

# 2

## Making DRY Chunks of the Main Template

### Are you down with DRY?

No, I'm not asking the state of your bath towels. DRY is an acronym for "Don't Repeat Yourself" and the concept comes from programmers talking about their code. The idea is you don't want to repeat code - you only want to store it in one place and then just point to it where needed. The hope is that by doing so you'll make future maintenance and updates more efficient by only having to edit the target code in one place.

ExpressionEngine 2 has a number of ways to help you to be DRY. Picture your collection of Tupperware - with a variety of sizes and shapes of food storage containers. EE has a similar collection - but EE's containers are for storing different chunks of code. EE has:

- User Defined Global Variables
- Snippets
- Embedded Templates

In this chapter, we start what I've termed "chunking" of the template, where we take parts - chunks - of the HTML code from the parent template and save them in one of EE's code storage containers.

Let's take a look at our options.

### User Defined Global Variables

ExpressionEngine has its own collection of Global Variables. We already used one, `{site_url}`, in the last chapter. In addition to the existing variables, EE also allows you to define your own Global Variables. Think of these as the smallest and simplest in EE's collection of code storage containers.

User Defined Global Variables:

- Are best for bits of static code.
- Aren't run through EE's template engine so can't contain dynamic EE tags or PHP, etc.

- Are evaluated last as EE assembles a page so can't be used to influence or set parameters on other EE tags.
- Can't be named the same as existing Global Variables.
- More in the EE User Guide:  
[http://expressionengine.com/user\\_guide/templates/globals/user\\_defined.html](http://expressionengine.com/user_guide/templates/globals/user_defined.html)

## Snippets

Snippets are new to ExpressionEngine 2. Sort of. If you used EE 1 and came across the first-party add-on module called Fresh Variables, then you'll be familiar with Snippets. Snippets are Fresh Variables brought in to the default EE install, given a more accessible place in the Control Panel and renamed.

Snippets are the middle-size code storage container and:

- Are evaluated (parsed) early in putting an EE-generated page together. This means you can use a Snippet to influence or set parameters on other EE tags and templates.
- Are treated as if that code was directly in the template being parsed.
- Can contain EE code, PHP - either full functions or parts and pieces.
- Are not considered full templates by EE - so require less overhead to pull them in.
- Don't have the individual template settings - caching, access, PHP parsing etc. The settings of the template that the Snippet is merged into would dictate these settings.
- Can't be organized in groups.
- Have a fixed-height editing interface - which might make large Snippets tough to edit (an exception to this would be if you were using a WebKit browser, which lets you expand and contract textareas).
- Can't be saved as external files and edited with an external text editor.
- Can't be rendered individually for previewing results.
- More in the EE User Guide:  
[http://www.expressionengine.com/user\\_guide/templates/globals/snippets.html](http://www.expressionengine.com/user_guide/templates/globals/snippets.html)

## Embedded Templates

Embedded templates are the largest of EE's code storage containers. Embedded templates have the most flexibility, but that flexibility comes at the price of greater system overhead.

Embedded templates:

- Are separate templates - with their own settings for caching, access, PHP etc.
- Are available at a specific URL (unless you've set them to be hidden - which I'll cover in a moment).
- Require EE to do more work. In the words of the EE User Guide: *"embedded templates are not included in the parent template, but rather added to them after the fact, using a separate query and full page parsing resources for each template."*
- Have the ability to receive data from the parent (or calling) template via Embed Variables for Embedded Templates.
- Can be organized into groups or stored with the parent templates that use them.
- Can be nested (embedded several layers deep).
- Can be hidden by naming them with a period first (.template\_name).
- Can be saved as external files and edited with a text editor.
- Have an adjustable-height text area for viewing more code.
- Can be rendered individually for previewing results.
- More in the EE User Guide:  
[http://expressionengine.com/user\\_guide/templates/embedding\\_templates.html](http://expressionengine.com/user_guide/templates/embedding_templates.html)

## Which One to Use?

In practice, I seldom use Global Variables. They are limited in functionality and if they store things I want clients to be able to edit then I have to give clients access to the Design area of the Control Panel, which I am loath to do. The EE User Guide mentions using Global Variables for things like Copyright Dates; but even for that I'd rather build it dynamically using the system time or have the content stored in a place that my clients can access it from the Content area of the Control Panel (I'll show you how in Chapter 6).

