COMP2203: Application Scripting

Session DA5

Making Life Easier with a Page Template

David Argles

Electronics & Computer Science
University of Southampton

Haven Consulting
david@argles.org

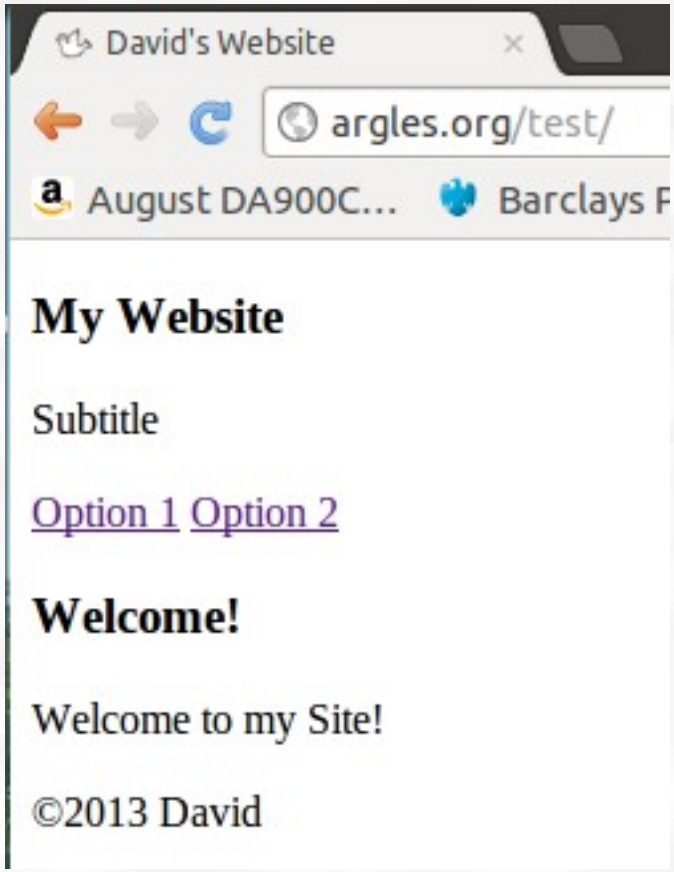
Session DA5 Outline:

- From last time...
- Becoming a DRY programmer
- Creating a page template
- Using our template
- Adding comments

Haven
Consulting
david@argles.org

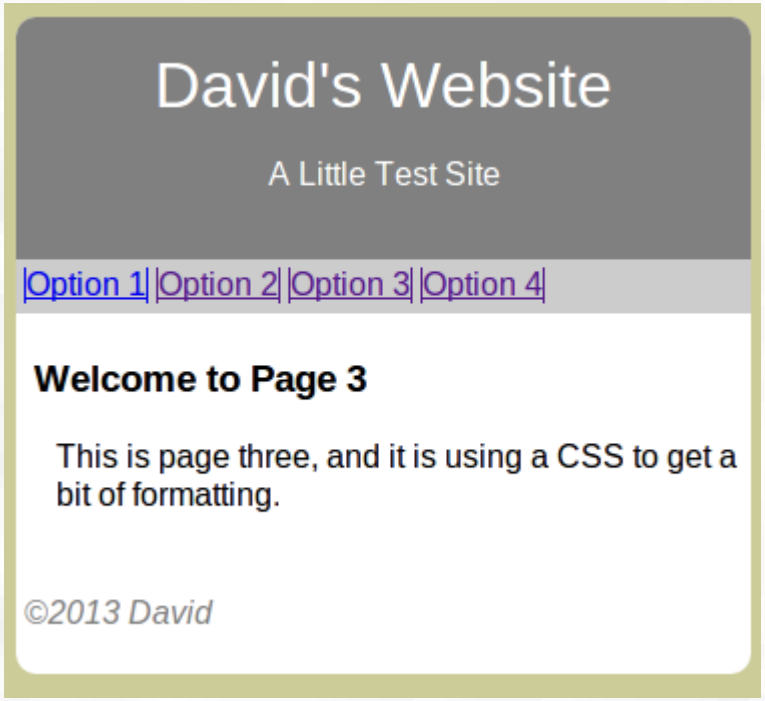
We had: an HTML5 Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>David's Website</title>
  </head>

