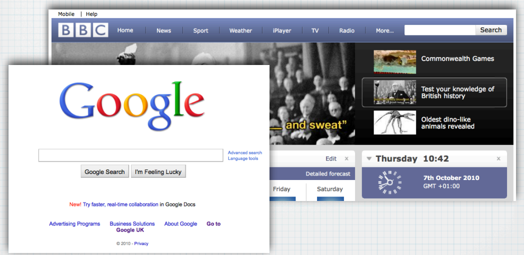
PHP

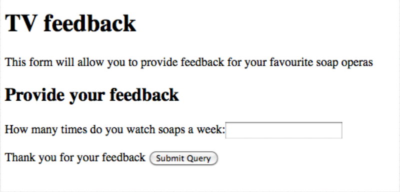
forms,
control structures - if / switch

Web forms

Most common way of getting data from user



Web forms



TV feedback

This form will allow you to provide feedback for your favourite soap operas

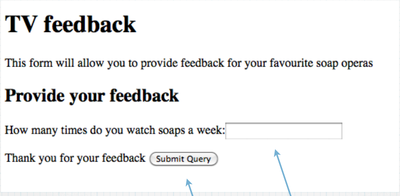
Provide your feedback

How many times do you watch soaps a week:

Thank you for your feedback [Submit Query]

Demo

Web forms



TV feedback

This form will allow you to provide feedback for your favourite soap operas

Provide your feedback

How many times do you watch soaps a week:

Thank you for your feedback [Submit Query]

Input field - text

Input field - submit button

Web forms

No PHP in here - can be saved as an .html file

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response1.php" method="get">

<p>How many times do you watch soaps a week:<input type="text" name="timesaweek" /></p>
<p>Thank you for your feedback <input type="submit" name="continue" /></p>

</form>

</body>
```

Web forms

form must be indicated by form element

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response1.php" method="get">




</form>

</body>
```

Needs at least three attributes name, action and method

Web forms

form must be indicated by form element

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response1.php" method="get">


</form>

</body>
```

Use name to give each form on the page a unique name

Web forms

form must be indicated by form element

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response1.php" method="get">


</form>

</body>
```

Use action to indicate the 'next' page where the form will be processed

Web forms

form must be indicated by form element

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response1.php" method="get">


</form>

</body>
```

Use method to indicate the way that data will be transferred

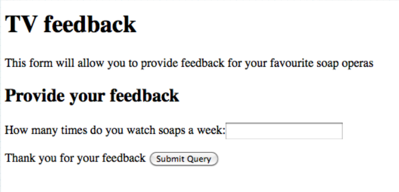
Web forms

Two ways of passing information between pages

get	form information is passed through the URL
post	form information is embedded in the HTTP stream

..more on this in a minute

form elements



TV feedback

This form will allow you to provide feedback for your favourite soap operas

Provide your feedback

How many times do you watch soaps a week:

Thank you for your feedback

```
<input type="text" name="timesaweek" />
<input type="submit" name="continue" />
```

Give every form element a unique name - these are used to pass the data values and in the processing page

method="get"

The form values are passed in the URL using name = value pairs

```
responseget1.php?timesaweek=4&continue=Submit+Query
```

method="get"

The form values are passed in the URL using name = value pairs

```
responseget1.php?timesaweek=4&continue=Submit+Query
```

Special encoding has to occur for spaces, =, & etc.

Called URL encoding and done automatically by the browser - note that name comes from the form

method="get"

The form values are passed in the URL using name = value pairs

```
responseget1.php?timesaweek=4&continue=Submit+Query
```

The continue button is also a form element - here with the default value

$_GET['....']

php creates a variable for each form element passed through using this notation

```
$_GET['form_element_name']
```

Best to create a variable to get values out at the top of the page

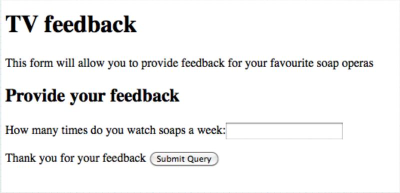
```
$timesaweek =$_GET['timesaweek'];
```

Web forms

responseget1.php page

```
<?php
$timesaweek =$_GET['timesaweek'];
?>

<html>
<head>
</head>
<body>
<h1>Responses</h1>
<p>Here are the results</p>
<p>
<?php
print "<p>The amount watched was ".$timesaweek."</p>";

