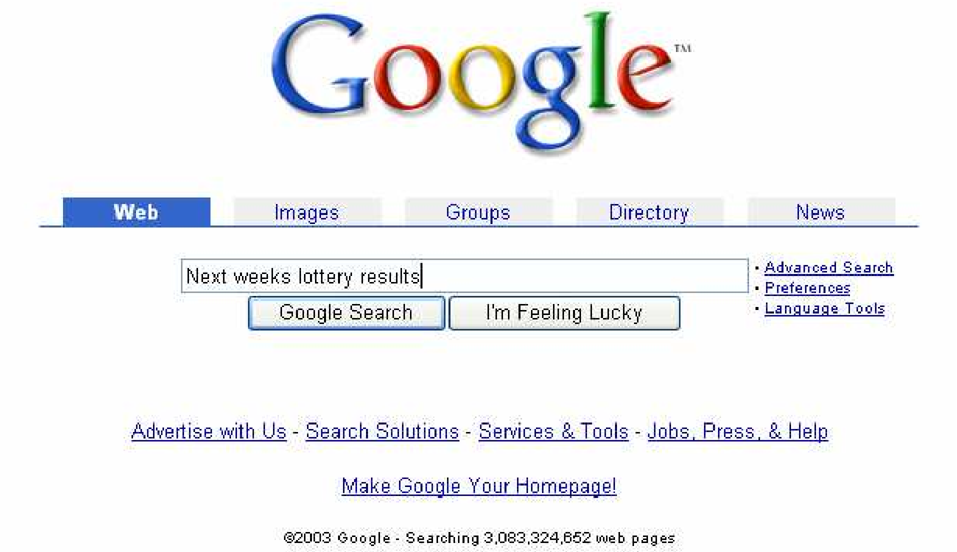
# HTML Forms

If you have been searching on the Internet then there is a very good chance you have used HTML forms. They are the most popular way of gathering information from the user (client) and sending it to the server.

If you want to see a form in action just go to google.com and enter a something into the search box. That is a form.

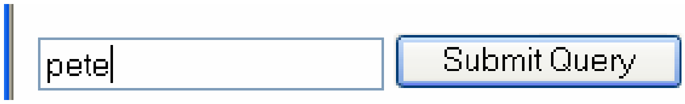

Google™
Web
Images
Groups
Directory
News
Google Search
I'm Feeling Lucky
Advanced Search
Preferences
Language Tools
Advertise with Us - Search Solutions - Services & Tools - Jobs, Press, & Help
Make Google Your Homepage!
©2003 Google - Searching 3,083,324,652 web pages

Next weeks lottery results

So how we go about creating one of these then? Well as with everything in HTML it is all about tags:

```
<form action=http://www.google.com/search  method=”get”>
<input type=”text” name=”q”>
<input type=”submit”>
</form>
```

The code above, if placed inside a HTML documents BODY tag, will create a simple form like the one below:



pete

## Exercise 1.

1. Type in the above code in a new HTML document, save it as form1.htm
2. Opening it in a browser, does it look like the screenshot above?
3. Type something in the box and click the *submit button*. What happens?

Here is a break down of what all the different parts do:

**`action=”…”`** Tells the browser where it should send the information collected by the form.

**`method=”get”`** There are two methods that the browser can use to send the data. GET or POST. Get is faster, not very secure and limited to < 1000 characters. POST has no character limit and is more secure. If in doubt use GET.

**`type=”text”`** What type of form input to display there are several as listed later.

**`name=”q”`** The form inputs must be named so the program the browser sends them to knows what to do with each of them. In the case of google it is expecting a piece of search text to be sent to it. It is excepting that search text to be called “**q**” for query.

**`type=”submit”`** Every form needs a method of sending itself or submitting itself to the destination given in the action property. Lucky one is provided for you, the submit button.

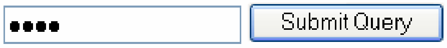
# Basic Form Inputs

## Text Boxes

```
<input type=”text” name=”text1” maxlength=”100”>
<input type=”text” name=”text2”>
<input type=”text” name=”text3” value=”Enter some text here”>
```

## Password Boxes (hide the users input)

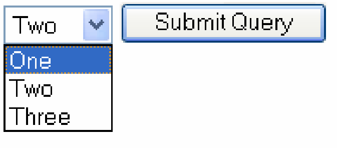
<input type=”password” name=”password1”>



## Select Boxes (combo boxes)

```
<select name=”combo1”>
<option value=”Number1”>One</option>
<option value=”Number2” selected>Two</option>
<option value=”Number3”>Three</option>
</select>
```

Note the *selected* property in the second option tag tells the browser that this option should be the one selected as default when the page first loads.



## Exercise 2.

1. Save your HTML file as form2.htm
2. Remove the text field code from the previous exercise.
3. Edit the file to hide the text users enter on your form.
4. Test your changed in the browser
5. Ask the lecturer to confirm your findings.

## Exercise 3.

1. Save you HTML file as form3.htm
2. Remove the password field code from the previous exercise.
3. Edit the file to include a Select box with five search options:
    a. Paris
    b. London
    c. New York
    d. Berlin
    e. Moscow
4. Test your page in your browser.
5. Does your form allow users to search for all five cities on Google?
6. Ask the lecturer to confirm your findings.

# Advanced Form Input

## Text Areas

Text areas are larger version of the text box. They have scroll bars and allow multiple lines to be entered. Hotmail and Yahoo use text areas to allow you to write your emails on their webpages.

```
<textarea name=”textarea1” rows=”8” cols=”20”>
This is the text that will first be  displayed!
</textarea>
```

## Exercise 4.

1. Save your HTML file as form4.htm
2. Remove the select box code from the previous exercise.
3. Add a text area to your form.
4. Test your page in your browser.
5. Experiment with the text area’s properties to try to find out what **rows** and **cols** do.
6. Ask the lecturer to confirm your findings.

## Radio Buttons

Radio buttons allow the browser to group together several checkboxes which the user can choose from. Only one of the boxes can be checked at anyone time though.

```
<input type=”radio ” name=”radio1” value=”one”> One
<br>
<input type=”radio” name=”radio1” value=”two”> Two
<br>
<input type=”radio” name=”radio1” value=”three”> Three
<br>
```

Note: Radio buttons are grouped together by setting the name property to be the same for all the buttons in the group. Once the form is submitted it is the setting stored in the *value* property of that radio button that is sent to the server and **not the text that follows it**.

## Exercise 5.

1. Save your HTML file as form5.htm
2. Remove the text area code from the previous exercise.
3. Edit your file to include five radio buttons on your form ensuring:
    a. Each one is on a separate line
    b. Each one searches for a different music Band
4. Test your page in your browser
5. Confirm your findings with your lecturer

## Exercise 6.

1. Create a new HTML page.
2. Include a form and set its action property to:

http://www.multimap.co.uk/map/places.cgi

3. Include a text field named “quicksearch”
4. Include a submit button.
5. Test your form in your browser
6. Confirm your findings with your lecturer