  <body>
   <container>
    <header>
      <h3>My Website</h3>
      <p>Subtitle</p>
    </header>

    <nav>
      <a href="">Option 1</a>
      <a href="">Option 2</a>
    </nav>

    <section>
      <h3>Welcome!</h3>
      <p>Welcome to my Site!</p>
    </section>

    <footer>
      <p>&copy;2013 David</p>
    </footer>
   </container>
  </body>
</html>
```



We added: some CSS

```
...comments...
body{font-family:verdana,arial,sans-serif;...etc...}
container{...some stuff...}
header{
  background-color:gray;
  color:white;
  padding:1em;
  text-align:center;}
header h{font-size:2em;padding:5px;}
nav{
  background-color:#ccc;
  padding:0.3em;}
nav a{
  border-left:solid;
  border-right:solid;
  border-width:0.1em;}
section{padding:0.3em;}
section h3{padding-left:5px;}
section p{padding-left:1em;}
footer{
  font-style:italic;
  color:gray;}
...more comments...
```



Becoming a DRY* Programmer

- That's not bad. We've:
  - kept the semantics of the formatting in the HTML page, and
  - removed the definitions of the semantics to a single CSS
- So we can change the look and feel of the entire site by changing a single file (the CSS)

*Don't Repeat Yourself

ECS Electronics & Computer Science University of Southampton

Haven Consulting
david@argles.org

Becoming a DRY Programmer

- But a real site will require lots of HTML pages
- And at the moment, this will mean repeating lots of stuff that doesn't change
  - like the menu options. If these change, we'll have to go through every page, find the right place, and update them consistently, a recipe for disaster :-(
- So let's create a single page template that will apply to every page in our site
- We can do this with (object)PHP

Haven Consulting
david@argles.org

Creating a Page Template

- Let's see what doesn't change from page to page:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css"
href="library/normal.css">
    <title>David's Website</title>
  </head>

  <body>
    <container>
      <header>
        <h3>My Website</h3>
        <p>Subtitle</p>
      </header>
      <nav>
        <a href="index.html">Home</a>
        <a href="page2.html">Page 2</a>
        <a href="page3.html">Page 3</a>
        <a href="page4.html">Page 4</a>
      </nav>
      <section>
        <h3>Welcome!</h3>
        <p>Welcome to my Site!</p>
      </section>

      <footer>
        <p>&copy;2013 David</p>
      </footer>
    </container>
  </body>
</html>
```

Creating a Page Template

- That's a lot that doesn't change!
- We can create a page object that provides all that code for us in one place rather than repeating it in every page
- We *could* follow the DOM and create a page template with a number of child objects...
- ...but why? It's good to follow the KISS* principle, so let's just create a simple page

*Keep It Stupidly Simple

Electronics & Computer Science
University of Southampton

Haven Consulting
david@argles.org

Creating a Page Template

- Look back at the slide with our page code...
- ...it would be good to create a page object that:
  - knows how to stream the first bit of HTML before <section> starts
  - knows how to stream the last bit of HTML after we've finished <section>
- And that's all we need!

Creating a Page Template

- So let's create a page object; we'll do it in PHP and call it htmlPage
- Then we can add two functions, streamTop() and streamBottom()

```
<?php
class htmlPage
{
  public function streamTop()
  {
  }

  public function streamBottom()
  {
  }
}
```


Haven Consulting
david@argles.org

Creating a Page Template

- Now we can add the HTML code we want to stream:

```
<?php
class htmlPage
{
  public function streamTop()
  {
?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css"
        href="library/normal.css">
    <title>David's Website</title>
  </head>

  <body>
    <container>
      <header>
        <h3>My Website</h3>
        <p>Subtitle</p>
      </header>
      <nav>
        <a href="index.html">Home</a>
        <a href="page2.html">Page 2</a>

        <a href="page3.html">Page 3</a>
        <a href="page4.html">Page 4</a>
      </nav>

      <section>
<?php
  }

  public function streamBottom()
  {
?>
      </section>

      <footer>
        <p>&copy;2013 David</p>
      </footer>
    </container>
  </body>
</html>
<?php
  }
}
```


