Session 1: PHP Arrays

The first session this year will be a session focused on understanding PHP ***arrays*** and ***associative arrays***. This will help you understand the first few sessions of WAD. This exercise will use the same scenario - HitTastic! - as you saw last year in DFTI, but will allow you to explore more deeply what the variables in the script represent. You will also be writing the script in a slightly different way.

Exercise

The main aim of this exercise is to help you understand PHP arrays and associative arrays.

HitTastic! is an online music database. On Edward2 (see here outside university or here inside university) the HitTastic! database has been set up. The songs are in a table called ***wadsongs***. Note that it is slightly different to the one you used last year. You can view the songs at

```
http://edward2/wad/hits.php
```

Complete the code below. This is the start of a script which takes an artist as a query string (GET variable) and searches for all songs by that artist in the ***wadsongs*** database table. Fill in the SQL as indicated.

```
<html>
<head>
<link rel='stylesheet' type='text/css'
href='http://www.free-map.org.uk/course/css/dfti0910.cs
</head>
<body>

<?php

$a = $_GET["artist"];
$conn = new PDO("mysql:host=localhost;dbname=dftitutori
                                  "dftitutorials", "dftit

$result = $conn->query("ADD YOUR SQL HERE");
$rows = $result->fetchAll(PDO::FETCH_ASSOC);

print_r($rows);

?>
```

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

Test it out by uploading to your space on Edward2 (same login details as last year) and supplying the artist as a ***query string*** using the following artists. You do not need a form! Just supply the artist as a query string.

- Oasis
- Coldplay
- David Bowie
- Madonna
- The Beatles

or any other artist that you KNOW has had UK number one hits.

What do you see? Can you see what the following lines do:

- $rows = $result->fetchAll(PDO::FETCH_ASSOC);
- print_r($rows);

$rows is a type of variable called an ***array***. An array is a variable which can hold more than one value. The ***fetchAll()*** function fetches all the matching rows into an array, represented by the variable ***$rows***. So ***$rows*** is an array of all the records in the database matching the SQL query.

The ***print_r()*** function displays the contents of this array. You will see from this that the array contains each matching row. You will also see that ***each record has an index***, so that the first record has index 0, the second record has index 1, and so on.

Now modify your code as follows:

```
<!DOCTYPE html>
<html>
<head>
<link rel='stylesheet' type='text/css'
href='http://www.free-map.org.uk/course/css/dfti0910.cs
</head>
<body>

<?php

$a = $_GET["artist"];
$conn = new PDO("mysql:host=localhost;dbname=dftitutori
                                "dftitutorials", "dftit

$result = $conn->query("ADD YOUR SQL HERE");
```

```
$rows = $result->fetchAll(PDO::FETCH_ASSOC);
foreach ($rows as $row)
{
    echo "Current row:<br />";
    print_r($row);
}

?>

</body>
</html>
```

What output does this give? Can you explain what is happening?

Looping through the results

We saw above that ***$rows*** is an array of each record returned from the SQL query. What the code above is doing is looping through each ***individual member*** of the array. Specifically:

```
foreach ($rows as $row)
```

is looping through each member of the array in turn and placing it inside the variable ***$row***. In other words, each time the loop runs, ***$row*** will contain the current row.

The line:

```
print_r($row);
```

again displays the contents of the variable ***$row***. You can see from this that ***the individual row, $row, is itself an array***. However, this time, the array is not indexed numerically: it is indexed using ***non-numerical indices*** representing the columns from the database. So, for example, ***$row ["artist"]*** represents the artist field of the current row. This type of array is known as an ***associative array***.

Displaying the results in a readable format

Now that we have explored what the ***$rows*** and ***$row*** variables represent, we are now going to complete the script so that it actually displays the search results in a readable format. Comment out the ***print_r()*** statements and echo the song, artist and year of each matching record.

(Credits for the database: the songs in the database are the UK number ones from 1960-2015, plus a few songs from this year. This information was taken from Wikipedia who took it in turn from the Official Charts Company).