WebMatrix provides. Press the 'Files' link to open the Files workspace. This is where you create files, including those you build in HTML. Once you've done this, you'll see an 'Add file to your site' link. This will open up a dialog that offers you lots of different file types.



Alternatively, at the top of the screen, there's a 'New' button, that you can use to get the same dialog.

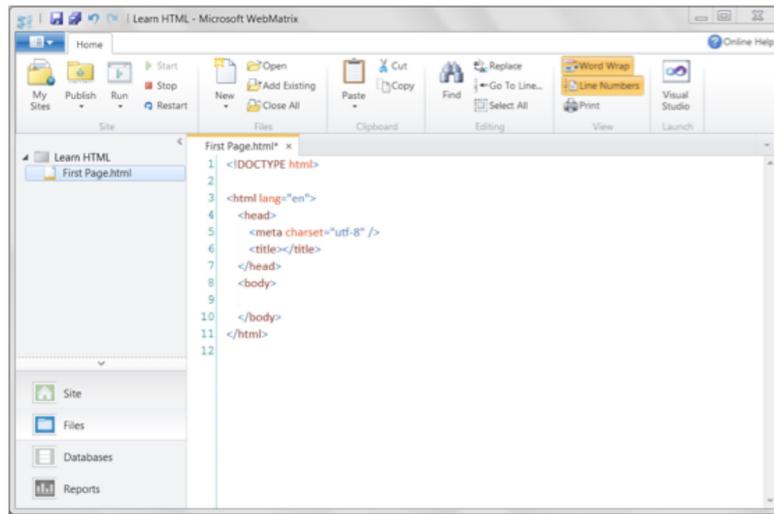


Either way, when you ask for a new file you'll get this dialog:



Select the 'HTML' option and call it 'FirstPage.html'. Press OK, and WebMatrix will create the HTML page for you and open it in the editor.

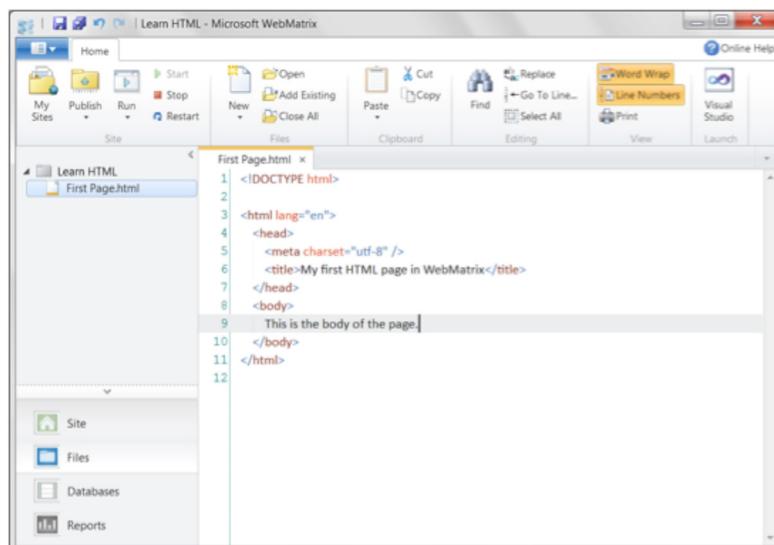
PART 2



You'll see that it has all the tags that we were talking about earlier, with a couple extra.

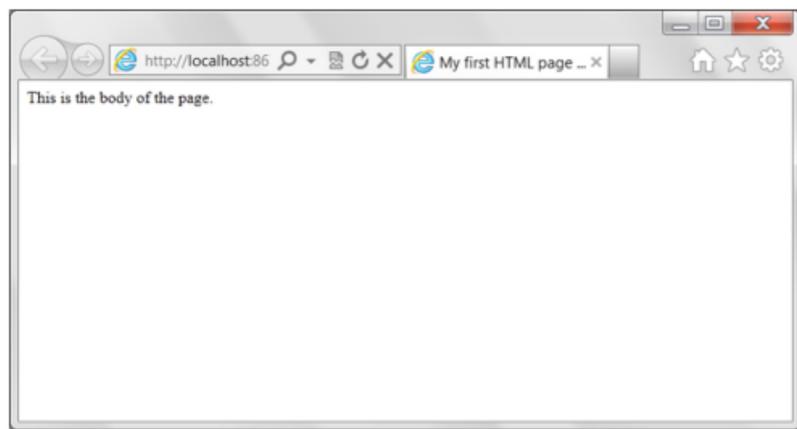
- <!DOCTYPE html> at the top of the page just tells the browser the document type, and it's a plain, vanilla HTML file.
- <meta> is used to contain information about the page. In this case it's specifying that the character set the page is using is something called 'utf-8'. It's English-centric.
- Additionally you can see lang='en' within the <html> tag definition. This is called an attribute. Attributes are used to describe a tag. In this case the HTML document is being described as being an English page.

Using WebMatrix, put some text between the <title> tags, and between the <body> tags, like this:



PART 2

Press the 'Run' button at the top of the page, and you'll see the page running in your default browser. You'll notice that the address of the page is <http://localhost:86/FirstPage.html>. This is because your page is actually running on a server on your machine, just like it would on the Internet.



So now you have an editor that you can use to edit your HTML, as well as a server on which your HTML runs. It's been pretty simple so far, right? In the next section, you'll see some more markup that you can use within the body of your page to make the content more attractive!

ASP Web pages

An introduction to Razor

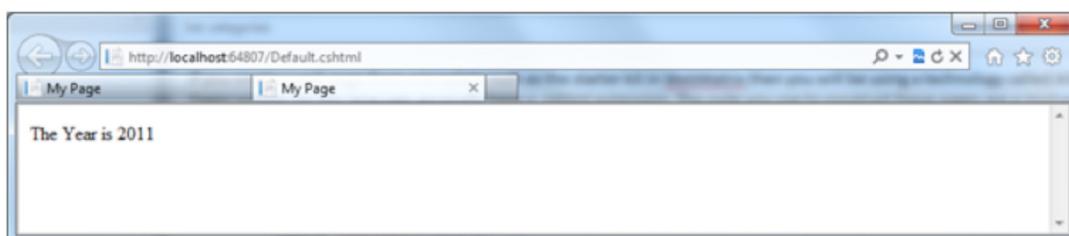
Written by Martin Beeby - Developer Evangelist, Microsoft

If you create a web page from a template such as the starter kit in WebMatrix then you'll be using a technology called ASP.NET web pages. Pages written for this language generally have a .cshtml extension. The code you use to construct these pages are a mixture of HTML and Razor.

Razor is a server side language that allows you to do more complicated things than HTML allows on its own. In the example below I've used Razor to get the current year from the server and print the year on the page.

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8" />  
    <title>My Page</title>  
  </head>  
  <body>  
    The Year is @DateTime.Now.Year  
  </body>  
</html>
```

If we run this we will see the following web page:



So how does this work? Well, when the server that is hosting the website is asked to serve the page to the user it will start sending the HTML to the browser. Anywhere it finds the @ character it will realise that this is Razor code and will execute it on the server. It will continue executing the code until it finds the next bit of HTML at which point it will switch back to sending the HTML to the browser.

In the real world, however, we might want to do something a little more complicated than just print the current year. Luckily razor enables you to do things like "If and Else logic" too. For example, if we wanted to write a different sentence depending on the year we could do something like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>My Page</title>
  </head>
  <body>
    @{
      string stringToPrint;
      if (DateTime.Now.Year > 2012)
        {stringToPrint = "It's greater than 2012";}
      else
        {stringToPrint = "It's less than 2012";}
    }
    @stringToPrint
  </body>
</html>
```



The team that built Razor wanted to make it easy to switch from code to HTML and so the language only requires you to use the @ character when you want to dive into code. You may have noticed in the example above I also used some {} brackets. This simply enables you to write code that spans multiple lines; this is often referred to as a code block. There are times when you want to use the @ character but you do not want to tell the page to execute code, for example if you are writing an email address.