?>
</p>
</body>
</html>
```

Web forms - POST example



TV feedback

This form will allow you to provide feedback for your favourite soap operas

Provide your feedback

How many times do you watch soaps a week:

Thank you for your feedback

Looks the same in the browser

method="post"

```
<body>
<h1>TV feedback</h1>
<p>This form will allow you to provide feedback for your favourite soap operas</p>
<h2>Provide your feedback</h2>

<form name="soapform" action="response2.php" method="post">

<p>How many times do you watch soaps a week:<input type="text" name="timesaweek" /></p>
<p>Thank you for your feedback <input type="submit" name="continue" /></p>

</form>

</body>
```

Small change here

But how is the data passed to the next page when using POST?

Web forms

Two ways of passing information between pages

get	form information is passed through the URL
post	form information is embedded in the HTTP stream

Web forms

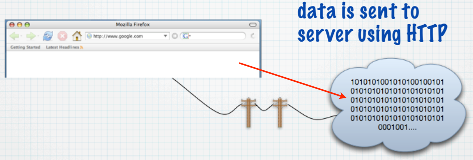
Two ways of passing information between pages

get	form information is passed through the URL
post	**form information is embedded in the HTTP stream**

How _POST works ...

User clicks on form

Page request AND data is sent to server using HTTP



```
POST /responses.php HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Content-Length: 27
Content-Type: application/x-www-form-urlencoded

timesaweek=4&continue=Submit+Query
```

$_POST['....']

php creates a variable for each form element passed through using this notation

```
$_POST['form_element_name']
```

Best to create a variable to get values out at the top of the page

```
$timesaweek =$_POST['timesaweek'];
```

Web forms

responsepost1.php page

```
<?php
$timesaweek =$_POST['timesaweek'];
?>

<html>
<head>
</head>
<body>
<h1>Responses</h1>
<p>Here are the results</p>
<p>
<?php
print "<p>The amount watched was ".$timesaweek."</p>";

?>
</p>
</body>
</html>
```

Uses the _POST array

Method choice - GET vs POST

GET - visible, bookmarkable

POST - not seen on screen, not bookmarkable

Control structures / flow control

- if ... else
- while ... do ..

Much of this material is explained in PHP programming 2nd Ed. Chap 2

Control structures / flow control

- if ... else
- while ... do ..

Much of this material is explained in PHP programming 2nd Ed. Chap 2

PHP files are processed top to bottom in sequence

```
<html>
<?php ... ?>
<head>
<?php ... ?>
<title>... <?php ... ?> ...</title>
</head>
<body>
<p>
<?php ... ?>
</p>
</body>
</html>
```

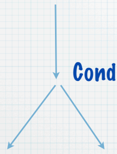
Starting at the top

Working down to the bottom

The control flow

But sometimes we need to have choices / alternatives

Start at the top

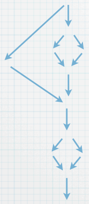


Condition here

Outcome 1 Outcome 2

Work down to the bottom

Can be complex flows



Done with an if

```
if (expression)
   statement
```

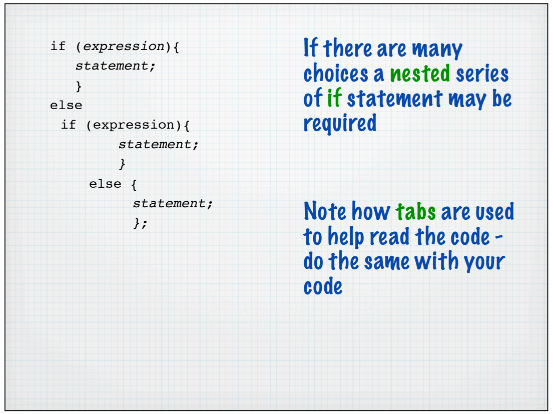
Perform the statement if the expression is true

```
if (expression)
   statement1
else
   statement2
```

Perform statement1 if the expression is true otherwise statement2

```
if (expression){
    statement;
    statement;
    }
else {
    statement;
    statement;
    };
```

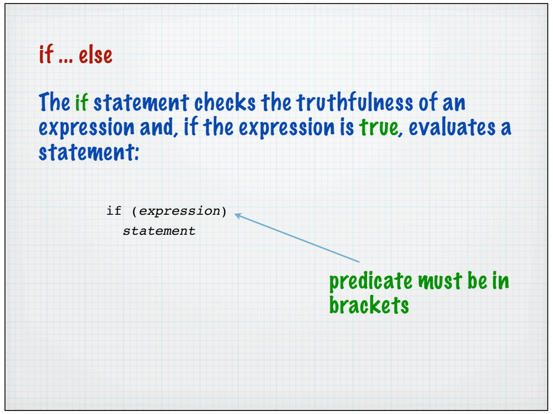
Perform blocks of statements if the expression is true otherwise ...