Since Snippets are new with EE 2, I have less experience with them. My intent is to use them when I don't need the full power of an Embedded Template. For me, the main reason to move from a Snippet to an Embedded Template is if I need to pass a variable between the parent and Embedded Template (no worries if I just spoke gibberish - we'll cover Embed variables for Embedded Templates soon as well).

One thing I have noted about Snippets is unless you come up with a naming schema, from a template code perspective, they are indecipherable from Global Variables. For example, `{mikes_new_chunk}` could be either a Global Variable or a Snippet, but from the template code you can't tell which. If you needed to change its contents, you'd be off on an Easter Egg hunt to figure out where it was coming from. For that reason, I will be implementing a naming schema. I'll preface Global Variables with `gv_` and Snippets with `snp_`.

Examples:

`{gv_copyright}` if it's a Global Variable

`{snp_copyright}` if it's a Snippet.

Embedded templates are always obvious in code because they are called in using an `{embed="template_group/template_name"}` statement. You can choose a different schema for your Global Variables and Snippets but I'd strongly recommend adopting some sort of naming convention as early in your development process as possible.

## What to Chunk?

Now that we know what containers we have available to store our chunks in, we should step back and take a look at the current page and what we're intending to do. Sometimes I can get carried away with the chunking and end up with a site that's more complex than it needs to be. The big question is, what elements (both from a visual and coding perspective) would benefit from being chunked?

Starting visually, the obvious candidates are anything that's sure to be repeated as we move through the site:

- The company name / logo
- The main navigation
- The search box
- The footer

We can add the Latest News and Latest Products blocks to that list and then promote them on multiple pages throughout the site without having to re-code

them each time. The Register Now box is interesting, but the way it's positioned in the middle of other content will make it difficult to embed. Let's just lose that box as there will be no need for it on this site.

Looking at the source code on the HTML, we'll also grab the HTML header including meta tags, the link to the stylesheet, and page title as those elements also repeat page to page.

I know - the page title needs to be different with each page. In Chapter 9, I'll show you how to keep the page title stored in the Embedded Template while keeping each page title unique.

## Chunking Plan

You'll find a copy of the index HTML with code-level comments added to denote where the code will be chunked up in the Companion Files at: `chapter_2/index_chunking_guide.html`. You can also download it from Train-ee: [http://train-ee.com/companion\\_files/sb/chapter\\_2/index\\_chunking\\_guide.html](http://train-ee.com/companion_files/sb/chapter_2/index_chunking_guide.html)

If you open the companion file in a text or HTML editor and peruse the code, you'll see a large amount of content that isn't being pulled out and embedded. That's the core content of the page and, in the next chapter of this series, we will pull that content out of the template and store it in an EE Channel.

I'm sure different EE developers do this differently. I prefer to keep the skeleton of the page's essential structure in the parent template. I don't like having `<div>` tags getting opened in one template and closed in another because doing so can make it hard to track down mis-matches or validation errors.

I will sometimes break high level tags, like the `<body>` tag between two different Embedded Templates, as long as the design doesn't call for different classes or ID's to be applied to it. The Design Template that we are using here is an example of this. By putting the opening `<body>` and `<div id="content">` tags in one chunk container (`html_header`) and closing them in another `{snp_footer}` the resulting parent template is much cleaner, easier to read and a bit more DRY.

Here's another puzzle: how small can your chunks be? For example, look at the layout of the Design Template we're working with here:



**Figure 13: Common Header Design Elements**

There is a logo, main navigation, some intro text and a search box. You could create separate chunks for each of these - and in the past I would have done so. There's just something fun about breaking that code out and having little stored chunks for each design element on the page. Over time, however, I've realized that if those elements are never used independently from each other - or in a different order - then there is no benefit to storing them separately.

Actually, there is a penalty in that the more chunks EE has to use to assemble the page, be it a Global Variable, Snippet or Embedded Template, the more system resources, queries and processing will be required.

So before you get carried away splitting up the design into smaller chunks than necessary ask yourself questions like: "Will I ever have pages without the intro text, or without the search box?" If the answer is "no", then don't chunk those pieces separately.

In our case every page will always have all of these header elements, so I'll create one Embedded Template to store them all.

Our plan has both Snippets and Embedded templates - so I'll show you how to do one of each. The reasons for choosing to put some of them into Embedded Template rather than Snippets will become clear as we continue to implement the entire site.