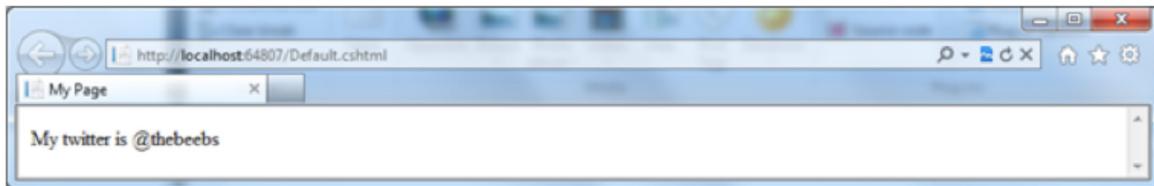
In the example below Razor is clever enough to realise that you are writing an email address and not breaking into code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>My Page</title>
</head>
<body>
  My email is martin.beeby@microsoft.com
</body>
</html>
```



However if you are writing a Twitter address then you'll need to add an extra @ character to let Razor know you are not writing code.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>My Page</title>
</head>
<body>
My twitter is @@thebeeb
</body>
</html>
```

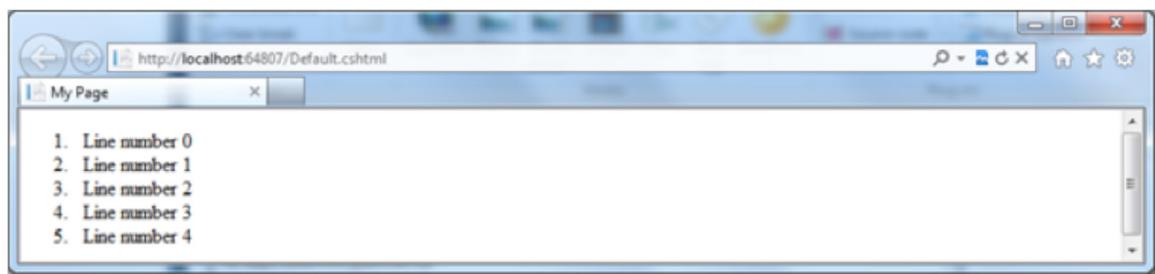


The last thing I'll show you in this section is how to create a list using a loop. Below I'll attempt to write the same sentence five times. To do this we use the for loop. To create a for loop we need to provide three pieces of information.

1. The variable that will count the number of times we have looped. In the example below we have used the variable name **i** but this can be called whatever you like e.g. **var i = 0;**
2. How long the loop should go on. In the example below we have said for as long as the loop counter is less than 5 e.g. **i < 5**
3. How much should we increase the loop counter by each time we loop. In the example below we have said by one e.g. **i++**

