

Chapter 27: Selection Menu Controls

Section 27.1: Select Menu

The `<select>` element generates a drop-down menu from which the user can choose an option.

```
<select name="">
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
  <option value="4">Four</option>
</select>
```

Changing the Size

You can change the size of the selection menu with the `size` attribute. A size of 0 or 1 displays the standard drop-down style menu. A size greater than 1 will convert the drop-down into a box displaying that many lines, with one option per line and a scrollbar in order to scroll through the available options.

```
<select name="" size="4"></select>
```

Multi-option Selection Menus

By default, users can only select a single option. Adding the `multiple` attribute allows users to select multiple options at once and submit all selected options with the form. Using the `multiple` attribute automatically converts the drop-down menu into a box as if it had a size defined. The default size when this occurs is determined by the specific browser you are using, and it is not possible to change it back to a drop-down style menu while allowing multiple selections.

```
<select name="" multiple></select>
```

When using the `multiple` attribute, there is a difference between using 0 and 1 for the size, whereas no difference exists when not using the attribute. Using 0 will cause the browser to behave in whatever default manner it is programmed to do. Using 1 will explicitly set the size of the resulting box to only one row high.

Section 27.2: Options

The options inside a selection menu are what the user will be selection. The normal syntax for an option is as follows:

```
<option>Some Option</option>
```

However, it's important to note that the text inside the `<option>` element itself is not always used, and essentially becomes the default value for attributes which are not specified.

The attributes which control the actual appearance and function of the option are `value` and `label`. The label represents the text which will be displayed in the drop-down menu (what you're looking at and will click on to select it). The value represents the text which will be sent along with form submission. If either of these values is omitted, it uses the text inside the element as the value instead. So the example we gave above could be "expanded" to this:

```
<option label="Some Option" value="Some Option">
```

Note the omission of the inside text and end tag, which are not required to actually construct an option inside the

menu. If they were included, the inside text would be ignored because both attributes are already specified and the text is not needed. However, you probably won't see a lot of people writing them this way. The most common way it's written is with a value that will be sent to the server, along with the inside text which eventually becomes the label attribute, like so:

```
<option value="option1">Some Option</option>
```

Selecting an option by default

You can also specify a certain option to be selected in the menu by default by attaching the `selected` attribute to it. By default, if no option is specified as selected in the menu, the first option in the menu will be selected when rendered. If more than one option has the `selected` attribute attached, then the last option present in the menu with the attribute will be the one selected by default.

```
<option value="option1" selected>Some option</option>
```

If you're using the attribute in a multi-option selection menu, then all the options with the attribute will be selected by default, and none will be selected if no options have the attribute.

```
<select multiple>
  <option value="option1" selected>Some option</option>
  <option value="option2" selected>Some option</option>
</select>
```

Section 27.3: Option Groups

You can neatly group your options within a selection menu in order to provide a more structured layout in a long list of options by using the `<optgroup>` element.

The syntax is very basic, by simply using the element with a `label` attribute to identify the title for the group, and containing zero or more options that should be within that group.

```
<select name="">
  <option value="milk">Milk</option>
  <optgroup label="Fruits">
    <option value="banana">Bananas</option>
    <option value="strawberry">Strawberries</option>
  </optgroup>
  <optgroup label="Vegetables" disabled>
    <option value="carrot">Carrots</option>
    <option value="zucchini">Zucchini</option>
  </optgroup>
</select>
```

When using option groups, not all options need to be contained within a group. As well, disabling an option group will disable all options within the group, and it is not possible to manually re-enable a single option within a disabled group.

Section 27.4: Datalist

The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element. It provides an "autocomplete" feature on `<input>` elements. Users will see a drop-down list of options as they write.

```
<input list="Languages">
```

```
<datalist id="Languages">
  <option value="PHP">
  <option value="Perl">
  <option value="Python">
  <option value="Ruby">
  <option value="C+">
</datalist>
```

Browser Support

Chrome	Edge	Mozilla	Safari	Opera
20.0	10.0	4.0	Not Supported	9.0