Basic Templating Tutorial
by Conor Gilmer

On a website composed of different pages, often it is only a relatively small portion of content of the webpage that changes, rather than re-coding the common sections of all these pages which normally deal with style, and layout, templating is used .
Templating is a way of ensuring consistency through out a website by using the same code for certain elements. Due to this templating formula it normally makes adding a new page easier since you generally only have to concentrate on the changes and coding of the new page.

For this tutorial I am going to use a simple example of a website using PHP with common elements such as a Header, a Navigation and footer on each page. For this you will have to set up files for the common elements and files for each page based on the basic page template.

Setting up Common elements

The *header.php* file contains the top of the webpage.
The *navbar.php* file contains the top of the webpage including the title, and the menu items i.e. In this example there is Home, Hobbies, Webelevate and Contact pages.
The *footer.php* file contains the end of the webpage.

Template for a Basic Page

A basic page consists of (e.g. test.php)

```
<?php include 'includes/header.php'; ?>
<?php include 'includes/navbar.php'; ?>
<!-- Webapge content goes here -->
<div class="jumbotron">
        <h2>Heading</h2>
        <p class="lead">Enter any text you want in here.</p>
      </div>
<!-- end of individual webpage content-->
<?php include 'includes/footer.php'; ?>
```

Adding a New page

To add a page to this site you would have to add the page formatted like test.php above above and then include it in the *navbar.php* file below

```
<div class="header"><h3 class="text-muted">Conor Gilmer </h3>
  <ul class="nav nav-pills pull-right">
    <li class="active"><a href="index.php">Home</a></li>
    <li class="active"><a href="hobbies.php">Hobbies</a></li>
    <li class="active"><a href="test.php">Test</a></li>
    <li class="active"><a href="webelevate.php">Webelevate</a></li>
    <li class="active"><a href="contact.php">Contact</a></li>
   </ul>
 </div>
```

Adding Code to all pages

This can be achieved relatively easily by adding it to the footer or header, e.g. in this example metrics code for google or piwik could be added to the footer file.

Conclusion

Even basic templating as outlined in this tutorial has benefits by insuring a consistency of layout and stype on webpages. Code is reused so you don't have to reenter when adding pages. There are more advanced systems of templating e.g. template engines, which render and format pages programmed in a templating language but which use the same basic ideas of tempating.

Appendix

The example I use here is located at http://site101.webelevate.net/conor/mysite/

Header.php

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="shortcut icon" href="/favicon.ico">

    <title>My Site simple templates</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.css" rel="stylesheet">
    <link href="css/jumbotron-narrow.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
      <script src="js/html5shiv.min.js"></script>
      <script src="js/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="container">
```

Footer.php

```
      <div class="footer">
        <p>&copy; Conor Gilmer 2013<br>
        <em><a
href="mailto:conor.gilmer@gmail.com">conor.gilmer@gmail.com</a></e
m></p>
      </div>
    </div>

    <script src="js/jquery.js"></script>
    <script src="js/bootstrap.min.js"></script>
<!--Google / Piwik analytics code can go here -->
  </body>
</html>
```