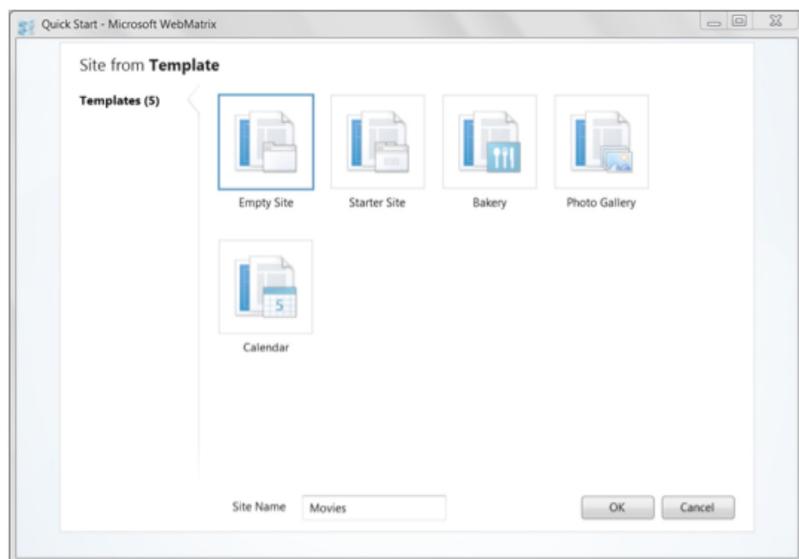
PART 3

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>My Page</title>
</head>
<body>
<ol>
@{
    for (var i = 0; i < 5; i++)
    {
        <li>Line number @i</li>
    }
}
</ol>
</body>
</html>
```

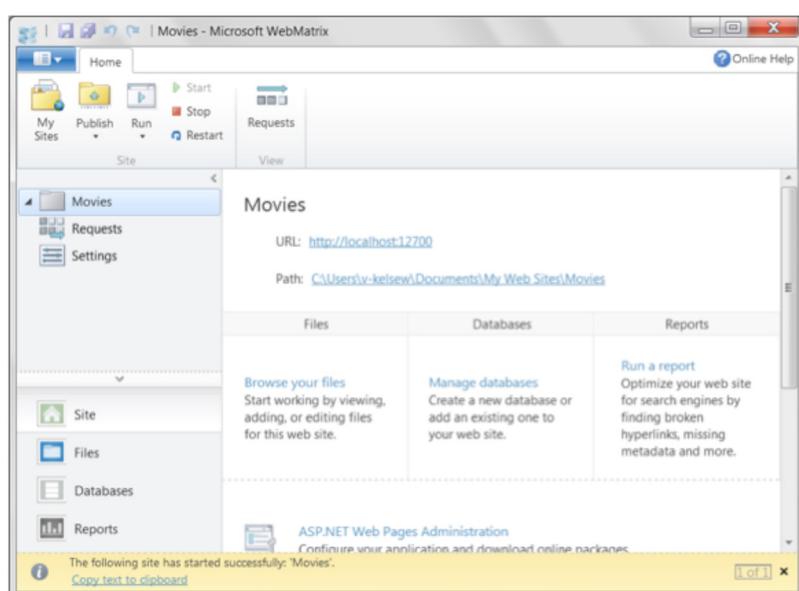


Your first web page

Choose the Site From Template option, and you'll see the dialog below. Note that you may see a number of different templates, as WebMatrix is growing all the time. The one you are interested in is the *Empty Site* template. Select this, and give it the name *Movies*.

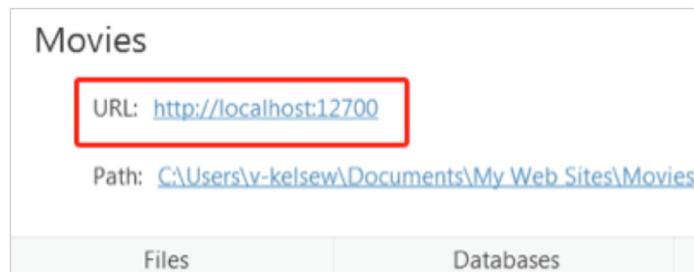


When you press OK, WebMatrix will create a new, empty, web site for you. This web site will then be loaded into the WebMatrix editor. You can see this here:



Before you go any further, it's good to understand a few things about what you see here. The first is that WebMatrix is more than just a code editing tool. It integrates a web server called IIS Express. A web server is a special piece of software that listens for requests for data over the Internet, and answers by delivering that data, usually to a web browser.

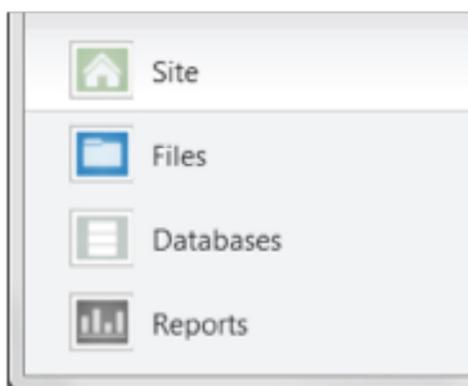
Whenever you open your browser and type something like <http://www.microsoft.com> you are calling the web server for Microsoft, and it answers by sending code such as HTML, JavaScript®, CSS, Pictures and more. Your browser then assembles them into a page.



Having a server built-in to WebMatrix makes it very easy for you to develop your web site in a way that behaves exactly like a web server on the Internet. If you look at the screen, below the name of the site (in this case 'Movies') you'll see that the server is serving this site at the address <http://localhost:8946>, which means that the *host* for the server is *local*, i.e. your development machine.

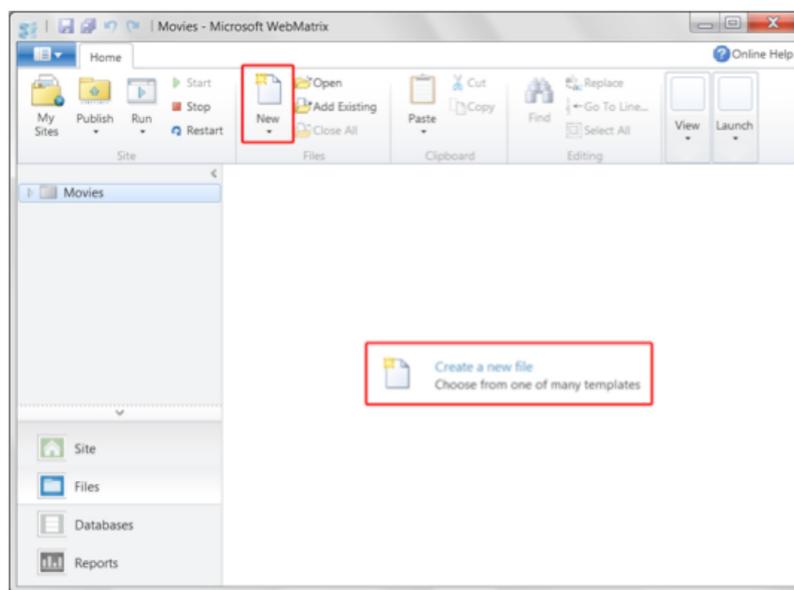
From within WebMatrix you can start the web server, and run your site, but if you do that now you'll get an error because you haven't yet created any content for the site. You'll do that in the next step.

You'll notice that WebMatrix allows you to move between different *workspaces* by selecting the buttons on the left hand side. At present the *Site* button is selected which gives you details on your web site, such as the URL of the site, and other tools that you can use such as monitoring your site requests. You'll look into each of these workspaces as you work through this section, but for now, select the *Files* workspace by pressing on its button.

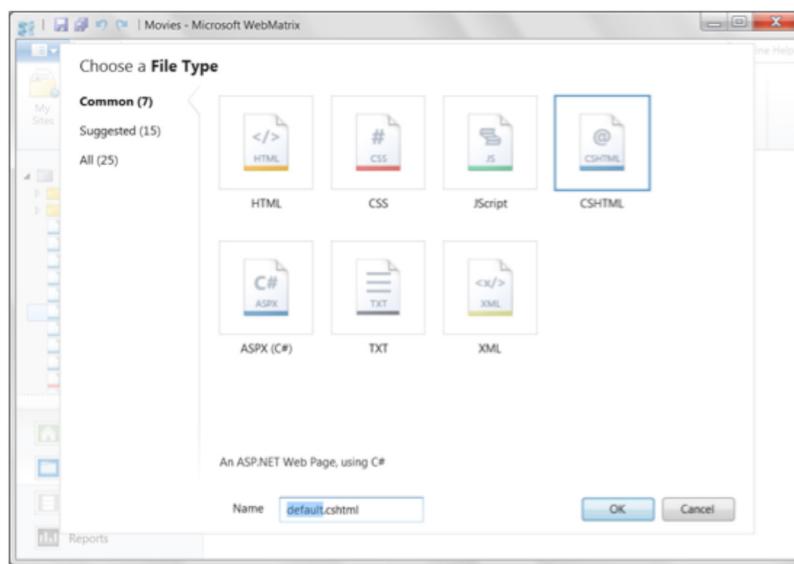


PART 3

WebMatrix will now open the *Files* workspace, and because you don't have any files in your web site yet, it looks pretty empty. However, it gives you a nice friendly button that allows you to add a file to your site, or you can use the *New* button in the toolbar to create a new file.



When you select either of these, you'll see the *Choose a File Type* dialog, which gives you lots of choices for lots of different file types that are typically used on the Web.



PART 3

Choose the HTML file type, and call it *default.html* and press OK. WebMatrix will now create a simple HTML file and open it for you.



A screenshot of a code editor window titled "default.cshtml". The code is as follows:

```
1  <!DOCTYPE html>
2
3  <html lang="en">
4      <head>
5          <meta charset="utf-8" />
6          <title></title>
7      </head>
8      <body>
9
10     </body>
11 </html>
12
```

Edit your page so that it looks like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>My Favorite Movies</title>
</head>
<body>
    <h1>A list of my Favorite Movies</h1>
    <ol>
        <li>It's a wonderful life</li>
        <li>Lord of the Rings</li>
        <li>The Fourth World</li>
        <li>The Lion King</li>
    </ol>
</body>
</html>
```

PART 3

You are entering some text into the `<title>` tag, as well as putting some code between the `<body>` tags. This code will use `<h1>` for Heading 1 style text, `` to tell the browser that you are rendering a list, and a few `` items to tell the browser that you are rendering some list items.

In the WebMatrix toolbar you'll see a 'Run' button.



Select it and WebMatrix will launch your browser, and open the web site running on your local server.



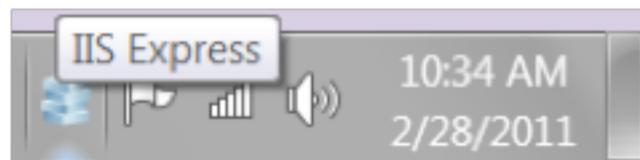
There's quite a lot happening here. Let's drill down and take a look.

The Web Server

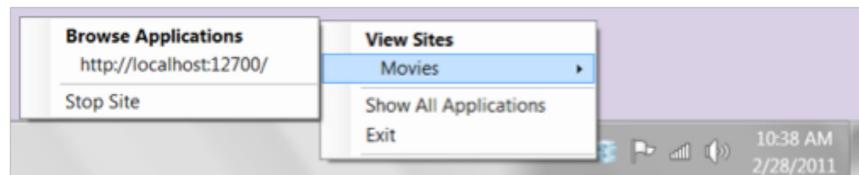
Notice the address bar on the browser? It hasn't opened the file from your hard disk, but has launched the web server, and pointed the browser at that web server, asking it to serve up the file default.html.



