

Extracted from:

ExpressionEngine 2

A Quick-Start Guide

This PDF file contains pages extracted from *ExpressionEngine 2*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

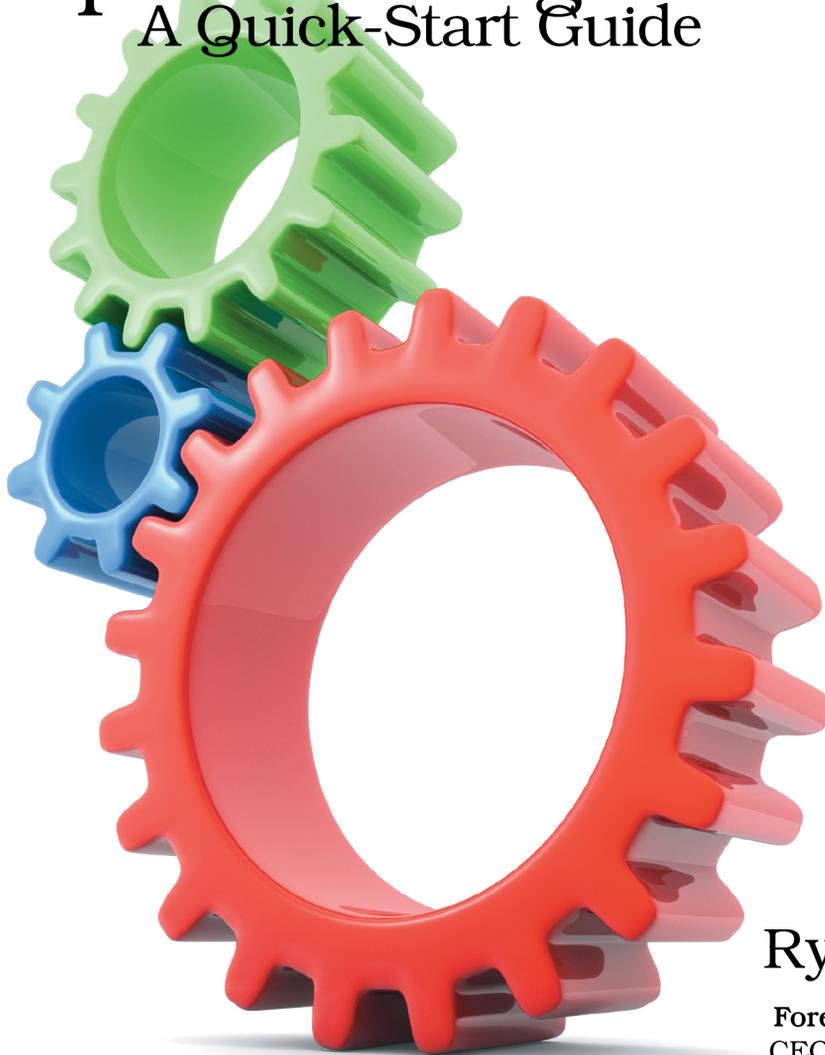
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

ExpressionEngine 2

A Quick-Start Guide



Ryan Irelan

Foreword by Rick Ellis,
CEO of EllisLab, creator
of ExpressionEngine

Edited by Susannah Davidson Pfalzer

Now that we've successfully installed and configured ExpressionEngine, let's jump right in and create our first web page. As we progress, we'll learn not only about templates but about ExpressionEngine variables and tags. By the end of the chapter, we'll have created a simple web page that displays our contact information and includes some blog entries. The work in this chapter will prepare us for the task of building our first website with ExpressionEngine.

2.1 Building Our First Template

Let's head back to the Control Panel (log in again, if necessary) so we can build our first template.

As you noticed in the previous chapter when we explored the Control Panel, ExpressionEngine has automatically created a series of template groups and templates during installation. But let's not worry about them right now. It's quicker and easier, and—most importantly—we will learn more if we just start from scratch. So, we will create our own template group and template for our new web page.

To create a new template group, click Template Group in the Create box on the home page of the Control Panel. We are now presented with the form for creating a new template group. Template group names must be a single word or multiple words connected by either underscores or hyphens. Spaces between words are not allowed (for example `my_template` is valid, while `my template` is not).

Since we're creating a single page with our contact information to which we can point business associates and friends, name the template group "contact." Some other options are available when creating a new template group (see [Figure 8, *Creating a new template group, on page 6*](#)), but we can leave them as is for now.

Click Submit to create the new template group. ExpressionEngine pops you back to the Template Management view. To get into the template group, select it from the list on the far left. You'll notice that there is already a template there called `index` with a small icon of a house next to it. What is this template? Well, ExpressionEngine creates an `index` template automatically when we create a new template group.