## Create a New Template Group

I like to keep all my Embedded Templates in their own group, so that my main groups only contain parent templates. Let's get a new embeds Template Group created. Here are the steps:

1. Enter the EE Control Panel.
2. Navigate to the Template Manager: Design > Templates > Template Manager.
3. Click the New Group button on the upper left portion of the screen.

4. Choose a name. I've called mine "**embeds**" - if you do the same it'll make following along with rest of the book much easier as this name will get used often in the code.
5. Click Submit.

## On to the Chunking!

I like to do this step incrementally - as it's easier to find and correct any mistakes if I don't change too much at once. My basic workflow is:

- With the full index template open, copy the section of code to chunk out.
- Create a new template in my "embeds" group or create the new Snippet.
- Paste in the section of code that I want to save in the new chunk container.
- Save it.
- Return to the parent site/index template in edit mode.
- Select the same portion of code that I copied earlier.
- Replace it either with the new Snippet variable or an embed statement to pull in the Embedded Template.
- Save the parent template by clicking Update.
- Click View Rendered Template button to ensure that the parent template remains unchanged visually. Sometimes I've grabbed one line of code too many or too little and this is where I catch those mistakes.
- Repeat the above process for each section of code in the chunking plan file.

Let's do one each of an Embedded Template and Snippet.

## Chunking an Embedded Template

The first chunk in our plan is the top of the HTML file from the doctype through the opening content div tag. The reason I chose to store this in an Embedded Template rather than a Snippet is the title tag.

For good search engine optimization (SEO), I want page-specific titles, and will want to set the contents of the title tag based on where in the site this template is being used. With an Embedded Template, I can both centralize this code and change bits of it depending on where I use it. With a Snippet, I can't do that.

We'll just chunk it out now and cover implementing the context-sensitive title tag in Chapter 9.

### **To create the Embedded Template:**

1. Navigate to the Template Manager: Design > Templates > Template Manager.
2. Open site/index for editing.
3. Select the code starting with the `<!DOCTYPE` down to include the opening `<body>` and `<div id="content">`.
4. Copy it to the clipboard.
5. Create a new template in the embeds Template Group.
6. Name it **"html\_header"**.
7. Click Create and Edit.
8. Paste in the code from the clipboard. Your template should consist of the following code:

Template Name: embeds/html\_header

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head>
  <meta http-equiv="content-type"
content="text/html; charset=iso-8859-2" />
  <meta name="author" content="free-css-templates.com" />
  <link rel='stylesheet' type='text/css' media='screen'
href="{stylesheet='site/stylesheet'}" />
  <title>Template1</title>
</head>
<body>
  <div id="content">
```

---

Companion File: [chapter 2/embeds/html\\_header.txt](#)

9. Click Update and Finished.
10. Return to the site/index template.
11. Select the same section of code.



12. Replace it with `{embed="embeds/html_header"}`.
13. Save the template.
14. Render the page and make sure to refresh your browser.
15. Make sure nothing changes visually. If it does, double check your code.

## Chunking a Snippet

The next chunk in our chunking plan is a Snippet. It will hold the site's footer. This can't be a Global Variable because I intend to use EE tags in it so that the content can be user-editable. However, the content will always be the same (never context-sensitive) so it doesn't need to be a full Embedded Template. The middle-size Snippets chunk container is not too big, not too small, but just right.

To create the Snippet:

1. Navigate to the Template Manager: Design > Templates > Template Manager.
2. Open site/index for editing.
3. Select the code starting with `<div id="footer">` through the closing `</html>` tag.
4. Copy it to the clipboard.
5. Navigate to the Snippets section of the Control Panel: Design > Templates > Snippets.
6. Click Create a New Snippet.
7. Name it **"snp\_footer"**.
8. Paste in the code from the clipboard. Your Snippet should contain the following code:

Snippet Name: snp\_footer

---

```
<div id="footer">
  <p class="right">
    Copyright &copy; 2006 Your Company, Design: <a
href="http://www.free-css-templates.com" title="Designed by
free-css-templates.com">free-css-templates.com</a>
  </p>
  <p>
    <a href="#">RSS Feed</a> &middot; <a
href="#">Contact</a> &middot; <a href="#">Accessibility</a>
&middot; <a href="#">Products</a> &middot; <a
```

