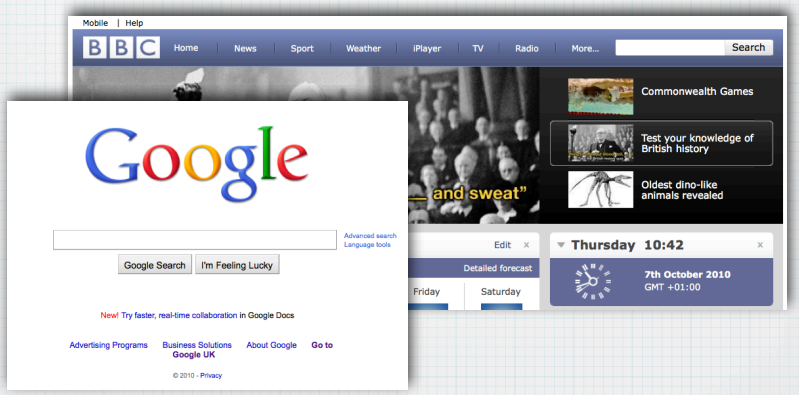
PHP

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

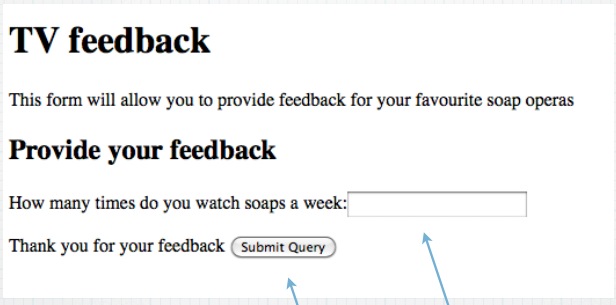
Provide your feedback

How many times do you watch soaps a week:

Thank you for your feedback Submit Query

Demo

Web forms



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

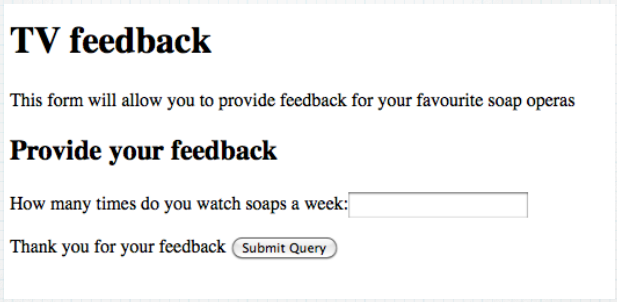
Web forms

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




```
<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

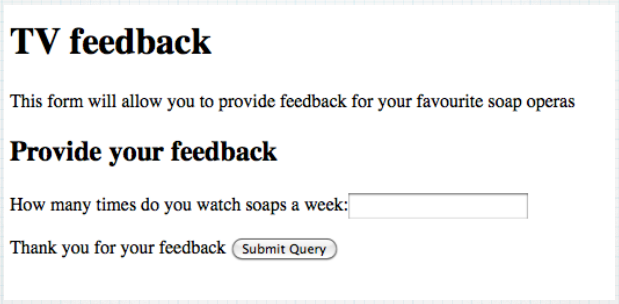
responseget1.php page

```
<?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



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

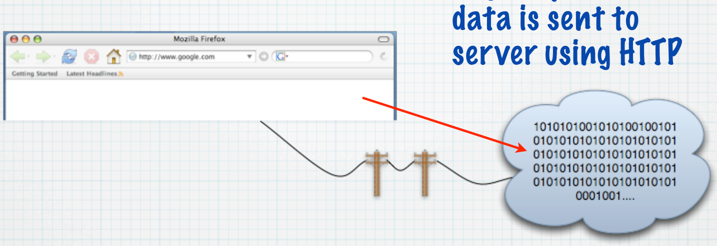
Two ways of passing information between pages

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
- Switch
- for loops
- while ... do ..
- do ... while ...

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

Control structures / flow control

- if ... else
- Switch
- for loops
- while ... do ..
- do ... while ...

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

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>";
};
```

Note - Excellent PHP manual on-line

http://uk.php.net/

Every entry has

- Description
- Syntax
- Examples
- Comments and hints from other users

Entry for rand()

switch

A switch statement is given an expression and compares its value to all cases in the switch; all statements in a matching case are executed, up to the first break keyword it finds. If none match, and a default is given, all statements following the default keyword are executed, up to the first break keyword encountered.

From PHP programming 2nd Ed. Chap 2

Probably best explained with an example

switch

When the if statement has many sub-if parts, a switch statement may be better

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

switch

```
switch($name) {
      case 'value 1 of name':
        // do something
        break;
      case 'value 2 of name':
        // do something
        break;
      case 'value 3 of name':
        // do something
        break;
      case 'value 4 of name':
        // do something
        break;
    }
```

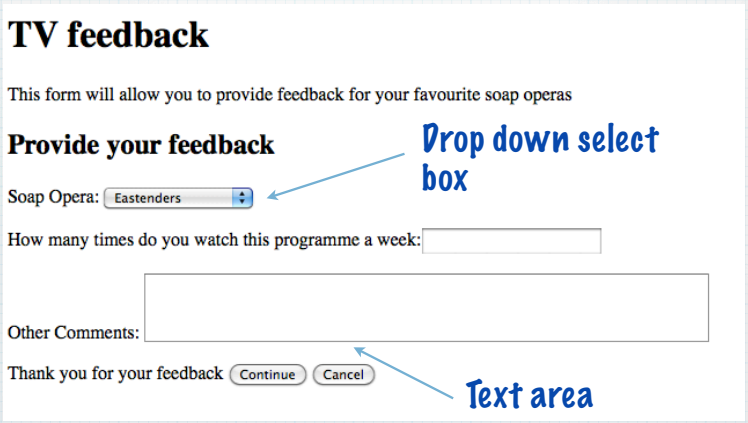
Best to choose this when $name has a limited set of values known in advance

switch

```
$myNumber=rand(1,4);
print "<p>The number is: ".$myNumber."</p>";
switch ($myNumber) {
    case 1:
        print "It's 1 which is low";
        break;
    case 2:
        print "It's 2 which is better than 1";
        break;
    case 3:
        print "It's 3 which is better than average";
        break;
    case 4:
        print "It's 4 - very high";
        break;
};
```

Note how the break is required to stop 'fall-through'

More form elements



Web forms

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>TV Feedback</title>
</head>
<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="form1" action="response3.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" />
<input type="submit" name="continue" value="Cancel" /></p>
</form>
</body>
</html>
```

Web forms response.php page

```
    <p>Here are the results</p>
    <p>
    <?php
    $soapname=$_POST['soapname'];
    $timesaweek =$_POST['timesaweek'];
    $rating=$_POST['rating'];
    $comments=$_POST['comments'];
    $continue=$_POST['continue'];

    if($continue=="Continue"){
        print "The soap watched is ";
        switch ($soapname) {
            case EE:
                print "Eastenders<br />";
                break;
            case CS:
                print "Coronation Street<br />";
                break;
            default:
                print "Emmerdale<br />";
                break;
            };
        print "The amount watched was ".$timesaweek."<br />";
        print "Comments ".$comments."<br />";
    }
    else
    {
        print "The cancel button was pressed";
    };
    ?>
    </p>
```