CSE 154

LECTURE 9: SUBMITTING DATA (POST)

Common UI control errors

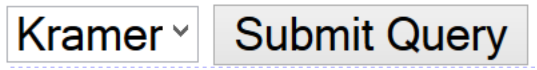
- “I changed the form's HTML code ... but when I refresh, the page doesn't update!”
  - By default, when you refresh a page, it leaves the previous values in all form controls
  - it does this in case you were filling out a long form and needed to refresh/return to it
  - if you want it to clear out all UI controls' state and values, you must do a full refresh
    - Firefox: Shift-Ctrl-R
    - Mac: Shift-Command-R

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>
```

HTML

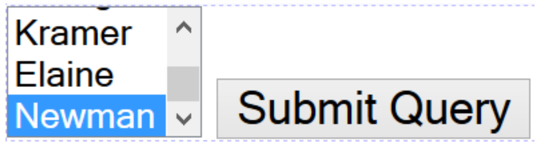


output

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

Using <select> for lists

```
<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>
```
HTML



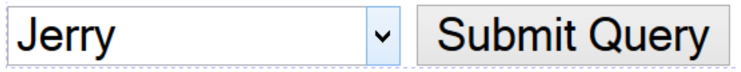
output

- 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>
```

HTML



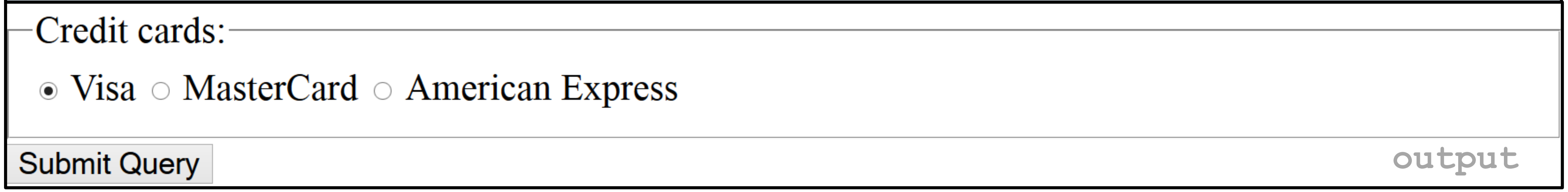
output

- 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>
```
HTML


Credit cards:

● Visa ○ MasterCard ○ American Express

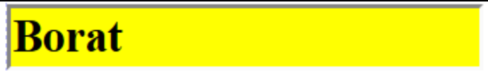
output

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

Styling form controls

```
element[attribute="value"] {
   property : value;
   property : value;
   ...
   property : value;
}                                                   CSS
```

```
input[type="text"] {
  background-color: yellow;
  font-weight: bold;
}                                                   CSS
```

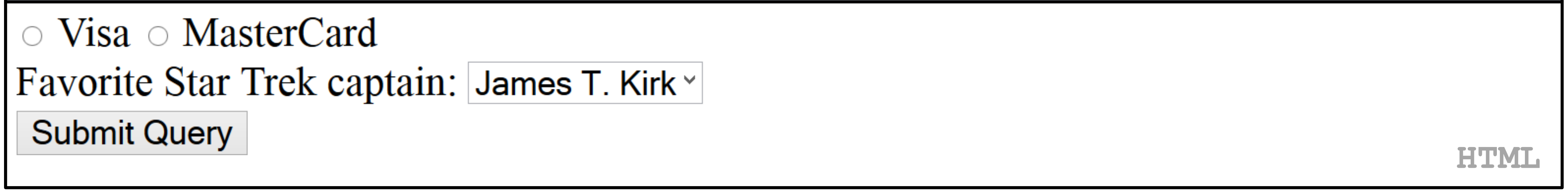

Borat

output

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

Problems with submitting data

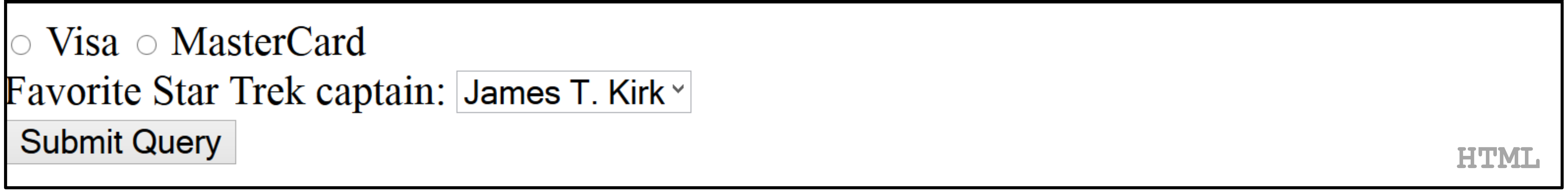
```html
<label><input type="radio" name="cc" /> Visa</label>
<label><input type="radio" name="cc" /> MasterCard</label> <br />
Favorite Star Trek captain:
<select name="startrek">
  <option>James T. Kirk</option>
  <option>Jean-Luc Picard</option>
</select> <br />
```
HTML



- this form submits to our handy params.php tester page
- the form may look correct, but when you submit it...
- [cc] => on, [startrek] => Jean-Luc Picard

The value attribute

