

## FIELDSET

FIELDSET is a block level element and defines a group of form elements as being logically related. By default, a border is drawn around the set of form elements to indicate that they are related. The width, type, and color of the border can be controlled with CSS, or it can be eliminated.

## LEGEND

Legend is used to define a title for the fieldset.

## EXAMPLE

```
<div>
<fieldset>
<legend>Student</legend>
<div class="left_col">
<label for="student_name">Name:</label>
</div>
<div class="right_col">
<input type="text" id="student_name" name="stu-name">
</div>
<div class="left_col">
<label for="student_id">ID</label>
</div>
<div class="right_col">
<input type="text" id="student_id" name="stu-id">
</div>
<div class="left_col">
<label for="student_phone">Phone:</label>
</div>
<div class="right_col">
<input type="text" id="student_phone" name="stu-phone">
</div>
<div class="left_col">
<label for="student_email">E-mail:</label>
</div>
<div class="right_col">
<input type="text" id="student_email" name="stu-email">
</div>
</fieldset>
</div>
```

**Note:** The preceding example will be spoken as follows by a screen reader — Student Name: edit, Student ID: edit, Student Phone: edit, Student E-mail edit.

## LABEL For

The <label for> tag defines a label to be associated with a particular form element. The for="uniqueID" should match the id="uniqueID" for the form element with which the label is to be associated.

## EXAMPLE

```
<div>
<label for="ssn">Social Security Number</label>
<input type="text" id="ssn" name="ss-number">
</div>

<div>
<label for="mi">I would like more information</label>
<input type="checkbox" id="mi" name="more-info">
</div>
```

```
<div>
<input type="radio" id="y1" name="pos-res" value="1">
<label for="y1">Yes</label>
</div>
```

## TABINDEX

TabIndex can be used to control the tab order for form elements. The tab sequence begins on the control with the smallest tabindex value and proceeds in increasing numerical order. A tabindex of zero can be used to make blocks of text and other “non-tabable” elements tabable, without effecting the tab order.

### EXAMPLE

```
<p tabindex="0">Pressing tab will move focus to this block of text. It will be read by screen-readers.</p>
```

## EVENT HANDLERS

An event handler causes a script to launch as the result of an action by the user. Some event handlers cannot be triggered without the use of a mouse, making them inaccessible to some users.

### EXAMPLE

Event Handler	Accessible
onBlur	Yes
onChange	Difficult
onClick	Yes
onFocus	Yes
onKeyDown	Yes
onKeyPress	Yes
onKeyUp	Yes
onMouseDown	No
onMouseOver	Difficult
onMouseout	No
onMouseup	No

## Resources

Web Accessibility Issues and Techniques demo site  
[http://accessibility.umn.edu/tl\\_files/accessibility/web\\_ac\\_techniques/index.html](http://accessibility.umn.edu/tl_files/accessibility/web_ac_techniques/index.html)

WebAIM: Creating Accessible Forms  
<http://webaim.org/techniques/forms/>

WebAIM: Usable and Accessible Form Validation and Error Recovery  
<http://webaim.org/techniques/formvalidation/>

Accessible Client-side Form Validation with HTML5 & WAI-ARIA  
<http://www.deque.com/accessible-client-side-form-validation-html5-wai-aria>

Accessible Form Validation with HTML5, WAI-ARIA, & jQuery Plugin  
<http://www.deque.com/accessible-client-side-form-validation-html5-wai-aria-jquery-validation-plugin>