For right now, we'll ignore the `index` template and create our own. At the top right of the template list, click the red New Template button. Let's call this new template `my_info`. We want the template type to be "web page," and we

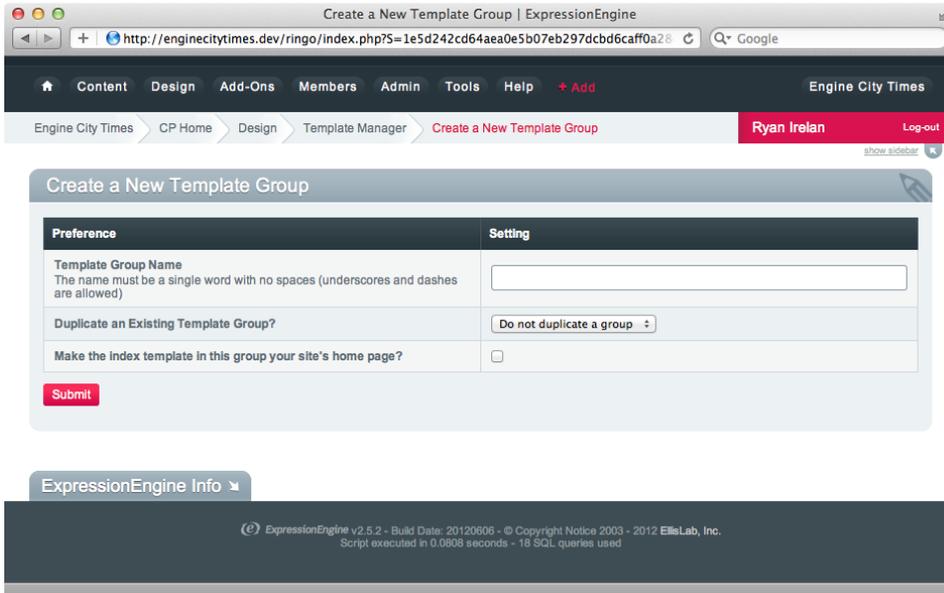


Figure 8—Creating a new template group

want ExpressionEngine to create an empty template.¹ Click Create to create the template.

Adding Our Contact Information

After creating the template, ExpressionEngine returns us to the Template Management page. In the list of templates to the right, find `my_info`, and click the name. You'll notice the document icon to the left of the template name; this indicates that this is a web page template.

At the template edit page, you should now see a blank template. Click into the large edit area, and let's do something simple to start. Type your name and then save the template by clicking Update. To view it in a web browser, click the red View Rendered Template button (at the top right). You should see something like this screenshot:

1. You do have the option of creating a new template by cloning an existing template, but in this instance we want to create an empty template.

The Different Types of Templates

In ExpressionEngine you can create six types of templates:

- *Web Page*: Use this type to create templates to display in the web browser. This is the template type that you will use most often.
- *RSS Page*: If you need to create an RSS feed for your site, you should use this template type.
- *CSS*: If you are using ExpressionEngine to store your CSS, you will want to select this template type. ExpressionEngine tags are not parsed in CSS templates.
- *JavaScript*: This template type should be used for storing and outputting JavaScript in ExpressionEngine.
- *Static*: This is a simple template that is not parsed by ExpressionEngine.
- *XML*: Use this type if you want to create and serve an XML document in ExpressionEngine.



Adding HTML to Our Template

Of course, ExpressionEngine templates can do more than just display simple, plain text. You can put in any valid HTML markup you want, and it will render properly in the browser. Let's go ahead and give our plain `my_info` template a little something extra.

But before we add any HTML to our template, let's first put in the proper document header and DOCTYPE. Put the following code in the `my_info` template, replacing the text we had in there. Notice that your name now lives between the body tags.

BasicsofTemplating/header.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
```

```

<title>Contact Information for Ryan Irelan</title>
</head>
<body>
  Ryan Irelan
</body>
</html>

```

Now we can begin adding very simple markup to our template. Let's wrap our name in an h1 tag. We'll also create an h2 tag and place the text "Contact Information" inside it. We also don't want to forget that we need to add our contact information. Something like this:

```
BasicsofTemplating/hello_world_simple.html
```

```

<h1>Ryan Irelan</h1>
<h2>Contact Information</h2>
<p>Please use this information to get in touch with me.</p>
<address>
  123 Expression Street<br />
  Engine City, USA<br />
  01234<br />
  <a href="mailto:yourname@email.com">
  youremail@email.com</a><br />
  111-555-1212
</address>

```

This should all sit between the body tags in the template. After you've added the previous code, save the template, and view it in your browser. You should see something like that in [Figure 9, Viewing our contact information, on page 9](#).

The complete markup should now look like this:

```
BasicsofTemplating/hello_world_simple_complete.html
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Contact Information for Ryan Irelan</title>
  </head>
  <body>
    <h1>Ryan Irelan</h1>
    <h2>Contact Information</h2>
    <p>Please use this information to get in touch
      with me.</p>
    <address>
      123 Expression Street<br />
      Engine City, USA<br />
      01234<br />
      <a href="mailto:yourname@email.com">
      youremail@email.com</a><br />

```



Figure 9—Viewing our contact information

```

111-555-1212
</address>
</body>
</html>

```

Congratulations, you just created your first ExpressionEngine web page! Before we go any further, a brief discussion on how ExpressionEngine makes content render in the browser is in order.

2.2 Dissecting ExpressionEngine URLs

When we went to view our new web page in the browser, you might have noticed that the URL had four different parts to it (see [Figure 10, *The building blocks of ExpressionEngine URLs, on page 10*](#)). ExpressionEngine uses the URL to determine which template (and content) to show. Let's break down the URL shown in the figure.

The first part of the URL is the base URL or the domain where your website is located. The second part is `index.php`, which is the file that processes all ExpressionEngine templates.² The third part is the ExpressionEngine template group (`contact`) we created earlier. Lastly, the fourth part is the `my_info` template that we created inside the template group.

It's also possible to view the index template of a template group (remember the one that ExpressionEngine created for us?) by just appending the template group to the base URL.

2. There are several ways to remove or change the `index.php` file from the URL. You can learn more on the ExpressionEngine website.

http://mysite.com/index.php/contact/my_info

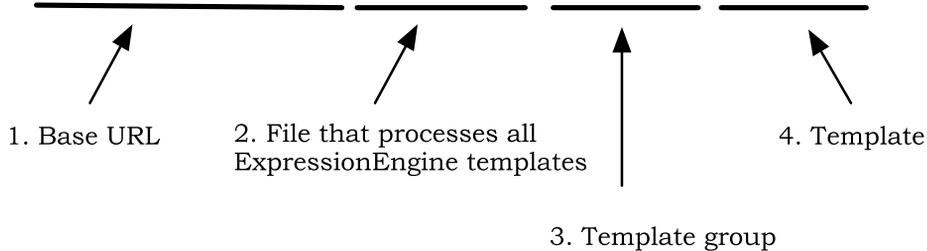


Figure 10—The building blocks of ExpressionEngine URLs

For example, <http://mysite.com/index.php/contact> will render the same page as <http://mysite.com/index.php/contact/index>. If there is no template specified, ExpressionEngine will just render the index template of that group. That's enough about the URLs; let's move forward and start displaying some dynamic content with our new template, shall we?

2.3 Displaying Dynamic Content

Up to this point we've displayed only static content—what we put in the template displayed exactly like that in the browser. Let's add something dynamic to our template. We want our contact information page to be timely, so using an ExpressionEngine tag,³ let's show the current date.

Go ahead and add the following code to your template. A good place is right below the h2 tags.

`BasicsofTemplating/current_date.html`

```
<p>Today is {current_time format="%D, %F %d, %Y"}</p>
```

Save the template, and reload the web page in your browser. You should now see something like [Figure 11, Dynamically displaying the current date and time, on page 11.](#)

The tag `{current_time format="%D, %F %d, %Y"}` tells ExpressionEngine to display the current date in the format we specified using the format parameter. If you've used other templating or programming languages, this syntax for displaying dates and times will look familiar to you.

3. We'll talk more about tags later in this chapter in [Section 2.5, Bringing Life to Templates with ExpressionEngine Tags, on page ?](#).



Figure 11—Dynamically displaying the current date and time

2.4 Working with ExpressionEngine Variables

Let's continue building on our `my_info` template by adding some more of our own content, but this time we'll use ExpressionEngine variables to make it display. First we'll see how the variables work, and then we will learn what they do and how we can use them in our website development with ExpressionEngine.

Our First Variable

We want to be sure that the friends and business associates who visit the page also know where to go to read my weblog. Because it's important that the reader not miss this link, we want to include it twice—once at the top of the page and once at the bottom. However, we don't want to repeat the same markup twice because down the road if we ever want to edit it and change the URL, we'll have to change the markup in two different places. This is where the variables in ExpressionEngine come in handy.

ExpressionEngine provides a way for us to assign variables right inside the template. We're going to create the variable using the `preload_replace` tag. This tag has three parts (see [Figure 12, Assigning a preload replacement variable, on page 12](#)). The first part defines what type of tag it is and indicates to ExpressionEngine that this tag will create a new instance of preload replacement. The second part is the variable name, and the third part is the value we're going to assign to the variable.

BasicsofTemplating/ee_variable_assign.php

```
{preload_replace:my_blog="

```

How Preload Replacement Variables Work

Preload replacement variables are not much different from the variables you'll encounter in most programming languages. They are very simple to learn and a powerful way to manage content—both dynamic and static—in your template.

If you notice that your variable renders in the browser like `{my_variable_name}`, this probably means ExpressionEngine did not recognize it, and you have not assigned it correctly. Check your syntax and tag structure for any errors.

Finally, preload replacement variables can contain more than just static content. As you learn more about ExpressionEngine, you'll see that you can assign the variable different dynamic content depending on the page that is rendered. It's very powerful!

