

COMP2203: Application Scripting

Session DA-php1

Introducing php

David Argles

Overview

We'll be looking at :

- Why php?
- Creating and including our menu
- Making it work!
- Using variables
- How “echo” works
- Some useful php control structures

Why php?

- **We noticed we have a problem...**
 - We want to Only Do Things Once
 - But we have the menu repeated on every html page
 - So if we add a page to our site, we have to change the menu on every page :-)

Creating a Menu

- So let's create a separate menu file
- We can call it “menu.html”
- And put it in our library directory
- Now we need to load it into our html pages

Including Our Menu

Our page was:

```
<!DOCTYPE html>
<html>
  <head>
    ...etc...
  </head>

  <body>
    <container>
      <header>
        ...etc...
      </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>
        ...etc...
      </section>

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

menu.html becomes:

```
<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>
```

Our page becomes:

```
<!DOCTYPE html>
<html>
  <head>
    ...etc...
  </head>

  <body>
    <container>
      <header>
        ...etc...
      </header>

      <?php include ("library/menu.html"); ?>

      <section>
        ...etc...

      </section>
    </body>
  </html>
```

Note:

- We're including our php session using
 - `<?php {.....} ?>`
- You might come across the shorthand version
 - `<? {.....} ?>`

But DON'T USE IT! (deprecated)

Wait! This Doesn't Work!

- If you make these changes, the php doesn't work!
- That's because we're still calling our page
 - “page3.html”
- The server needs to know it's a php page, so we must call it
 - “page3.php”
- Now it works!
- Actually, if we have a php page that has mostly html in (like this one), we could call it
 - “page3.phtml”

Including Other Things

- **There's other code that's:**
 - the same on every page
 - would need to be changed on every single page if...
 - ...like the `<head>` section (e.g. if we changed the style sheet)
- **It would be nice to include all of the `<head>` section in an include file**

Problem

- **But there's a problem**
 - The <head> section includes the <title> declaration...
 - ...which changes on every page
- **So I need a function which I can call with a parameter which specifies what the <title> of that page should be**

Including a php Function File

Our page was:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css"
href="library/normal.css">
    <title>Page 3</title>
  </head>

  <body>
    <container>
      <header>
        ...etc...
      </header>

      <?php include ("library/menu.html"); ?>

      <section>
        ...etc...

      </body>
    </html>
```

functions.php becomes:

```
<?php
function head($title){
  $output="  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css"
href="library/normal.css">
    <title>".$title."</title>
  </head>";
  return $output;
}
?>
```

Our page becomes:

```
<?php include("functions.php"); ?>
<!DOCTYPE html>
<html>

  <?php echo(head("Page 3")); ?>

  <body>
    <container>
      <header>
        ...etc...

    </html>
```

Notes:

- We're using the php command “echo” to put some text into the HTML output stream
 - Syntax: `echo(“some text”);`
- But you can omit the brackets
 - `echo “some text”;`

Notes:

- **We've also “concatenated” text and variables**
 - `$output = “<head>etc...”.$title.”...etc</head>”;`
- **You can use single or double quotes – but they are not the same.**
 - Single quotes are literal
 - Double quotes interpret, so you could put:
`$output = “<head>etc...$title...etc</head>”;`
- **You can be even lazier...**
 - Instead of `<?php echo $var; ?>`
 - You can put `<? =$var ?>`

Don't Repeat Yourself

- So now I have a way to move repeated code into a separate file
- If I want to change something, I can plan so I only need to do it once, whatever the change
- But I can make things better still...
- ...next week
- But first:

if Statements

- **Basic syntax:**

- `if(some_condition) do_something();`
`else do_something_else();`

- **The long version:**

- `if (condition){`
`...statements...}`
`else{`
`...other statements...}`

- **Actually I prefer this layout:**

- `if (condition)`
`{`
`...statements...`
`}`
`else`
`{`
`...other statements...`
`}`

if Statements

- Follow C-like syntax for the condition, e.g.:
 - `if($value<3)...`
 - `if($value<=7)...`
 - `if($value==93)...`
 - Beware of:
 - `if($value=23)...`
- which sets `$value` to 23!

for Loops

- **Basic syntax:**

- `for($count=0; $count<10; $count++)`
 `do_something;`

- **Long version:**

- `for($count=0; $count<10; $count++)`
 {
 ...some statements...
 }

- **Or the other layout as before**