```
href="#">Disclaimer</a> &middot; <a
href="http://jigsaw.w3.org/css-
validator/check/referer">CSS</a> and <a
href="http://validator.w3.org/check?uri=referer">XHTML</a><br
/>
    </p>
</div>
</div>
</body>
</html>
```

Companion File: [chapter 2/snippets/snp\\_footer.txt](#)

9. Click Update and Finished.
10. Return to the site/index template.
11. Select the same section of code.
12. Replace it with {snp\_footer}.
13. Save the template.
14. Render the page and refresh your browser.
15. Make sure nothing changed visually. If something has go back and review your code (see the Not Working? section at the end of this chapter for troubleshooting tips).

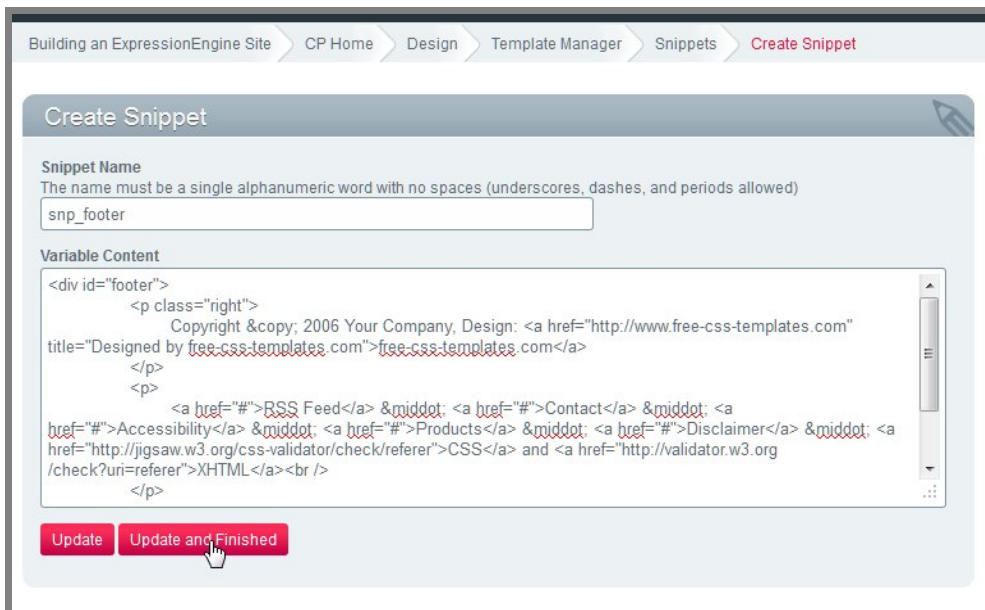


Figure 14: Creating a Snippet

## Lather, Rinse, Repeat

After following those two examples, you should be able to chunk out the rest of the site/index template. Use the chunking guide as your reference and create the following:

Template Name: `embeds/page_header`

---

```

<div id="title">
  <h1>Your <span class="green">Company</span></h1>
  <span id="slogan">The Best Slogan here</span>
</div>

<div id="menu">
  <div class="submit">
    <ul>
      <li><a href="#" ><span>Home</span></a></li>
      <li><a href="#"><span>Products</span></a></li>
      <li><a
href="#"><span>Affiliates</span></a></li>
      <li><a href="#" class="selected"><span>About
us</span></a></li>
      <li><a href="#"><span>Contact</span></a></li>
    </ul>
  </div>
</div>

<div id="subheader">
  <div class="rside">
    <div class="padding">
      <div id="search">
        <form action="#" method="get">
          <p><input type="text" name="search"
size="20" maxlength="250" class="text" value="" />
          <input type="submit" value=""
class="btn1" /></p>
        </form>
      </div>
    </div>
  </div>
  <div class="lside">
    <div class="padding">
      <strong>Lorem ipsum</strong> dolor sit amet,
consectetuer adipiscing elit. Morbi elementum, ipsum nec
auctor condimentum, magna odio aliquet elit, non bibendum
lorem ligula at nisi. <a href="#" title="">Proin sapien</a>.
    </div>
  </div>
</div>

```

---

Companion File: [chapter\\_2/embeds/page\\_header.txt](chapter_2/embeds/page_header.txt)

Snippet Name: `snp_latest_news`

---