Haven Consulting
david@argles.org

Creating a Page Template

- Notice we can switch back and forth between php and HTML
- So our class definition begins in php
  - <?php ...etc.. (now in php)
- But after we declare the start of streamTop(), we go back into HTML
  - ?> ...(now back in HTML)...
- Then we go back into php to close the function
  - <?php }

Haven Consulting
david@argles.org

Using Our Page Template

- Now we've got a page template, we need to use it in our HTML page
- First we can delete all the code we put into our template:

```
<h3>Welcome!</h3>
<p>Welcome to my Site!</p>
```

Using Our Page Template

- Next we need to tell it to use our template class...

```
<?php
 require("library/htmlPage.php");
?>
      <h3>Welcome!</h3>
      <p>Welcome to my Site!</p>
```

- ..and to create an instance of htmlPage:

```
<?php
  require("library/htmlPage.php");
  $page = new htmlPage();
?>
      <h3>Welcome!</h3>
      <p>Welcome to my Site!</p>
```


Haven Consulting
david@argles.org

Using Our Page Template

- Now all that's left to do is to tell it to stream the top code before our page code, and stream the bottom code after our page code:

```
<?php
  require("library/htmlPage.php");
  $page = new htmlPage();
  $page->streamTop();
?>
      <h3>Welcome!</h3>
      <p>Welcome to my Site!</p>
<?php
  $page->streamBottom();
?>
```


Haven Consulting
david@argles.org

And That's It!

- All we need to do is:
  - save this as index.html, page1.html, page3.html and page4.html
  - update the content in each page to reflect them being different pages
- then enjoy the site!

Ah... Except...

- **Except the pages don't work!**
  - That's because we've called them xxx.html so the server isn't treating them as php pages
  - We need to rename them as xxx.php
- **And now the menu is all wrong!**
  - Ah ha! But we only need to change the menu ONCE in htmlPage! SUCCESS!!

Haven Consulting
david@argles.org

Separation of Concerns

- So now we have:
  - the page style defined ONCE in normal.css
  - the page template defined ONCE in htmlPage.php
  - the individual page content defined ONCE in each xxx.php

Haven Consulting
david@argles.org

Adding Comments

- Our htmlPage class ought to be commented!
- Firstly, the whole file needs comments:

```
/**
 * htmlPage.php provides a basic web page class for our website
 *
 * It defines a class, htmlPage, which has two functions, streamTop()
 * and streamBottom()
 *
 * @author David Argles <d.argles@gmx.com>
 * @version 10-08-2013, 21:58h
 * @copyright 2013 Haven Consulting
 */

/**-----------------------------------------
 *              End of Code
 *-----------------------------------------
 */
```

Adding Comments

- We also need to add comments for the htmlPage class itself:

```
/**
 * htmlPage provides a basic web page class for our website
 *
 * It defines two functions, streamTop() and streamBottom() which
 * stream the necessary HTML code to make up our boilerplate page
 *
 * @param void
 * @return void
 */
```


Haven Consulting
david@argles.org

Adding Comments

- And for the two functions, streamTop() and streamBottom():

```
/**
 * streamTop() streams all the code necessary for our boilerplate
 * HTML page before the main page content
 *
 * @param void
 * @return void
 */
 public function streamTop()
 ...etc...

/**
 * streamBottom() streams all the code necessary for our boilerplate
 * HTML page after the main page content
 *
 * @param void
 * @return void
 */
 public function streamBottom()
 ...etc...
```


Haven Consulting
david@argles.org

Break

There is actually still an annoying little problem in what we've done so far. Anyone spot it?

Haven
Consulting
david@argles.org