Look down at the system tray of your PC, and you'll see a little icon indicating that IIS Express, the web server, is running.



Right click on it, and you'll see that it is running your Movies site.

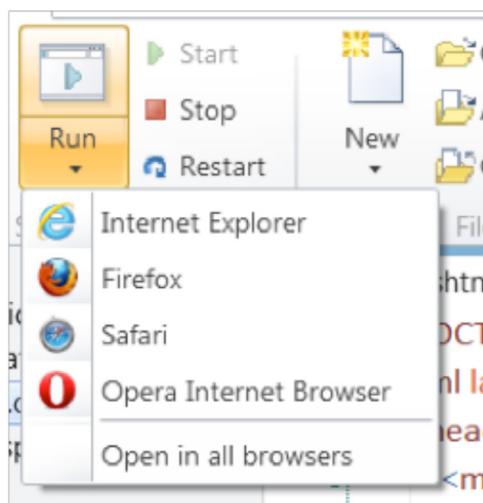


The Page Title

Look now at the browser tab for the page. It should contain the text 'My Favorite Movies'. For comparison, here's the same site running in Internet Explorer. Note: different browsers handle the page title in different ways.



A really nice aspect of WebMatrix, other than giving you a web server that runs on your development machine, is the ability to quickly launch any of your installed browsers. You can see this by clicking the down arrow at the bottom of the Run button in the WebMatrix tools ribbon.



The list will only show the browsers that you have installed.

The Page Body

In your code you used the `<h1>` tag around the phrase 'A list of my Favorite Movies'. This instructed the browser to use the 'Heading 1' style, and as such you can see that it was rendered in a large, bold font.

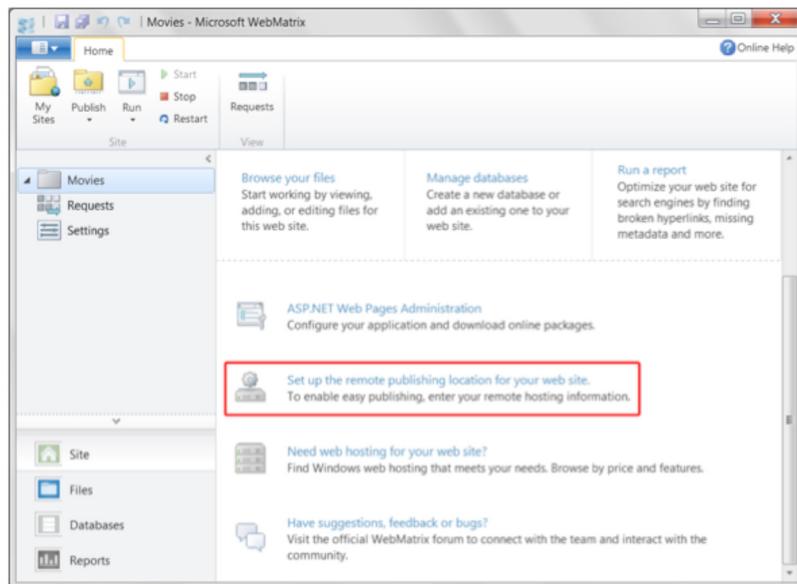


Finally, you wrote out the list of your favourite movies using `` (for 'Ordered List') and `` tags (for 'List Item'), and the browser rendered these. Note that you never put any bullets into the list, but the browser rendered bullets anyway – this is because it is the default style that the browser will use to render `` items. In part three of this tutorial you'll see how to modify this style to get the list items to render the way that you want them to.

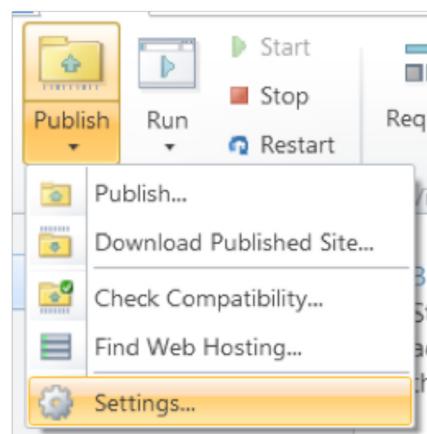
Publishing

Deploying your site

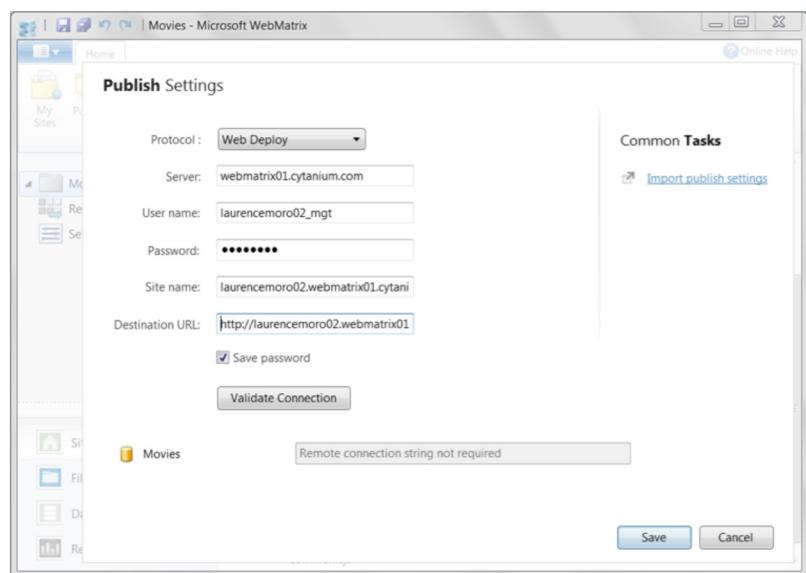
The next step is to get your application onto the Internet using a web hosting provider. WebMatrix ships with an integrated Web Hosting gallery which helps you find the best hosting providers to suit your site. If you go to the 'Site' workspace in WebMatrix you can see this:



Alternatively, you can just go straight to <http://www.microsoft.com/web/hosting> where you'll find the full gallery of web hosting providers with a variety of offers to suit your needs. Now, open the sites workspace in WebMatrix, and find the 'Publish' button. Drop it down by clicking the arrow at the bottom of it, and select 'Settings'



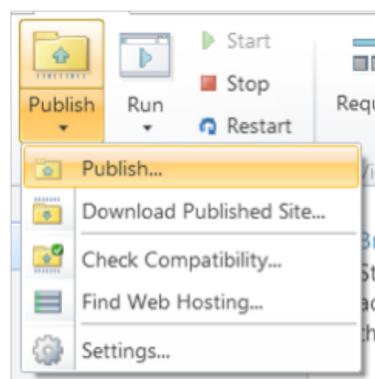
Publish Settings will open. Fill out the details that you got from a hoster's invite email:



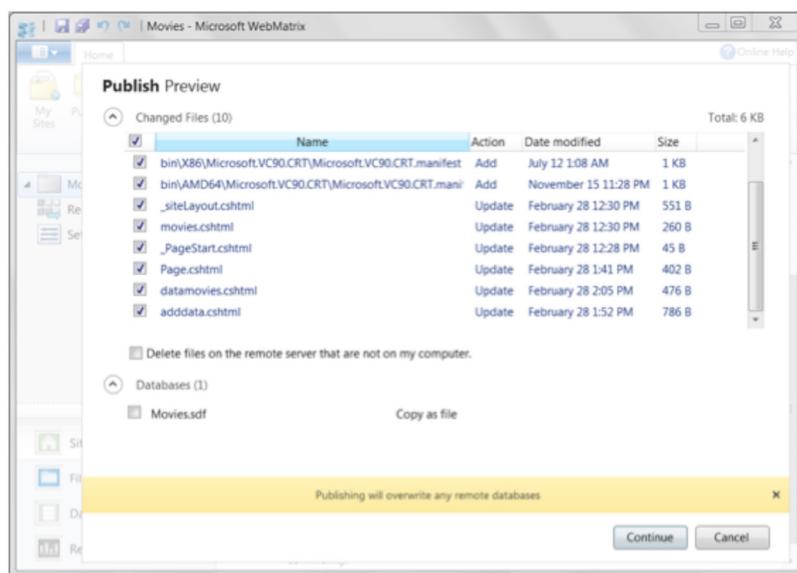
Click 'Validate Connection' to make sure that the settings are correct:



If the settings are correct you should see a 'Connected Successfully' tick as above. Click 'Save' to exit the Publish Settings dialog. Now, drop down the 'Publish' button again, and this time select 'Publish...'.



You'll see a 'Publish Preview' screen where WebMatrix determines the changes on your site. Depending on what you've done recently, you may see different settings from those here, particularly first time when there are lots of file dependencies on your development PC that aren't on your server. If that's the case, don't worry. Just make sure the 'Databases' box is checked as it is here and click 'Continue'.



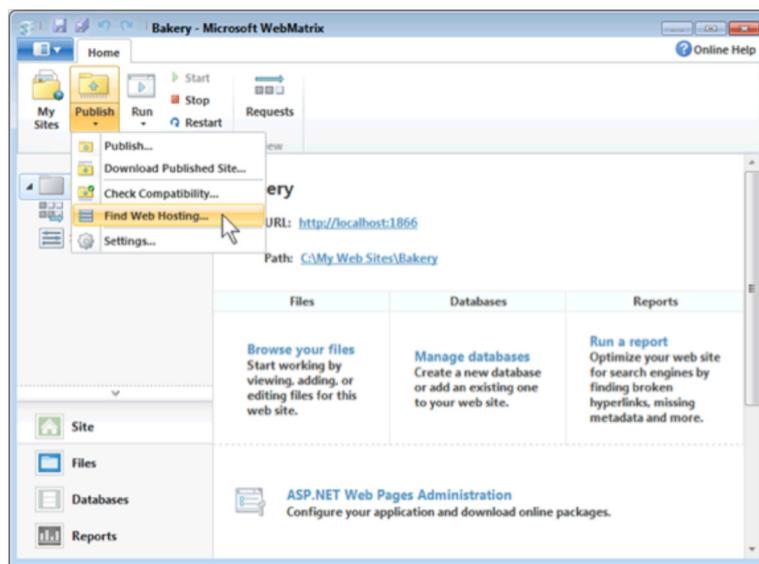
Publishing will start, and you'll see its progress in the yellow bar at the bottom of the WebMatrix window. When it's done you'll see that it's complete, and for convenience you'll get the URL of the finished site:



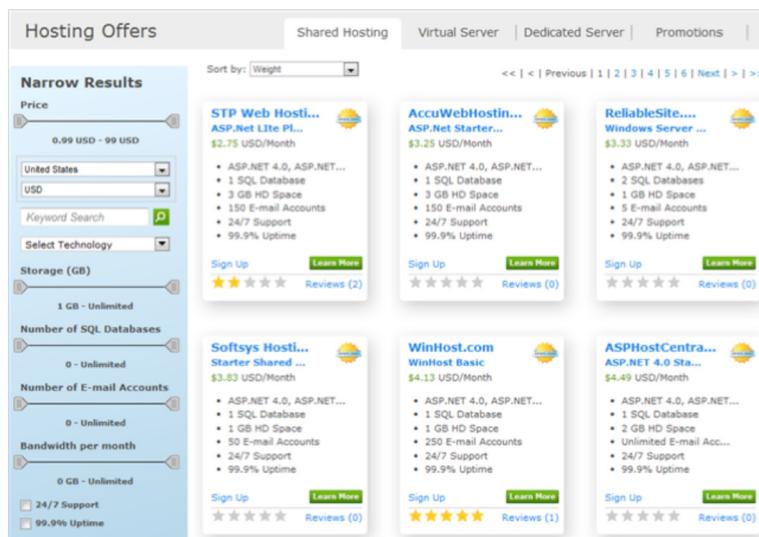
Setting Up a Web Hosting Account

To set up web hosting, you must find a web hosting provider and set up an account. To view some of the best hosting offers visit <http://www.ubelly.com>

In the **Home** tab, click the arrow below **Publish**, and then click **Find Web Hosting**.



The **Find Web Hosting** web page lists hosting providers.



1. Select one of the hosting providers and click **Learn More**.
2. Follow the directions provided by the hosting provider's site to set up a new account.

Support for a SQL Server database might be an option that costs extra. If you plan to publish a SQL Server Compact database (.sdf file), you don't need this option.

After your account is set up, the hosting provider typically sends you an email message that contains information you'll need in order to publish your website. The values might look like this:

- **Server:** webmatrix01.contoso.com
- **User Name:** user123
- **Password:** password
- **Site Name:** user123.webmatrix01.contoso.com
- **Destination URL:** <http://user123.webmatrix01.contoso.com>

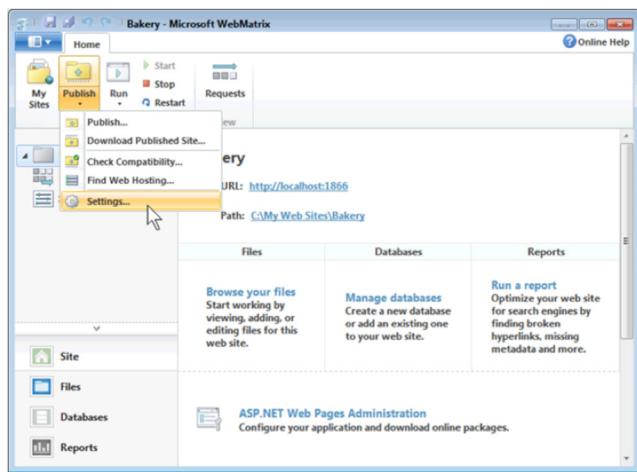
If your hosting account includes a SQL Server database, you'll also get information that you will need to publish your database, which will typically look like this:

Your database connection string is: Password=password;User ID=username;Initial Catalog=username;hostingcompany;Data Source=sql1.hostingcompany.com

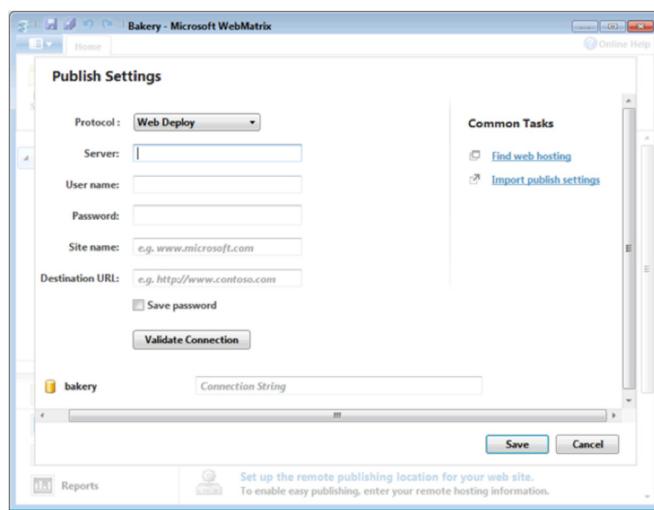
Some hosting providers will send you a *.PublishSettings* file which will contain these settings for you. When you configure WebMatrix for publishing, you can use the **Import Profile** feature to import these settings from the *.PublishSettings* file so that you won't have to type them in by hand.

