CSE 154

LECTURE 9: EVENTS AND TIMERS
Checkboxes: `<input>`

*yes/no choices that can be checked and unchecked (inline)*

<table>
<thead>
<tr>
<th>Checkbox</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lettuce</td>
<td>Lettuce</td>
</tr>
<tr>
<td></td>
<td>tomato</td>
<td>Tomato</td>
</tr>
<tr>
<td></td>
<td>pickles</td>
<td>Pickles</td>
</tr>
</tbody>
</table>

- none, 1, or many checkboxes can be checked at same time
- when sent to server, any checked boxes will be sent with value on:
- use checked="checked" attribute in HTML to initially check the box
Radio buttons: `<input>`

sets of mutually exclusive choices (inline)

```html
<input type="radio" name="cc" value="visa" checked="checked" /> Visa
<input type="radio" name="cc" value="mastercard" /> MasterCard
<input type="radio" name="cc" value="amex" /> American Express
```

- grouped by name attribute (only one can be checked at a time)
- must specify a value for each one or else it will be sent as value on
Text labels: `<label>`

- `<label><input type="radio" name="cc" value="visa" checked="checked" /> Visa</label>`
- `<label><input type="radio" name="cc" value="mastercard" /> MasterCard</label>`
- `<label><input type="radio" name="cc" value="amex" /> American Express</label>`

- associates nearby text with control, so you can click text to activate control
- can be used with checkboxes or radio buttons
- label element can be targeted by CSS style rules
Drop-down list: `<select>`, `<option>`

menus of choices that collapse and expand (inline)

```html
<select name="favoritecharacter">
  <option>Jerry</option>
  <option>George</option>
  <option selected="selected">Kramer</option>
  <option>Elaine</option>
</select>
```

- option element represents each choice
- select optional attributes: disabled, multiple, size
- optional selected attribute sets which one is initially chosen

output

Kramer  Submit Query
Using `<select>` for lists

```html
<select name="favoritecharacter[]" size="3" multiple="multiple">
  <option>Jerry</option>
  <option>George</option>
  <option>Kramer</option>
  <option>Elaine</option>
  <option selected="selected">Newman</option>
</select>
```

- optional multiple attribute allows selecting multiple items with shift- or ctrl-click
  - must declare parameter's name with [] if you allow multiple selections
- option tags can be set to be initially selected
Option groups: `<optgroup>`

```html
<select name="favoritecharacter">
  <optgroup label="Major Characters">
    <option>Jerry</option>
    <option>George</option>
    <option>Kramer</option>
    <option>Elaine</option>
  </optgroup>
  <optgroup label="Minor Characters">
    <option>Newman</option>
    <option>Susan</option>
  </optgroup>
</select>
```

- What should we do if we don't like the bold appearance of the optgroups?
Grouping input: `<fieldset>`, `<legend>`

*groups of input fields with optional caption (block)*

```html
<fieldset>
  <legend>Credit cards:</legend>
  <input type="radio" name="cc" value="visa" checked="checked" /> Visa
  <input type="radio" name="cc" value="mastercard" /> MasterCard
  <input type="radio" name="cc" value="amex" /> American Express
</fieldset>
```

- Visa ○ MasterCard ○ American Express

- fieldset groups related input fields, adds a border; legend supplies a caption
Styling form controls

- attribute selector: matches only elements that have a particular attribute value
- useful for controls because many share the same element (input)
The `innerHTML` property

```
<button onclick="addText();">Click me!</button>
<span id="output">Hello </span>
```

```js
function addText() {
    var span = document.getElementById("output");
    span.innerHTML += " bro";
}
```

- can change the text inside most elements by setting the `innerHTML` property
Abuse of innerHTML

// bad style!
var paragraph = document.getElementById("welcome");
paragraph.innerHTML =
  "</p>text and <a href="page.html">link</a"; 

- `innerHTML` can inject arbitrary HTML content into the page
- however, this is prone to bugs and errors and is considered poor style
- we forbid using `innerHTML` to inject HTML tags; inject plain text only
  - (later, we'll see a better way to inject content with HTML tags in it)