**INT106 – Lab – Form and Event#1**

**\*Credit:** [**www.w3schools.com**](http://www.w3schools.com) **and Jennifer Niederst Robbins**, **Learning web design 4th edition**

**HTML Form**

Now, let’s go to the main course. Let’s see about form and event.

**The <form> Element**

HTML forms are used to collect user input. For Example:

|  |
| --- |
| <form>  .  form elements  .  </form> |

HTML forms contain form elements.  
Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

**The <input> Element**

The <input> element is the most important form element.  
The <input> element has many variations, depending on the type attribute.  
Here are the types used in this chapter:

text Defines normal text input

radio Defines radio button input (for selecting one of many choices)

submit Defines a submit button (for submitting the form)

**Text Input**

<input type="text"> defines a one-line input field for text input:  
[**http://www.w3schools.com/html/tryit.asp?filename=tryhtml\_form\_text**](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_text)

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form>  **First name:<br>**  **<input type="text" name="firstname">**  **<br>**  **Last name:<br>**  **<input type="text" name="lastname">**  </form>  <p>Note that the form itself is not visible.</p>  <p>Also note that the default width of a text field is 20 characters.</p>  </body>  </html> |

**Radio Button Input**

<input type="radio"> defines a radio button.  
Radio buttons let a user select ONE of a limited number of choices:  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_radio>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form>  **<input type="radio" name="sex" value="male" checked>Male**  **<br>**  **<input type="radio" name="sex" value="female">Female**  </form>  </body>  </html> |

**Checkbox buttons**

<input type="checkbox"> defines a checkbox button.

|  |
| --- |
| <p>What type of music do you listen to?</p>  <ul>  <li><input type="checkbox" name="genre" value="punk" checked> Punk  rock</li>  <li><input type="checkbox" name="genre" value="indie" checked> Indie  rock</li>  <li><input type="checkbox" name="genre" value="hiphop"> Hip Hop</li>  <li><input type="checkbox" name="genre" value="rockabilly">  Rockabilly</li>  </ul> |

**The Submit Button**

<input type="submit"> defines a button for submitting a form to a form-handler.  
The form-handler is typically a server page with a script for processing input data.  
The form-handler is specified in the form's action attribute:  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  First name:<br>  <input type="text" name="firstname" value="Mickey">  <br>  Last name:<br>  <input type="text" name="lastname" value="Mouse">  <br><br>  <input type="submit" value="Submit">  </form>  <p>If you click "Submit", the form-data will be sent to a page called "action\_page.php".</p>  </body>  </html> |

**The Action Attribute**

The action attribute defines the action to be performed when the form is submitted.  
The common way to submit a form to a server, is by using a submit button.  
Normally, the form is submitted to a web page on a web server.  
In the example above, a server-side script is specified to handle the submitted form:

|  |
| --- |
| <form action="action\_page.php"> |

If the action attribute is omitted, the action is set to the current page.

**The Method Attribute**

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the forms:

|  |
| --- |
| <form action="action\_page.php" method="GET"> |

Or

|  |
| --- |
| <form action="action\_page.php" method="POST"> |

**When to Use GET?**

You can use GET (the default method):  
If the form submission is passive (like a search engine query), and without sensitive information.  
When you use GET, the form data will be visible in the page address:

|  |
| --- |
| action\_page.php?firstname=Mickey&lastname=Mouse |

\*Note: GET is best suited to short amounts of data. Size limitations are set in your browser.

**When to Use POST?**

You should use POST:  
If the form is updating data, or includes sensitive information (password).  
POST offers better security because the submitted data is not visible in the page address.

**The Name Attribute**

To be submitted correctly, each input field must have a name attribute.  
This example will only submit the "Last name" input field:  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit_id>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  First name:<br>  <input type="text" value="Mickey">  <br>  Last name:<br>  <input type="text" **name="lastname"** value="Mouse">  <br><br>  <input type="submit" value="Submit">  </form>  <p>If you click "Submit", the form-data will be sent to a page called "action\_page.php".</p>  <p>The first name will not be submitted, because the input element does not have a name attribute.</p>  </body>  </html> |

**Grouping Form Data with <fieldset>**

The <fieldset> element groups related data in a form.  
The <legend> element defines a caption for the <fieldset> element.  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_legend>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  **<fieldset>**  **<legend>Personal information:</legend>**  **First name:<br>**  **<input type="text" name="firstname" value="Mickey">**  **<br>**  **Last name:<br>**  **<input type="text" name="lastname" value="Mouse">**  **<br><br>**  **<input type="submit" value="Submit">**  **</fieldset>**  </form>  </body>  </html> |

**HTML Form Attributes**

An HTML <form> element, with all possible attributes set, will look like this:

For example:

|  |
| --- |
| <form action="**action\_page.php" method="GET" target="\_blank" accept-charset="UTF-8"**  **enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate**>  .  form elements  .  </form> |

Here is the list of <form> attributes:

**accept-charset** Specifies the charset used in the submitted form (default: the page charset).

**action** Specifies an address (url) where to submit the form (default: the submitting page).

**autocomplete** Specifies if the browser should autocomplete the form (default: on).

**enctype** Specifies the encoding of the submitted data (default: is url-encoded).

**method** Specifies the HTTP method used when submitting the form (default: GET).

**name** Specifies a name used to identify the form (for DOM usage: document.forms.name).

**novalidate** Specifies that the browser should not validate the form.

**target** Specifies the target of the address in the action attribute (default: \_self).

**HTML Form Elements**

**The <select> element** defines a drop-down list:  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_select>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  **<select name="cars">**  **<option value="volvo">Volvo</option>**  **<option value="saab">Saab</option>**  **<option value="fiat">Fiat</option>**  **<option value="audi">Audi</option>**  **</select>**  <br><br>  <input type="submit">  </form>  </body>  </html> |

**The <option> elements** defines the options to select.  
The list will normally show the first item as selected.  
You can add a selected attribute to define a predefined option.