```
if (expression){
   statement;
   }
else
  if (expression){
         statement;
         }
      else {
           statement;
           };
```

If there are many choices a nested series of if statement may be required

Note how tabs are used to help read the code - do the same with your code

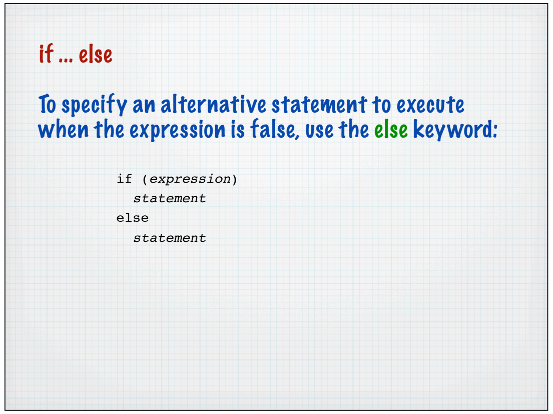


if ... else

The if statement checks the truthfulness of an expression and, if the expression is true, evaluates a statement:

```
if (expression)
  statement
```

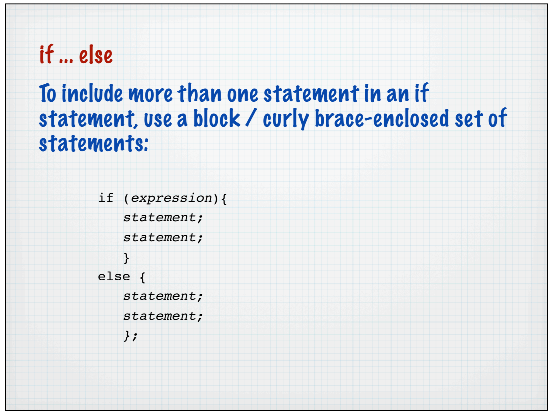
predicate must be in brackets



if ... else

To specify an alternative statement to execute when the expression is false, use the else keyword:

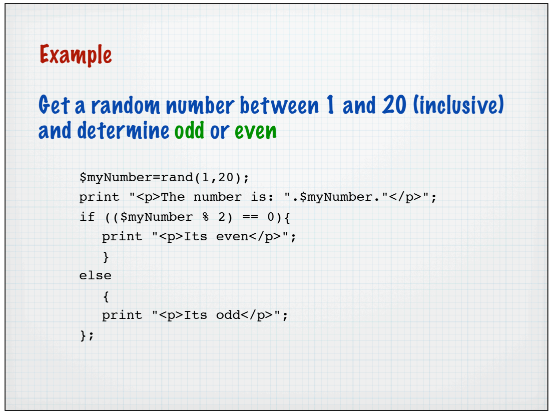
```
if (expression)
  statement
else
  statement
```



if ... else

To include more than one statement in an if statement, use a block / curly brace-enclosed set of statements:

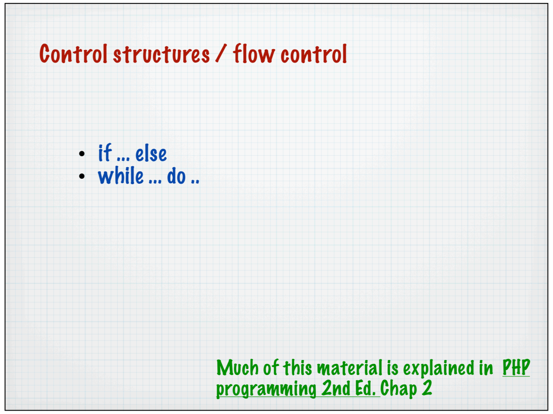
```
if (expression){
   statement;
   statement;
   }
else {
   statement;
   statement;
   };
```



Example

Get a random number between 1 and 20 (inclusive) and determine odd or even