```html
<label><input type="radio" name="cc" value="visa" /> Visa</label>
<label><input type="radio" name="cc" value="mastercard" />
     MasterCard</label> <br />
Favorite Star Trek captain:
<select name="startrek">
  <option value="kirk">James T. Kirk</option>
  <option value="picard">Jean-Luc Picard</option>
</select> <br />
```
HTML


○ Visa ○ MasterCard
Favorite Star Trek captain: James T. Kirk

- value attribute sets what will be submitted if a control is selected
- [cc] => visa, [startrek] => picard

URL-encoding

- certain characters are not allowed in URL query parameters:
  - examples: " ", "/", "=", "&"
- when passing a parameter, it is URL-encoded (reference table)
  - “Allison's cool!?" → “Allison%27s+cool%3F%21"
- you don't usually need to worry about this:
  - the browser automatically encodes parameters before sending them
  - the PHP $_GET and $_POST arrays automatically decode them
  - ... but occasionally the encoded version does pop up (e.g. in Firebug)

Submitting data to a web server

- though browsers mostly retrieve data, sometimes you want to submit data to a server
  - Hotmail: Send a message
  - Flickr: Upload a photo
  - Google Calendar: Create an appointment
- the data is sent in HTTP requests to the server
  - with HTML forms
  - with **Ajax** (seen later)
- the data is placed into the request as parameters

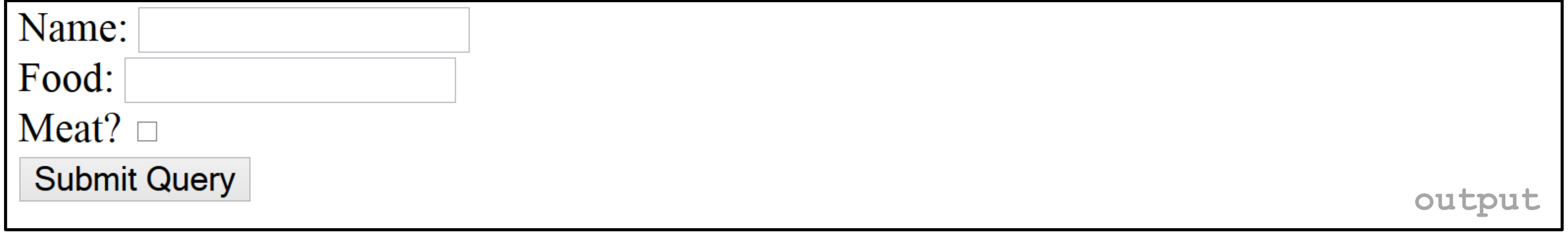
HTTP GET vs. POST requests

- GET : asks a server for a page or data
  - if the request has parameters, they are sent in the URL as a query string
- POST : submits data to a web server and retrieves the server's response
  - if the request has parameters, they are embedded in the request's HTTP packet, not the URL
- For submitting data to be saved, POST is more appropriate than GET
  - GET requests embed their parameters in their URLs
  - URLs are limited in length (~ 1024 characters)
  - URLs cannot contain special characters without encoding
  - private data in a URL can be seen or modified by users

Form POST example

```
 <form action="http://foo.com/app.php" method="post">
   <div>
     Name: <input type="text" name="name" /> <br />
     Food: <input type="text" name="meal" /> <br />
     <label>Meat? <input type="checkbox" name="meat" /></label>
<br />
     <input type="submit" />
   <div>
 </form>
```

HTML


Name:
Food:
Meat? ☐
Submit Query

output

GET or POST?

```php
if ($_SERVER["REQUEST_METHOD"] == "GET") {
  # process a GET request
  ...
} elseif ($_SERVER["REQUEST_METHOD"] == "POST") {
  # process a POST request
  ...
}                                                    PHP
```

- some PHP pages process both GET and POST requests
- to find out which kind of request we are currently processing,
- look at the global $_SERVER array's "REQUEST_METHOD" element

Including files: include

```
include("filename");                                        PHP
```

```
include("header.html");
include("shared-code.php");                                 PHP
```

- inserts the entire contents of the given file into the PHP script's output page
- encourages modularity
- useful for defining reused functions needed by multiple pages
- related: include_once, require, require_once