```
<h3>Latest news</h3>
<div class="lcontent">
  <p><strong>Lorem ipsum</strong> dolor sit amet,
consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat. Quis
nostrud <a href="#" title="">exerci</a> tation ullamcorper
suscipit. <br /><a href="#" title="">Read more...</a></p>
</div>
```

---

Companion File: [chapter 2/snippets/snp\\_latest\\_news.txt](chapter 2/snippets/snp_latest_news.txt)

Snippet Name: `snp_latest_products`

---

```
<h3>Latest Products</h3>
<div class="lcontent">
  <p>
    
    <strong>Lorem ipsum</strong> dolor sit amet,
consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut
wisi enim ad <a href="#" title="">minim veniam</a>.
  </p>
  <p>
    
    Quis nostrud exerci tation <strong>ullamcorper
suscipit</strong> lobortis nisl ut aliquip ex ea commodo
consequat. Dolor in hendrerit in <a href="#"
title="">vulputate</a> velit esse molestie consequat.
  </p>
  <p>
    
    <strong>Lorem ipsum</strong> dolor sit amet,
consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut
wisi enim ad <a href="#" title="">minim veniam</a>.
  </p>
</div>
```

---

Companion File: [chapter 2/snippets/snp\\_latest\\_products.txt](chapter 2/snippets/snp_latest_products.txt)

## After Chunking

You now should have a drastically simplified site/index template with all the repeatable elements chunked out and referenced in the parent template using embed statements and Snippet variables. Here's what I have:

Template Name: site/index

---

```
{embed="embeds/html_header"}
  {embed="embeds/page_header"}
  <div id="maincontent">
    <div id="right_side">
      {snp_latest_news}
      {snp_latest_products}
    </div>
    <div id="left_side">
      <h2 class="underline">About us</h2>
      
      <p><strong>Lorem ipsum dolor</strong> sit amet,
consectetur adipiscing elit. Morbi elementum, ipsum nec
auctor condimentum, magna odio aliquet elit, non bibendum
lorem ligula at nisi. Proin sapien. In hac <a href="#"
title="">habitasse platea dictumst</a>. Nunc non magna. Duis
pharetra laoreet dolor. Mauris tristique interdum turpis.
Praesent sed pede. Aenean porttitor ultrices tortor. Aenean a
purus. Praesent eu sapien.</p>
      <p>Nulla posuere, sapien sit amet euismod
suscipit, magna diam sodales magna, nec semper eros tortor et
nulla. Nunc nec mauris. Aenean tincidunt. Nam sem. Etiam quis
justo. Donec pellentesque pellentesque tellus. <strong>Integer
bibendum</strong> massa sed diam. In in elit. Ut blandit arcu
sed ligula. Aenean interdum. Suspendisse adipiscing suscipit
nisl. In non leo et justo posuere placerat. In tincidunt
cursus turpis. Maecenas suscipit tempor ante. Vestibulum
pretium aliquam metus.</p>
      <blockquote><p class="mpnull">Sapien sit amet
euismod suscipit, magna diam sodales magna, nec semper eros
tortor et nulla. Nunc nec mauris. Aenean tincidunt. Nam sem.
Etiam quis justo. Donec pellentesque pellentesque tellus.
<strong>Integer bibendum</strong> massa sed diam. In in
elit.</p></blockquote>
      <p>Nulla posuere, sapien sit amet euismod
suscipit, magna diam sodales magna, nec semper eros tortor et
nulla. Nunc nec mauris. Aenean tincidunt. Nam sem. Etiam quis
justo. Donec pellentesque pellentesque tellus. <strong>Integer
bibendum</strong> massa sed diam. </p>
```