```
$myNumber=rand(1,20);
print "<p>The number is: ".$myNumber."</p>";
if (($myNumber % 2) == 0){
   print "<p>Its even</p>";
   }
else
   {
   print "<p>Its odd</p>";
};
```



Control structures / flow control

- if ... else
- while ... do ..

Much of this material is explained in PHP programming 2nd Ed. Chap 2

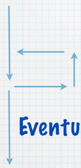
Control structures / flow control

- if ... else
- while ... do ..

Much of this material is explained in PHP programming 2nd Ed. Chap 2

Sometimes we need to do things many times

Start



Round the Loop

Eventually come out of the loop ...

Loops / Iteration / doing things over and over and over and over ....

a while loop

while loop

The structure of a while statement is:

```
while (condition)
    statement
```

Loop continues whilst condition is true

or with many statements -

```
while (condition){
    statement;
    statement;
    statement;
    statement;
};
```

Do something in here to change the condition (unless you want it to continue ∞)

while loop

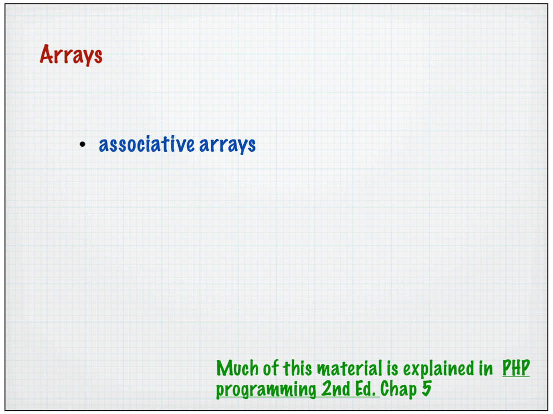
```
$today="Monday";

while($today<>"Friday"){
print "<p>Today is ".$today."</p>";
if (rand(1,10)>7){
    $today="Friday";
    };
};

print "Final value of today is ".$today;
```

If the condition is initially false, the loop doesn't execute at all (i.e. 0 times)

while loop

```
$today="Friday";

while($today<>"Friday"){
print "<p>Today is ".$today."</p>";
if (rand(1,10)>7){
    $today="Friday";
    };
};