Configuring WebMatrix for Publishing

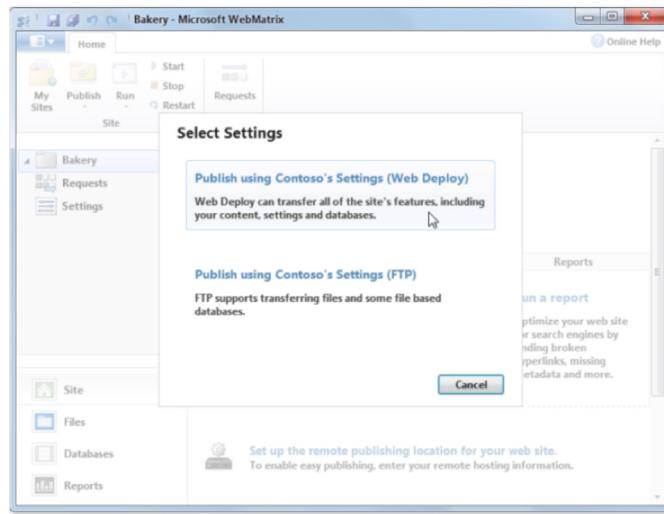
1. Open the website in WebMatrix.
2. In the **Home** tab, click the arrow below **Publish**, and then click **Settings**.



The **Publishing Settings** dialog box is displayed.



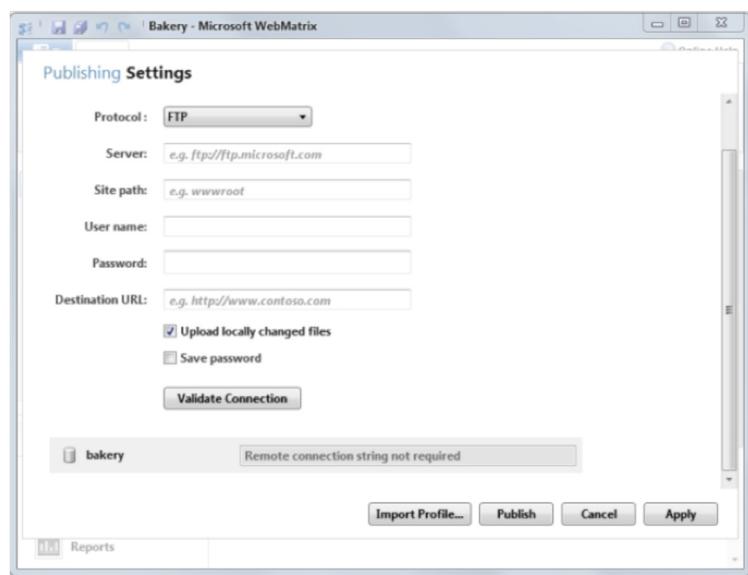
3. If you have the *.PublishSettings* file from your hosting provider, click **Import publish settings** to import the settings. Your profile may ask you whether you want to use Web Deploy or FTP to publish. If you selected a hosting provider through WebMatrix, you can use Web Deploy. If you're using a hosting provider that doesn't support Web Deploy, or if you want to publish using FTP instead of Web Deploy, select FTP.



4. If you don't have a *.PublishSettings* file, follow the steps below to enter your publishing settings by hand. Otherwise, you can skip down to the **Validate Connection** step.
5. If you selected a hosting provider through WebMatrix, you can leave **Protocol** set to **Web Deploy**. If you're using a hosting provider that doesn't support Web Deploy, or if you want to publish using FTP instead of Web Deploy, change the **Protocol** to **FTP**.
6. If you are using Web Deploy, enter the information you received from the hosting provider in the text boxes:

Option	What to enter
Server	Enter the "Service URL" or "Server Name" provided by the hosting company, like this: server1.lucernepublishing.com If the service URL includes the prefix https:// and the suffix :8172/msdeploy.axd , you can leave out the prefix and suffix.
User Name/ Password	Enter the credentials that the hosting provider provided for your account. (If you also signed up for a database for your account, the database might have a different username and password, so be sure that you use the right ones).
Site Name	Enter the "Site" or "Site/Application" value from the hosting company, like this: user123.server1.lucernepublishing.com
Destination URL	Enter the URL that people will use to browse your site over the Internet, like this: contoso.com -or- contoso.com/blog

7. If you want to use FTP instead of Web Deploy, select **FTP**. Some of the fields are different:



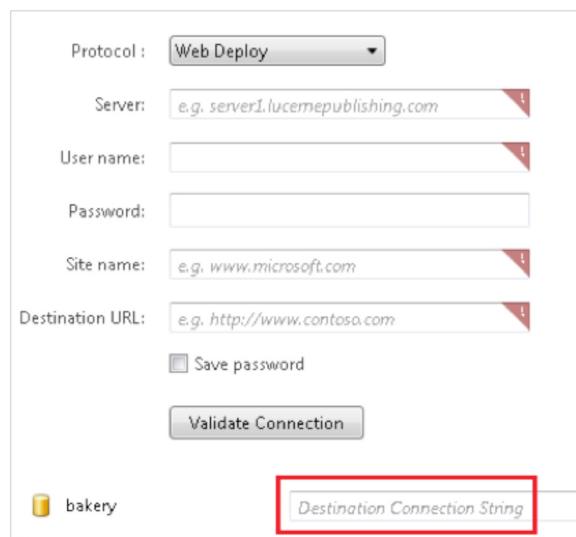
8. If you are using FTP:

1. Enter the **Server**, **User name**, **Password**, and **Destination URL** as described above for Web Deploy.
2. In the **Site path** box, enter the value identified by the hosting company as the "Site Root," such as /wwwroot. This field is optional.
9. If you want to upload only the files that have changed locally, select **Upload locally changed files**.
10. If you don't want to enter the password every time you publish, select the **Save password** check box.
11. Click **Validate Connection** to verify that WebMatrix is able to connect to the hosting provider. WebMatrix displays the results to the right of the button.



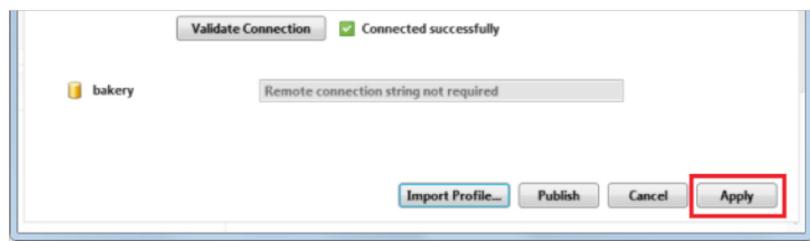
If the connection test fails, double-check that you entered the information correctly. If you did but validation continues to display an error, see the hosting provider's resources for help.