```
        <div class="textbox">
            <h2><a href="#" title="">Register
now!</a></h2>
            <p>Nulla posuere, sapien sit amet euismod
suscipit, magna diam sodales magna, nec semper eros tortor et
nulla. Nunc nec mauris.</p>
            <p>Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Morbi <a href="#"
title="">elementum</a>, ipsum nec auctor condimentum, magna
odio aliquet elit, non bibendum lorem ligula        at nisi.
Proin sapien. In hac habitasse platea dictumst.</p>
        </div>
        <h2>Our Assets</h2>
        <p>In hac habitasse platea dictumst. Nunc non
magna. Duis pharetra laoreet dolor. Mauris tristique interdum
turpis. Praesent sed pede. Aenean porttitor ultrices tortor.
Aenean a purus. Praesent eu sapien.</p>
        <p>In hac habitasse platea dictumst. Nunc non
magna. Duis pharetra laoreet dolor.</p>
        <ul>
            <li>In hac <strong>habitasse</strong>
platea dictumst</li>
            <li>Lorem ipsum dolor <a href="#"
title="">sit amet</a></li>
            <li>habitasse platea dictumst</li>
            <li>sapien sit amet euismod</li>
        </ul>
    </div>
</div>
{snp_footer}
```

---

Companion File: [chapter 2/index after chunking.txt](#)

## The Results

Once you have all these template changes made, you should be able to load the rendered version of the site/index template and have it appear exactly as it did at the end of Chapter 1.

# Your Company

The Best Slogan here

Search

Home
Products
Affiliates
About us
Contact

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit. Morbi elementum, ipsum nec auctor condimentum, magna odio aliquet elit, non bibendum lorem ligula at nisi. [Proin sapien](#).

### About us

**Lorem ipsum dolor** sit amet, consectetur adipiscing elit. Morbi elementum, ipsum nec auctor condimentum, magna odio aliquet elit, non bibendum lorem ligula at nisi. [Proin sapien](#). In hac [habitasse platea dictumst](#). Nunc non magna. Duis pharetra laoreet dolor. Mauris tristique interdum turpis. Praesent sed pede. Aenean porttitor ultrices tortor. Aenean a purus. Praesent eu sapien.

Nulla posuere, sapien sit amet euismod suscipit, magna diam sodales magna, nec semper eros tortor et nulla. Nunc nec mauris. Aenean tincidunt. Nam sem. Etiam quis justo. Donec pellentesque pellentesque tellus. **Integer bibendum** massa sed diam. In in leo et justo posuere placerat. In tincidunt cursus turpis. Maecenas suscipit tempor ante. Vestibulum pretium aliquam metus.

Sapien sit amet euismod suscipit, magna diam sodales magna, nec semper eros tortor et nulla. Nunc nec mauris. Aenean tincidunt. Nam sem. Etiam quis justo. Donec pellentesque pellentesque tellus. **Integer bibendum** massa sed diam. In in elit.

Nulla posuere, sapien sit amet euismod suscipit, magna diam sodales magna, nec semper eros tortor et nulla. Nunc nec mauris. Aenean tincidunt. Nam sem. Etiam quis justo. Donec pellentesque pellentesque tellus. **Integer bibendum** massa sed diam.

#### Register now!

Nulla posuere, sapien sit amet euismod suscipit, magna diam sodales magna, nec semper eros tortor et nulla. Nunc nec mauris.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi [elementum](#), ipsum nec auctor condimentum, magna odio aliquet elit, non bibendum lorem ligula at nisi. Proin sapien. In hac habitasse platea dictumst.

### Latest news

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Quis nostrud [exerci](#) tation ullamcorper suscipit. [Read more...](#)

### Latest Products

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad [minim veniam](#).

Quis nostrud exerci tation **ullamcorper suscipit** lobortis nisl ut aliquip ex ea commodo consequat. Dolor in hendrerit in [vulputate](#) velit esse molestie consequat.

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad [minim veniam](#).

[RSS Feed](#) · [Contact](#) · [Accessibility](#) · [Products](#) · [Disclaimer](#) · [CSS](#) and [XHTML](#)

Copyright © 2006 Your Company, Design: [free-css-templates.com](#)

Figure 15: Home Page after Chunking

59

## Not Working?

Here are some things to check if your site/index isn't rendering correctly:

- Make sure you've used braces `{ }` and not parenthesis `( )` around your Snippet name or Embed statement.
- Make sure every opening brace has a matching closing brace.
- Check for singular vs. plurals in your code (`{snp_latest_products}` and not `{snp_latest_product}`).
- Check for misspellings - either in the chunk container names or in the parent template code.
- EE is case sensitive - if you named your Embedded Template `Page_Header` but then tried to call it in using `{embed="embeds/page_header"}`, it won't work.
- Check your rendered source code to make sure you have matching div tag pairs, etc.
- If you copied and pasted any code in from another source, make sure you didn't get smart quotes. If so, replace them with straight quotes.