print "Final value of today is ".$today;
```

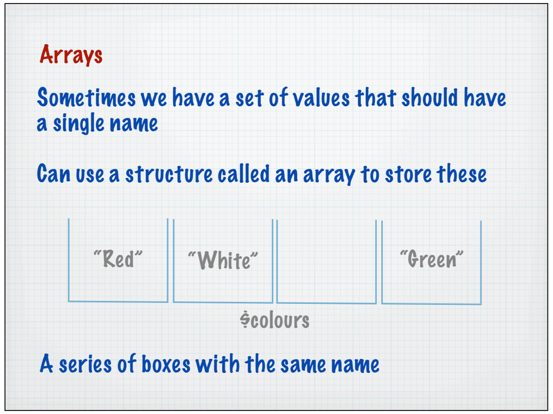
If the condition is initially false, the loop doesn't execute at all (i.e. 0 times)



Arrays

- associative arrays

Much of this material is explained in PHP programming 2nd Ed. Chap 5



Arrays

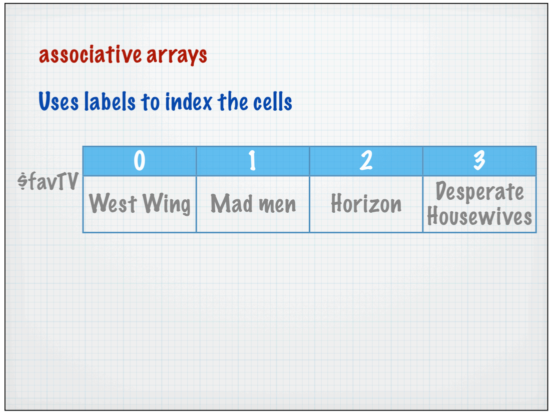
Sometimes we have a set of values that should have a single name

Can use a structure called an array to store these

"Red"	"White"		"Green"

$colours

A series of boxes with the same name

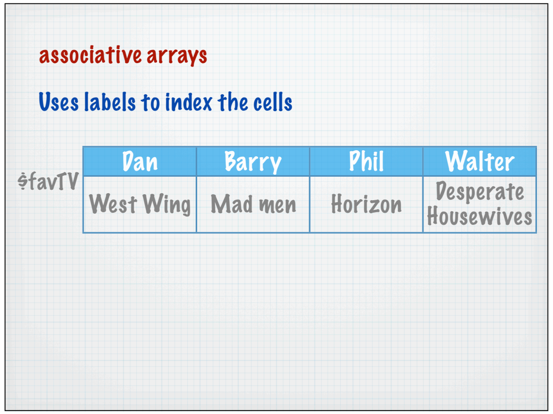


associative arrays

Uses labels to index the cells

$favTV

0	1	2	3
West Wing	Mad men	Horizon	Desperate Housewives

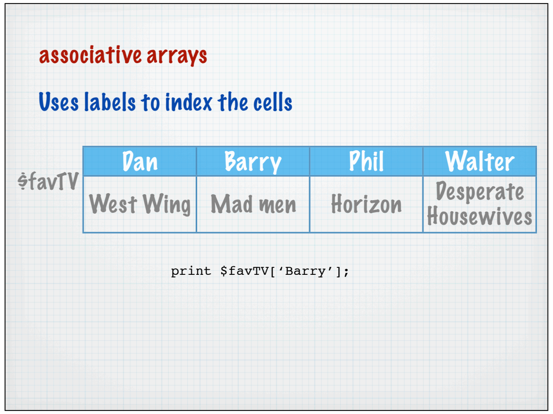


associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives



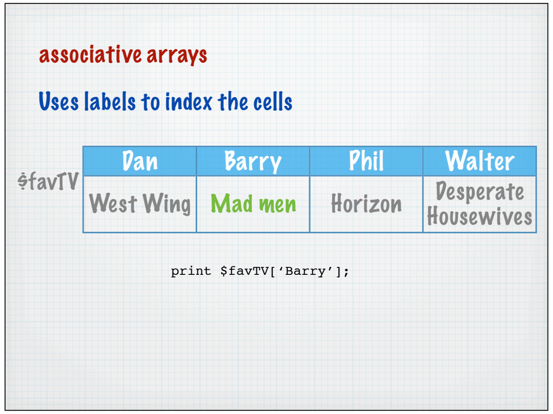
associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```



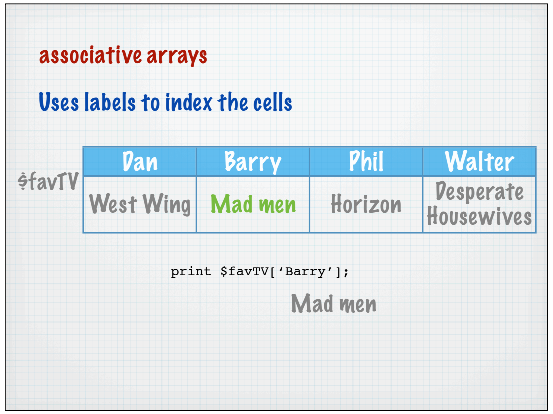
associative arrays

Uses labels to index the cells

$favTV

Dan	Barry	Phil	Walter
West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```



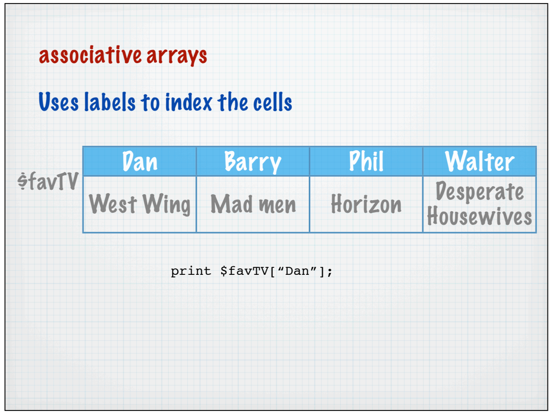
associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV['Barry'];
```