12. If your website uses a full SQL Server or SQL Server Express database (that is, not a SQL Server Compact database), enter the connection string that was sent to you by the hosting provider.



If the database doesn't appear in the dialog box, make sure it has a connection string in the *Web.config file*. If a SQL Server Compact database does not appear in this dialog box, make sure that the *.sdf* file is in the *App_Data* folder.

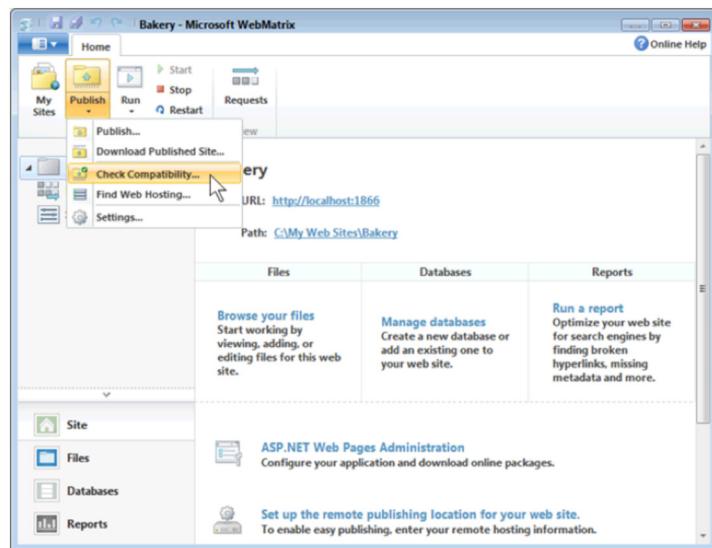
13. If you do not want to publish at this point and just want to save your publishing settings, click **Apply**, and then click **Cancel** to close the **Publishing Settings** wizard.



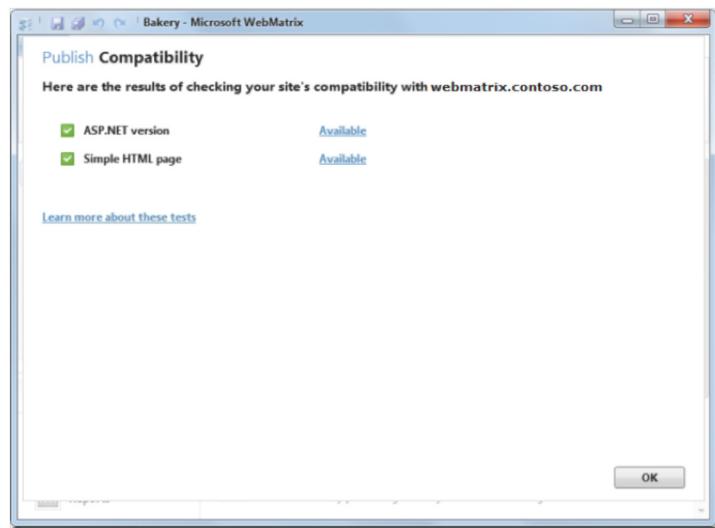
Checking for Publishing Compatibility

If you are not sure whether your hosting provider supports all of the software dependencies that your web application requires, you can have WebMatrix check this for you before you try to publish.

1. In the ribbon, click **Publish**, and then select **Check Compatibility**.



2. The results of the check appear in the **Publish Compatibility** dialog box.



3. Click **OK** to return to the main WebMatrix screen.

You do not need to perform this step every time you publish, or even at all. WebMatrix always checks for dependencies every time it publishes and will let you know if there is a problem. However, if you have a site that has many dependencies and takes a long time to publish, you may want to use this feature.

Building a WordPress website

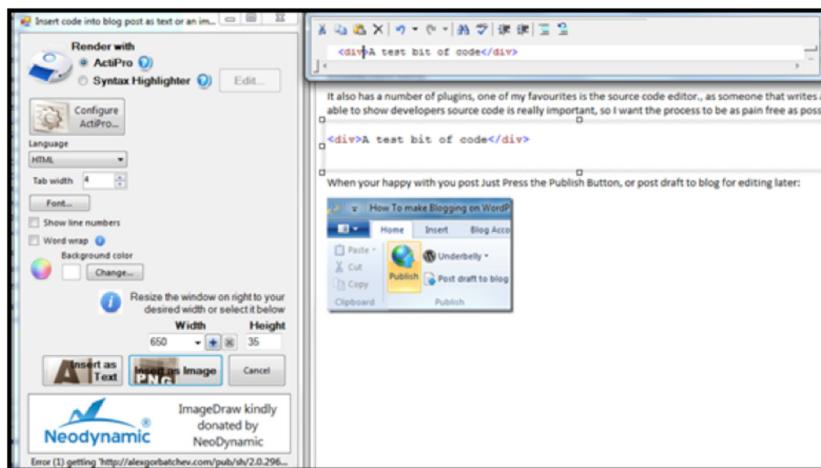
Introduction to WordPress

WordPress is the world's most popular CMS, enabling people without development skills to produce websites that look great quickly and giving skilled web developers a foundation on which they can extend and improve.

How to make Blogging on WordPress even Easier

Ubelly.com is built on WordPress, and it's wonderful. It's really easy to manage and customise. One thing that's a bit of a pain is writing blog posts using the web-based WordPress post editing screen. Although it's great to make a few changes, it can be a bit fiddly when it comes to adding pictures or code samples. That's where Windows Live® Writer can help, as it gives you a desktop editor which is a little easier to use. So for example, you can cut and paste images into a blog post directly from your computer and when you hit publish all those pictures will be saved up to the server for you. You don't have to upload them separately.

It also has a number of plugins - one of my favourites is the source code editor. I write a lot of developer tutorials and being able to show source code is really important. All I do is press the "Insert Source Code" button, choose the language, choose if I want it rendered using Syntax Highlighter, ActiPro or as a plain old PNG image and it inserts it straight into the blog. If I want to edit it later all I need to do is click on it and make the changes in the code window.



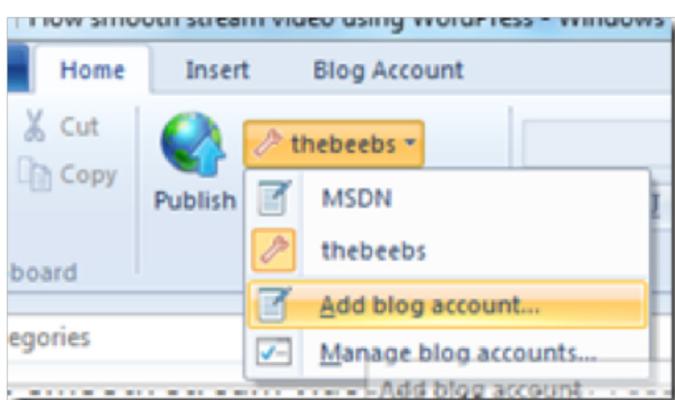
<div>A test bit of code, so that you can see</div>

When you're happy with your post you can press the Publish button and the blog is posted live on your WordPress website.