```
{preload_replace:my_blog="http://ryanirelan.com"}
```

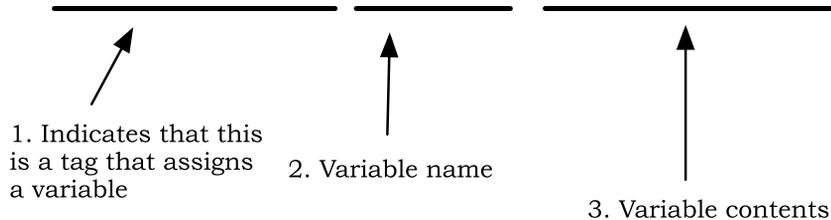


Figure 12—Assigning a preload replacement variable

The variable name is `my_blog`, and the value of that variable is some HTML to create the link to my blog. Place the preload replacement tag at the very top of the `my_info` template. It's a good idea to keep your preload replacement tags altogether in one place. Most people prefer the top of the template; it helps keep the template more organized.

Save the template, and then reload the web page.

Oops! What happened? Where are the links to the blog? They're not there because we haven't yet placed the variable itself in the template. We only assigned the blog link to the variable but didn't tell ExpressionEngine to display it.

We add variables to the template by surrounding the variable name from the `preload_replace` tag with curly braces. This is what our variable will look like: `{my_blog}`.

Place this new variable at the top of the page, just above the h1, and then again at the bottom of the page below the last p tag. To make our link display nicely, wrap the variable in paragraph tags.

BasicsofTemplating/hello_world_with_variables.php

```
{preload_replace:my_blog="<a href="http://ryaniirelan.com">
Visit my blog</a>"}
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Contact Information for Ryan Irelan</title>
  </head>
  <body>
    <p>{my_blog}</p>
    <h1>Ryan Irelan</h1>
    <h2>Contact Information</h2>
    <p>Today is {current_time format="%D, %F %d,
    %Y"}</p>
    <p>Please use this information to get in
    touch with me.</p>
    <address>
      123 Expression Street<br />
      Engine City, USA<br />
      01234<br />
      <a href="mailto:yourname@email.com">
      youremail@email.com</a><br />
      111-555-1212
    </address>

    <code:bold><p>{my_blog}</p></code:bold>
  </body>
</html>
```

After you make these changes, save the template. Reload the page in your browser, and you should see the links to the blog at both the top and bottom of the page, similar to [Figure 13, Linking my blog using preload replacement variables, on page 15.](#)

If you ever want to edit these links to my blog to point to a new URL, it's just a matter of editing in one place: at the top of the template page where we assigned the preload replacement variable. Go ahead and give it a try. Change the variable to one of your favorite websites, save the template, and reload the page.

Up to this point we've created our contact page template, dynamically displayed the date using the `current_time` variable, and learned how to use preload replacement variables. We're now ready to move forward and implement ExpressionEngine tags.

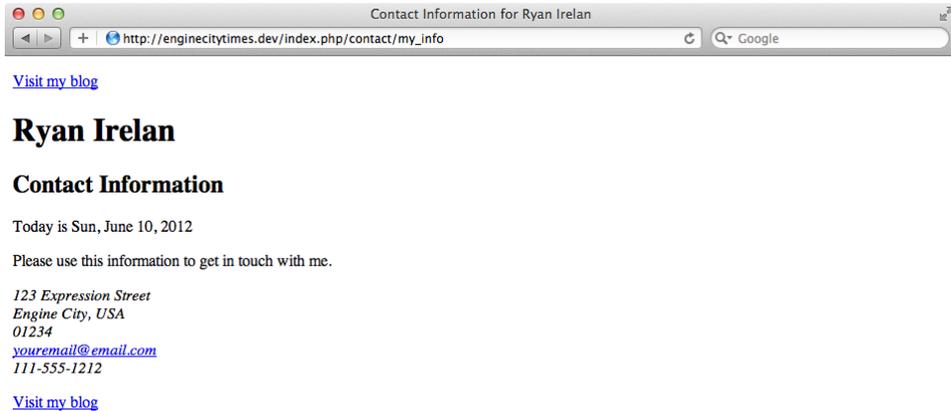


Figure 13—Linking my blog using preload replacement variables