Mad men

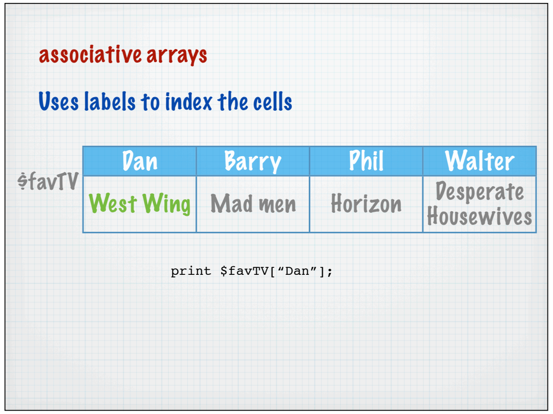


associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV["Dan"];
```

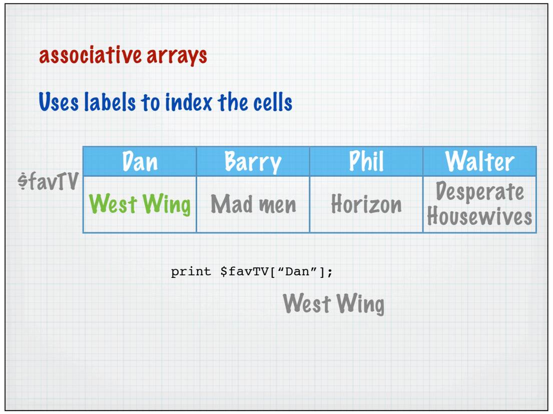


associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV["Dan"];
```



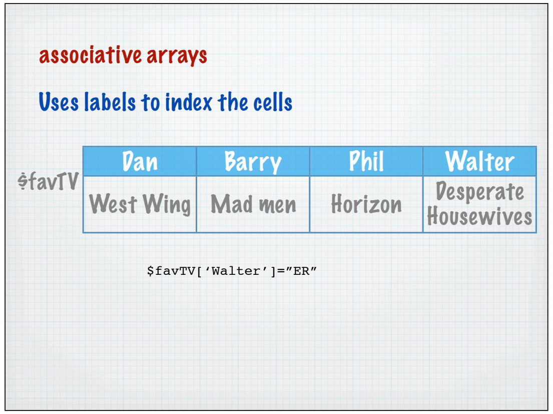
associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	Desperate Housewives

```
print $favTV["Dan"];
```

West Wing

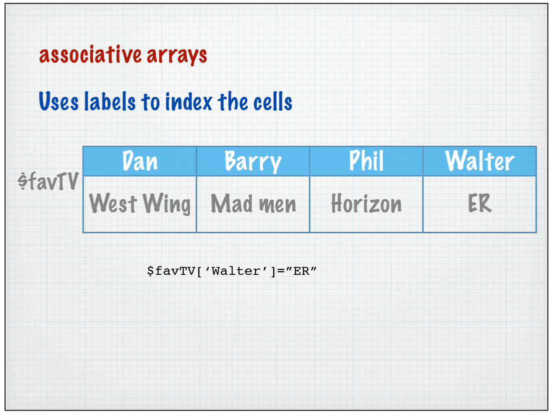


associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	Desperate Housewives

```
$favTV['Walter']="ER"
```



associative arrays

Uses labels to index the cells

	Dan	Barry	Phil	Walter
$favTV	West Wing	Mad men	Horizon	ER

```
$favTV['Walter']="ER"
```

associative arrays

Use simple assignment to create the array

```
$favTV[“Dan”]=”West Wing”;
```

	Dan
$favTV	West Wing

associative arrays

Use simple assignment to create the array

```
$favTV[“Dan”]=”West Wing”;
$favTV[“Barry”]=”Mad men”;
```

	Dan	Barry
$favTV	West Wing	Mad men

associative arrays

Use simple assignment to create the array

```
$favTV[“Dan”]=”West Wing”;
$favTV[“Barry”]=”Mad men”;
$favTV[“Phil”]=”Horizon”;
```

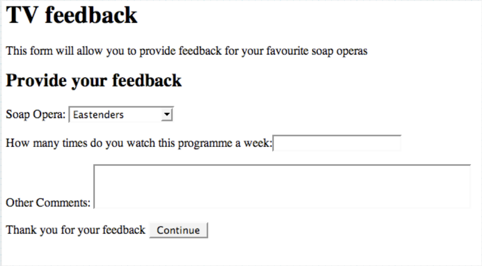
	Dan	Barry	Phil
$favTV	West Wing	Mad men	Horizon

useful functions

See appropriate references for full definitions of these

functions	explanation
count()	no of array cells
array_pad()	create an array with the same value
array-slice()	extract part of an array
array_keys()	returns an array of the keys
array_key_exists()	see if a key exists