<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_select_pre>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <p>You can preselect an option with the selected attribute.</p>  <form action="action\_page.php">  <select name="cars">  <option value="volvo">Volvo</option>  <option value="saab">Saab</option>  **<option value="fiat" selected>Fiat</option>**  <option value="audi">Audi</option>  </select>  <br><br>  <input type="submit">  </form>  </body>  </html> |

**The <textarea> Element**

The <textarea> element defines a multi-line input field (a text area):  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_textarea>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  **<textarea name="message" rows="10" cols="30">**  **The cat was playing in the garden.**  **</textarea>**  <br><br>  <input type="submit">  </form>  </body>  </html> |

**The <button> Element**

The <button> element defines a clickable button:  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_button>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  **<button type="button" onclick="alert('Hello World!')">Click Me!</button>**  </body>  </html> |

**Specialized text entry fields**

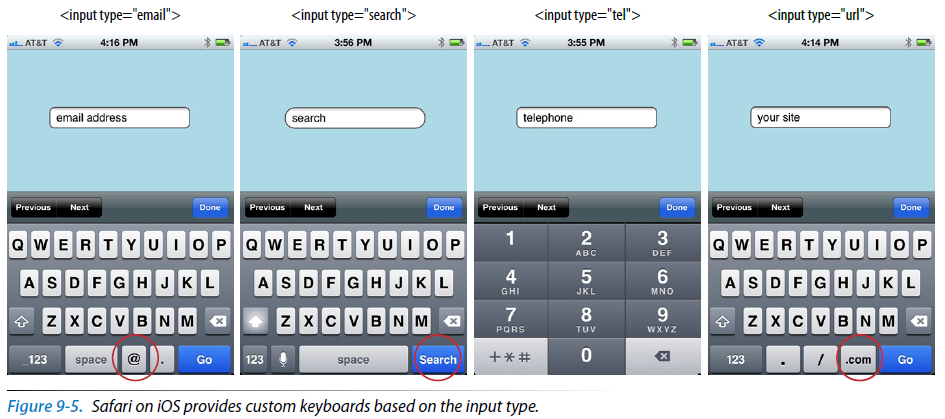
In addition to the generic single-line text entry, there are a number of input types for entering specific types of information such as passwords, search terms, email addresses, telephone numbers, and URLS.

**Password entry field**

A password field works just like a text entry field, except the characters are obscured from view using asterisk (\*) or bullet (•) characters, or another character determined by the browser.

<input type="password" name="pswd" maxlength="8" id="form-pswd">

**Other input types:**



**\*NOTE: The field will also validate the entered value for you.**

**The Reset button**

Just put in:  
<input type="reset">

And you will get a reset button which will reset all the input on the page.

**Practice#1**

|  |
| --- |
| You are the web designer in charge of creating the entry form for the Forcefield Sneakers “Pimp My Shoes!” Contest. The copy editor has handed you a sketch of the form’s content, complete with notes of how some controls should work. There are sticky notes from the programmer with information about the script and variable names you need to use.  Your challenge is to turn the sketch into a functional online form. I’ve given you a head start by creating a bare-bones document containing the text content and some minimal markup and styles. This document, contest\_entry.html, is available online at www.learningwebdesign.com/4e/materials. The source for the entire finished form is provided in Appendix A if you want to check your work. |

**Practice#2**

|  |
| --- |
| Use your HTML programming skills to create a web page that looks like this. |

**HTML5 Form Elements**

HTML5 added the following form elements:

<datalist>

<keygen>

<output>

\*Note: By default, browsers do not display unknown elements. New elements will not destroy your page.

**HTML5 <datalist> Element**

The <datalist> element specifies a list of pre-defined options for an <input> element.  
Users will see a drop-down list of pre-defined options as they input data.  
The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_datalist>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  <input list="browsers" name="browser">  **<datalist id="browsers">**  **<option value="Internet Explorer">**  **<option value="Firefox">**  **<option value="Chrome">**  **<option value="Opera">**  **<option value="Safari">**  **</datalist>**  <input type="submit">  </form>  <p><b>Note:</b> The datalist tag is not supported in Safari or IE9 (and earlier).</p>  </body>  </html> |

**\*NOTE**: In the <datalist> element, the options are only suggestions. The users can enter their own answer directly.

**HTML5 <keygen> Element**

The purpose of the <keygen> element is to provide a secure way to authenticate users.  
The <keygen> element specifies a key-pair generator field in a form.  
When the form is submitted, two keys are generated, one private and one public.  
The private key is stored locally, and the public key is sent to the server.  
The public key could be used to generate a client certificate to authenticate the user in the future.

<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_keygen>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php">  Username:  <br>  <input type="text" name="user">  <br><br>  Encryption:  <br>  **<keygen name="security">**  <br><br>  <input type="submit">  </form>  </body>  </html> |

**HTML5 <output> Element**

The <output> element represents the result of a calculation (like one performed by a script).  
<http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_output>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <form action="action\_page.php"  **oninput="x.value=parseInt(a.value)+parseInt(b.value)">**  **0**  **<input type="range" id="a" name="a" value="50">**  **100 +**  **<input type="number" id="b" name="b" value="50">**  **=**  **<output name="x" for="a b"></output>**  <br><br>  <input type="submit">  </form>  </body>  </html> |