



Creating Forms

(and available form fields)





Forms on the Web

- The majority of forms on the web are HTML forms.
- Other forms are used part of web applications such as:
 - Flash Applications
 - PDFs
 - Web Services
 - .NET Applications



All HTML forms must:

- Be enclosed in form tags
 - `<form>`
- Have one or more inputs
- Have a submit button



The `<form>` Tag

- The basic block of a form is the `<form>` tag which defines the area of form.
 - Only controls and attributes within the area defined by the form element will be processed or passed to a script.
- HTML form is a section of a document containing normal content, markup, special elements called “controls”.



Form Tag Attributes

Syntax for form tags:

- `<form action="process.asp" method="post">`

Required Elements:

- **action**
 - This defines what will happen when the form is submitted.
- **Method**
 - get
 - post



Get Method

Used to pass information collected by the form via the url.

- Most useful:
 - For short strings,
 - When you want the information visible to the user,
 - Or to allow the form information to be bookmarked for later use without filling the form out again.
- Don't use:
 - For sensitive information like userid and password,
 - Lots of information being passed by the form.



Post Method

Post Method hides the form field information when it is transferred to the page or application that will be processing the form.

- Most useful:
 - For information you don't want displayed,
 - Passing longer form responses
- Don't use:
 - If you want people to be able to save the form information,
 - Need real security for the form results.



The following types of controls are available within a form:

- Text
- Textarea
- Password
- Checkboxes
- Radio Buttons
- Select Boxes
- Buttons
- Image
- Hidden Input
- File Select
- Object controls



Elements of a Control

Required attributes of a control:

- Type
- Name

Optional attributes of a control:

- | | | | |
|------------|------------|-----------|--------------|
| •value | •src | •onfocus | •onmousedown |
| •checked | •alt | •onselect | •onmouseup |
| •disabled | •tabindex | •onchange | •onmouseover |
| •readonly | •accesskey | •onselect | •onmouseout |
| •size | •tabindex | •accept | •onkeypress |
| •maxlength | •usemap | •onblur | •onkeydown |
| •id | •ismap | •onclick | •onkeyup |



- Is probably the most commonly used form control. Syntax:
 - `<input type="text" name="email">`
- Options frequently used:
 - ID – for style or script reference
 - Size – set width
 - Maxlength – maximum number of characters for the form field



Password

- Password is a type of textbox where the form field entry is hidden by asterisks.
 - `<input type="password" name="comment" />`
- Commonly uses the same attributes as a text field.
- Less commonly used in both control types:
 - tabindex
 - accesskey
 - event triggers by key or mouse action



Textarea

- This control has a different syntax than the two other text input types.
 - `<textarea name="textfield2"></textarea>`
- Commonly used attributes:
 - cols – width of form field
 - rows – number of rows displayed
- Optional attributes specific to this control:
 - wrap – default is virtual
 - Virtual
 - Off
 - Physical



Radio Buttons

- When you want the user to make only one choice from a fixed set of option.
 - `<label>Radio Buttons:<input name="computerOwner" type="radio" value="yes" />I own a computer </label>`
- For multiple options make sure that the name element is the same for all items in the group.
 - `name="computerOwner"`



Checkboxes

- Use when you want to offer non-exclusive options.
 - `<input type="checkbox" name="checkbox" value="0-5" />`
- Required attribute: value
- Control specific attribute “checked” to set the default state of the checkbox:
 - `<label><input name="computers" type="checkbox" value="1-5" checked="checked" />1-5</label>`



Good Practice Tip

- When using a control that has multiple separate fields such as checkboxes and radio button use “fieldset” and legend to create a logical group.
 - Computers in Household: 0-5 6-11 12-25 25-50
 - <fieldset>
 - <legend>Computers in Household: </legend>
 - <label><input type="checkbox" name="computers" value="0-5" />0-5</label>
 - <label><input type="checkbox" name="computers1" value="6-11" />6-11</label>
 - <label><input type="checkbox" name="computers2" value="12-25" />12-25</label>
 - <label><input type="checkbox" name="computers3" value="25-50" />25-50</label>
 - </ fieldset>



Select Box

- A select box offers multiple options within one form control.
- Can be either:
 - Menu – displayed as a drop down box
 - `<select name="skill">`
`<option value="0" selected="selected">--select--`
`</option>`
 - List – set a multiple line display with the number of the items to chose from displayed
 - `<label>Skill Level
`
`<select name="skill1" size="6">`



Option Group

Within a select box you can group choices together with a non-selectable option such as:

```
<label for="favcity">Which is your favorite city?</label>
```

```
<select id="favcity" name="favcity">
```

```
<optgroup label="North America">
```

```
<option value="5">New York</option>
```

```
<option value="6">Houston</option>
```

```
<option value="7">Montreal</option>
```

```
</optgroup>
```

```
<optgroup label="Central America">
```

```
<option value="10">Cozamel</option>
```

```
<option value="11">Belize</option>
```

```
<option value="12">Panama</option>
```

```
</optgroup>
```


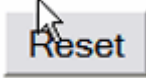
```
<optgroup label="South America">
```

```
<option value="8">Buenos Aires</option>
```

```
<option value="9">Portillo</option>
```

```
</optgroup>
```

```
</select>
```

- Two default button types:
 - `<input type="submit" name="Submit" value="Submit" />` A rectangular button with a thin border and the word "Submit" in the center.
 - `<input type="reset" name="reset" value="Reset" />` A rectangular button with a thin border and the word "Reset" in the center. A mouse cursor is hovering over the top-left corner of the button.
- Change the text on the button



Creates a graphical submit button. The value of the src attribute specifies the URI of the image that will decorate the button. For accessibility reasons, authors should provide alternate text for the image via the alt attribute.

```
<input name="imageField" type="image"  
  src="register.gif" width="72" height="20" border="0"  
  alt="register" />
```





- Use this control type to store information between client/server exchanges that would otherwise be lost due to the stateless nature of HTTP
- Common example might be to pass an item number or other identifier such as:
 - `<input name="sku" type="hidden" value="P-457" />`



■ File Select

- This control type allows the user to select files so that their contents may be submitted with a form.



- `<label for="file">Select File</label> <input type="file" name="file" id="file" />`
- Note: *using this control will not cause the file selected to be uploaded.*

- The object control is used to create custom controls and is beyond the scope of this presentation



Accessibility

- There are some attributes that are designed with accessibility in mind but also benefit usability:
 - Tabindex – specifies the order the cursor or focus moves from one field to the next. If it isn't specified the order maybe different depending on browser and OS.
 - Accesskey – while a recommendation of the W3C because of conflicts with operating system shortcuts, browser and program specific shortcut assignments it is difficult to find combinations that do not conflict. If you use them make sure you have a visible legend.



- **For more information on forms check:**
 - W3C Form <http://www.w3.org/TR/REC-html40/interact/forms.html#form-controls>
 - WebAim – Creating Accessible Forms
<http://www.webaim.org/techniques/forms/>