My software never has bugs. It just develops random features.

Unknown

4
Generating html

4.1 php as dynamic html generator

When using PHP as a scripting language to create dynamic websites, it should generate HTML . The word *dynamic* in the previous sentence means that everytime the page is visited, the script is executed and the content only exists in the working memory of the server.

If your script is crafted well, it should produce valid html. PHP automatically adds the correct headers and sends the output to the browser.

So, to create a dynamic website, you should know HTML (and CSS) as well. HTML is not the subject of this course. If you are not very familiar with it, you won't be able to make a good website, so start studying it. A good place to start is `www.w3schools.com/html`.

4.2 Mixing php and html

In a PHP file you can mix HTML and PHP code.

PHP code needs to be put between `<?php` and `?>`. All the rest in your code is treated as static text and thus interpreted as HTML in the browser.

Look at the example below. We want to display the day of the week every time a user visits our page. All static HTML code is placed outside the PHP starting and ending tags. The result is a dynamic page with only one region changing everytime the page is visited.

For a complete description of the `date()` function, have a look at `www.php.net/manual/en/function.date.php`.

Listing 4.1 Mixing php and html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>is it friday yet?</title>
</head>
<body>
<h1>Is it friday yet?</h1>

<p>Today is <?php echo strtolower(date('l')); ?></p>

</body>
</html>
```

In the next example, we will add more dynamic parts to our page. You will see that you can open php tags and close them as many times as you like.

Another principle in the example below is the fact that there is a seperation between php -logic and html -logic. If you have seen design patterns like Model-View-Controller (MVC, have a look at `http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller`) or uses a framework that implements these patterns, you'll recognise the fact that there is a seperation between programming logic and presentation logic.

It really is a good practise to do this. Seperate you php -logic (your controller and model actions) and your presentation part (view). When creating an advanced dynamic website, you probably will use seperated template files which only contain html and placeholders for the output of variables.

Listing 4.2 Mixing more php and html

```
<?php
// checking the weekday and setting some variables
if (date('w') == 5)
{
        $title = 'yes! it is friday';
        $cssclass = 'yes';
        $text = '<strong>yes</strong>';
}
else
{
        $title = "no! it's not friday yet";
        $cssclass = 'no';
        $text = '<strong>bummer</strong>';
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title><?php echo $title; ?></title>
        <style type="text/css">
                .yes {font-size: 1.2em; color: #dd0000;}
                .no {font-size: 0.9em;}
        </style>
</head>
<body>
<h1>Is it friday yet?</h1>

<p class="<?php echo $cssclass; ?>"><?php echo $text; ?></p>

</body>
</html>
```

4.3 Exercises

Now that you know how to use PHP and how to produce a decent HTML webpage, it's time for bigger exercises.

From now on all output should be valid xhtml 1.0-strict. Use a validator (e.g. `http://validator.w3.org/`) to check your code. Your HTML should also be semantically correct (e.g. `h1`,`li`, `p` used correctly).

You can add CSS to your pages if you want. It's a good idea to make one css-file and reuse it all the time.

Make sure this CSS is valid as well. If you want to use CSS 3 styles, you can, but you won't be judged on fancyness.

Exercise 4.1 - generate an unordered list

- Download the source file from
  `http://dynweb.webontwerp.khleuven.be/exercises/4.1-source.txt`
- Create a PHP script that achieves the result you can see at
  `http://dynweb.webontwerp.khleuven.be/exercises/4.1.php`.
  At least all bold text should be generated with PHP -code.
  Don't forget to also have a look at the source-code.
- Upload your file so that it is accessible at
  `http://<studentnr>.webontwerp.khleuven.be/exercises/4.1.php`

Exercise 4.2 - generate a table

- Download the source file from
  `http://dynweb.webontwerp.khleuven.be/exercises/4.2-source.txt`
- Create a PHP script that achieves the result you can see at
  `http://dynweb.webontwerp.khleuven.be/exercises/4.2.php`.
  Try to make is as dynamic as possible (number of columns and number of rows).
  Don't forget to also have a look at the source-code.
- Upload your file so that it is accessible at
  `http://<studentnr>.webontwerp.khleuven.be/exercises/4.2.php`
- Advanced: have a look at
  `http://dynweb.webontwerp.khleuven.be/exercises/4.2b.php`.
  Change your code so that you only have to change the data-array to get the different outputs 4.2 and 4.2b.
  Get the new data from
  `http://dynweb.webontwerp.khleuven.be/exercises/4.2b-source.txt`.

Exercise 4.3 - generate a page with form, menubars and newsitems

- Download the source file from
  `http://dynweb.webontwerp.khleuven.be/exercises/4.3-source.txt`
- Create a PHP script that achieves the result you can see at
  `http://dynweb.webontwerp.khleuven.be/exercises/4.3.php`.
  Make sure you first have a look at the examples of lesson02.
- Upload your file so that it is accessible at
  `http://<studentnr>.webontwerp.khleuven.be/exercises/4.3.php`
- Advanced: play with the data in the `$newsitems`, `$formdata`, `$questiontypes`, `$current_uri` variables in the beginning of the file and make sure your page still works.