Web forms



TV feedback

This form will allow you to provide feedback for your favourite soap operas

Provide your feedback

Soap Opera: Eastenders

How many times do you watch this programme a week:

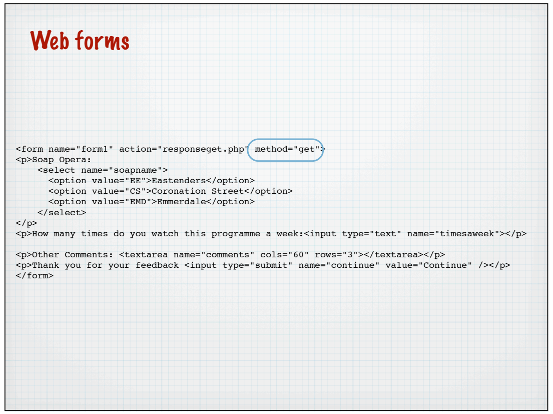
Other Comments:

Thank you for your feedback

Web forms

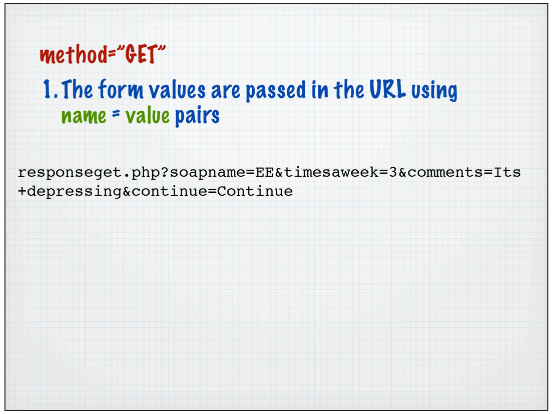
Two ways of passing information between pages

get	form information is passed through the URL
post	form information is embedded in the HTTP stream



Web forms

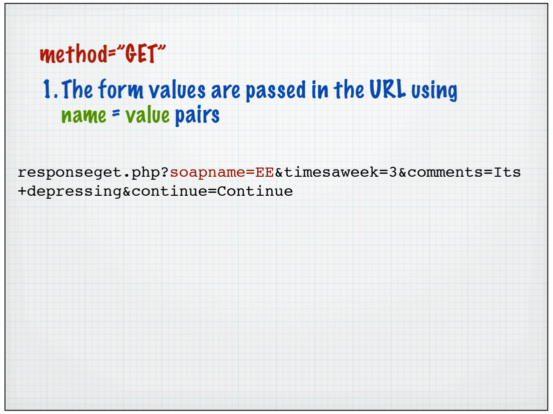
```
<form name="form1" action="responseget.php" method="get">
<p>Soap Opera:
    <select name="soapname">
      <option value="EE">Eastenders</option>
      <option value="CS">Coronation Street</option>
      <option value="EMD">Emmerdale</option>
    </select>
</p>
<p>How many times do you watch this programme a week:<input type="text" name="timesaweek"></p>

<p>Other Comments: <textarea name="comments" cols="60" rows="3"></textarea></p>
<p>Thank you for your feedback <input type="submit" name="continue" value="Continue" /></p>
</form>
```



method="GET"

1. The form values are passed in the URL using name = value pairs

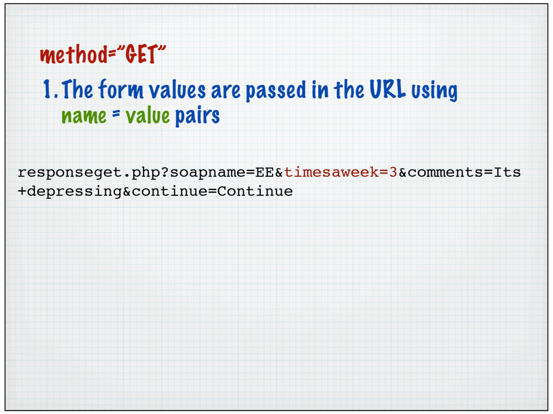
```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```



method="GET"

1. The form values are passed in the URL using name = value pairs

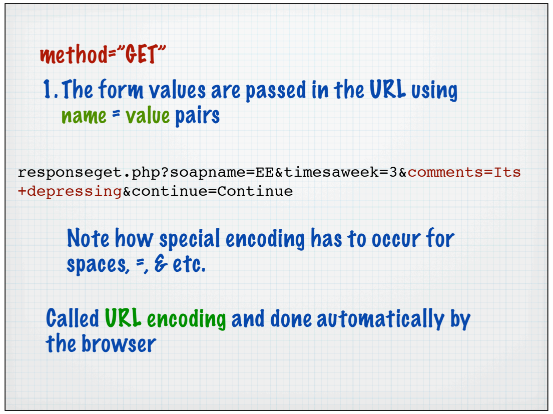
```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```



