

## PHP Handout # 2

### PHP Arrays

---

An array is a special variable, which can store multiple values in one single variable.

In PHP, there are three kind of arrays:

- **Numeric array** - An array with a numeric index
- **Associative array** - An array where each ID key is associated with a value
- **Multidimensional array** - An array containing one or more arrays

#### Numeric Arrays

Numeric arrays can be created in two ways:

By automatically assigning values:

```
$cars=array("Saab","Volvo","BMW","Toyota");
```

Also by assigning the index manually:

```
$cars[0]="Saab";  
$cars[1]="Volvo";  
$cars[2]="BMW";  
$cars[3]="Toyota";
```

We can access the array values by referring to the array name and index:

```
<?php  
$cars[0]="Saab";  
$cars[1]="Volvo";  
$cars[2]="BMW";  
$cars[3]="Toyota";  
echo $cars[0] . " and " . $cars[1] . " are Swedish cars."  
?>
```

#### Associative Arrays

An associative array, each ID key is associated with a value.

This example assigns age values to different persons:

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

Another ways is to assign it individually:

```
$ages['Peter'] = "32";  
$ages['Quagmire'] = "30";  
$ages['Joe'] = "34";
```

And to access the array just indicate the ID of the array being accessed:

```
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";

echo "Peter is " . $ages['Peter'] . " years old.";
?>
```

## Multidimensional Arrays

Here we have a multidimensional array with automatically assigned ID keys:

```
$families = array
(
    "Griffin"=>array
    (
        "Peter",
        "Lois",
        "Megan"
    ),
    "Quagmire"=>array
    (
        "Glenn"
    ),
    "Brown"=>array
    (
        "Cleveland",
        "Loretta",
        "Junior"
    )
);
```

If we write an output for this array this is what it would look like:

```
Array
(
    [Griffin] => Array
        (
            [0] => Peter
            [1] => Lois
            [2] => Megan
        )
    [Quagmire] => Array
        (
            [0] => Glenn
        )
    [Brown] => Array
        (
            [0] => Cleveland
            [1] => Loretta
            [2] => Junior
        )
)
```

Lets try accessing and displaying a single value from the array above:

```
echo "Is " . $families['Griffin'][2] . " a part of the Griffin family?";
```

## PHP Form Handling

---

Any form element in an HTML page will automatically be available to your PHP Scripts.

For example lets create an HTML page and a PHP one.

<pre>&lt;html&gt; &lt;body&gt;  &lt;form action="welcome.php" method="post"&gt; Name: &lt;input type="text" name="fname" /&gt; Age: &lt;input type="text" name="age" /&gt; &lt;input type="submit" /&gt; &lt;/form&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<pre>&lt;html&gt; &lt;body&gt;  Welcome &lt;?php echo \$_POST["fname"]; ?&gt;!  You are &lt;?php echo \$_POST["age"]; ?&gt; years old.  &lt;/body&gt; &lt;/html&gt;</pre>
form.html	welcome.php

When a user fills out the form above and click on the submit button, the form data is sent to welcome.php  
View the output.

### The \$\_GET function

The built-in \$\_GET function is used to collect values from a form sent with method="get".

Information sent from a form with the **GET** method is **visible** to everyone (it will be displayed in the browser's address bar) and **has limits on the amount of information to send** (max. 100 characters).

<pre>&lt;form action="welcome.php" method="get"&gt; Name: &lt;input type="text" name="fname" /&gt; Age: &lt;input type="text" name="age" /&gt; &lt;input type="submit" /&gt; &lt;/form&gt;</pre>	<pre>Welcome &lt;?php echo \$_GET["fname"]; ?&gt;.  You are &lt;?php echo \$_GET["age"]; ?&gt; years old!</pre>
form.html	welcome.php

When using method="get" in HTML forms, all variable names and values are displayed in the URL.

### The \$\_POST function

The built-in \$\_POST function is used to collect values in a form with method="post".

Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

View the first example of form handling.

Information sent from a form with the **POST** method is **invisible** to others and has **no limits on the amount of information to send**.

## The \$\_Request function

The \$\_REQUEST function can be used to collect form data sent with both the GET and POST methods.

### Example

Welcome <?php echo \$\_REQUEST["fname"]; ?>!<br />

You are <?php echo \$\_REQUEST["age"]; ?> years old.

**That concludes this handout**