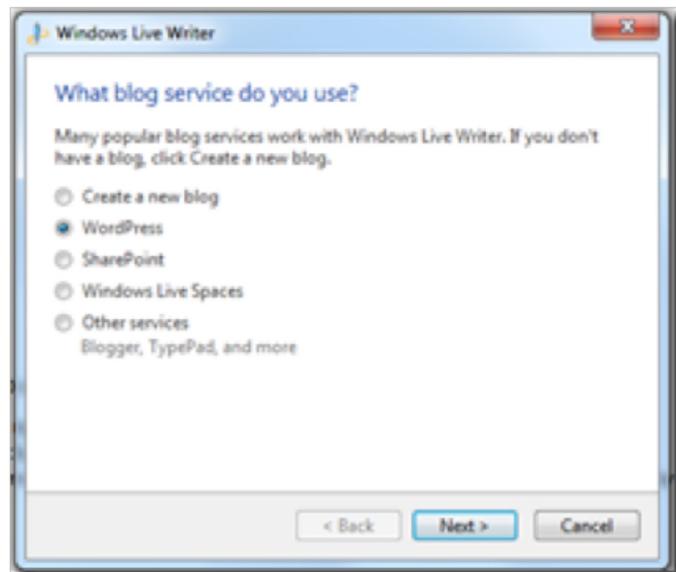


Setup Instructions

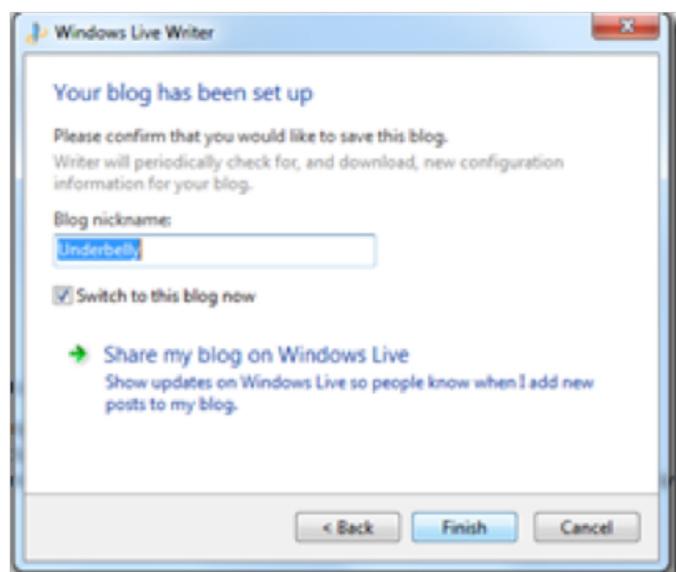
- Download the application and install it
- Select Add blog account



- Choose WordPress:



- You will be asked for your blogs URL. I entered ours <http://www.ubelly.com> and then entered my username and password.
- That's it your set up. You can now write your blogs and publish directly from your desktop.



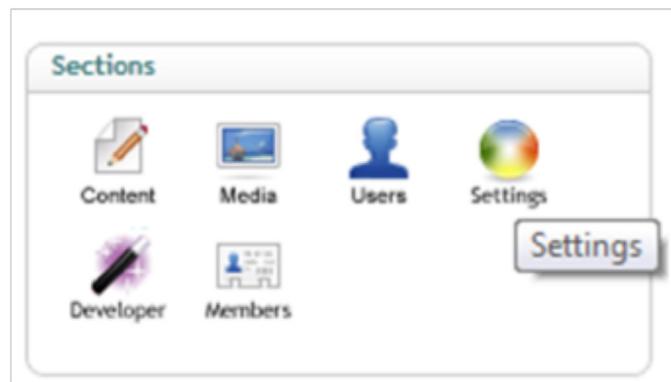
Building an Umbraco Web Site

Introduction to Umbraco

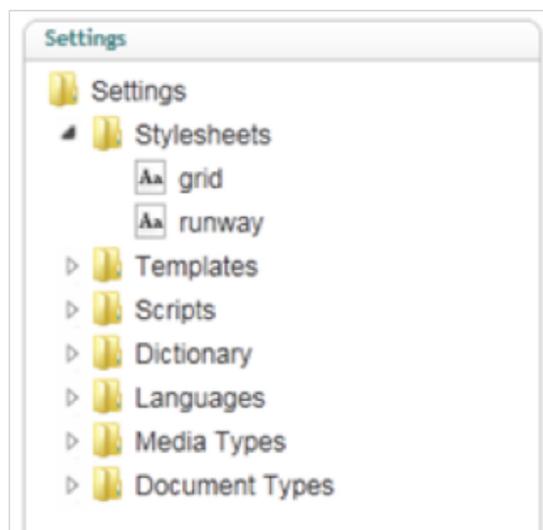
What do the bands Take That and Abba have in common? No it's not that they both have lead singers that like one too many cream cakes (I'm not mentioning names), it's that they both run their websites using Umbraco. Umbraco is a .Net open source content management system that powers over 85,000 websites and has one of the most friendliest developer communities in the world. You can install Umbraco similar to WordPress, as a Web App in WebMatrix.

Umbraco: Changing your style

When you install Umbraco, the basic Runway is installed. This gives you a basic skin for you to get started, but what if you want to make changes to the StyleSheet? One of the great features of Umbraco is the ability to change almost anything directly from the administrator dashboard. Luckily it's really simple. Just login to the Umbraco dashboard and in the bottom left hand corner there is a box called **Sections**. Click on the "Settings" section.

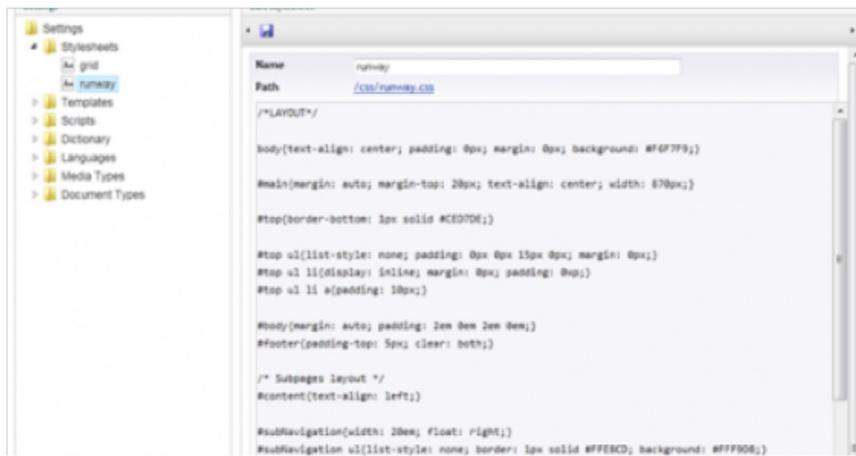


This will load the settings section in the left hand pane.



PART 6

If you click on runway you can then edit the style sheet directly in your browser. Changes you make will then be reflected on your site.



The screenshot shows the SharePoint Designer ribbon with the 'Runway' style sheet selected in the 'Style Sheets' list. The right pane displays the CSS code for the 'runway.css' file.

```
/* LAYOUT */

body{text-align: center; padding: 0px; margin: 0px; background: #F0F7F9; }

#main{margin: auto; margin-top: 20px; text-align: center; width: 870px; }

#top{border-bottom: 3px solid #CE00DE; }

#top ul{list-style: none; padding: 0px 0px 15px 0px; margin: 0px; }

#top ul li{display: inline; margin: 0px; padding: 0px; }

#top ul li a{padding: 10px; }

#body{margin: auto; padding: 2em 0em 2em 0em; }

#footer{padding-top: 5px; clear: both; }

/* Subpages layout */

#content{text-align: left; }

#subNavigation{width: 20em; float: right; }

#subNavigation ul{list-style: none; border: 1px solid #FFEB3B; background: #FFF90B; }
```