method="GET"

1. The form values are passed in the URL using name = value pairs

```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```



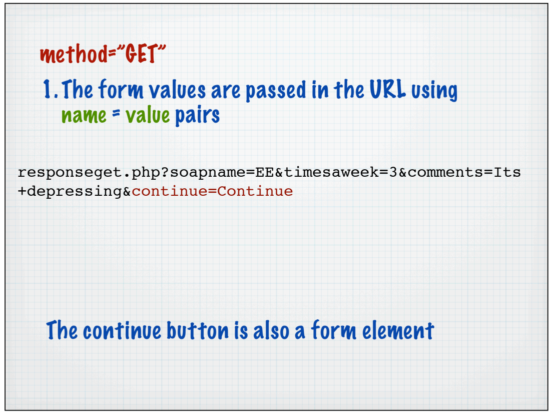
method="GET"

1. The form values are passed in the URL using name = value pairs

```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```

Note how special encoding has to occur for spaces, =, & etc.

Called URL encoding and done automatically by the browser



method="GET"

1. The form values are passed in the URL using name = value pairs

```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```

The continue button is also a form element

$_GET associative array

PHP automatically creates an associative array with the passed URL values

```
responseget.php?soapname=EE&timesaweek=3&comments=Its
+depressing&continue=Continue
```

	soapname	timesaweek	comments	continue
$_GET	EE	3	its depressing	Continue

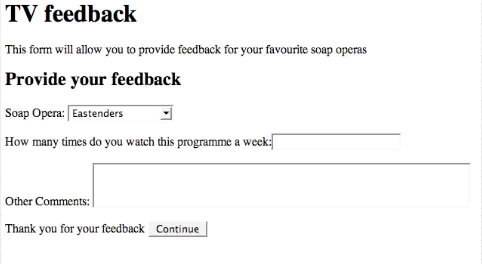
Web forms

responseget.php page

```
$soapname=$_GET['soapname'];
$timesaweek =$_GET['timesaweek'];
$comments=$_GET['comments'];


print "The soap watched is ".$soapname;
print "The amount watched was ".$timesaweek."<br />";
print "Comments ".$comments."<br />";
```

We can use the _GET associative array to get at the form selections

Web forms - POST example



TV feedback

This form will allow you to provide feedback for your favourite soap operas

Provide your feedback

Soap Opera: Eastenders

How many times do you watch this programme a week:

Other Comments:

Thank you for your feedback Continue

Looks the same in the browser

Web forms

```
<form name="form1" action="responsepost.php" method="post">
<p>Soap Opera:
    <select name="soapname">
      <option value="EE">Eastenders</option>
      <option value="CS">Coronation Street</option>
      <option value="EMD">Emmerdale</option>
    </select>
</p>
<p>How many times do you watch this programme a week:<input type="text" name="timesaweek"></p>

<p>Other Comments: <textarea name="comments" cols="60" rows="3"></textarea></p>
<p>Thank you for your feedback <input type="submit" name="continue" value="Continue" /></p>
</form>
```

Small change here

$_GET associative array

PHP automatically creates an associative array with the passed URL values from the HTTP stream

	soapname	timesaweek	comments	continue
$_POST	EE	3	its depressing	Continue

Web forms

responsepost.php page

```
$soapname=$_POST['soapname'];
$timesaweek =$_POST['timesaweek'];
$comments=$_POST['comments'];

print "The soap watched is ".$soapname
print "The amount watched was ".$timesaweek."<br />";
print "Comments ".$comments."<br />";
```

Uses the _POST array

name-value pairs INSIDE the http stream