

Core Servlets and JavaServer Pages / 2e
Volume 1: Core Technologies
Marty Hall • Larry Brown

Processing HTML Forms

Agenda

- **Sending data from forms**
- **The FORM element**
- **Text controls**
- **Push buttons**
- **Check boxes and radio buttons**
- **Combo boxes and list boxes**
- **File upload controls**
- **Server-side image maps**
- **Hidden fields**
- **Grouping controls**
- **Tab ordering**

How HTML Forms Transmit Data

- **HTML contains a variety of user interface controls that enable the collection of data from a Web page**
 - Each control has a name and value
 - A URL is associated with the form; this is where the names and values of the controls, in the form `name1=value1&name2=value2...nameN=valueN`, are sent for processing
- **The string can be sent either via a GET or POST method**
 - The GET request appends the form data to the end of the specified URL after a question mark
 - The POST request sends the data after the HTTP request headers and a blank line

Sending Data with GET

...

```
<BODY BGCOLOR="#FDF5E6">
```

```
<H2 ALIGN="CENTER">A Sample Form Using GET</H2>
```

```
<FORM ACTION="http://localhost:8088/SomeProgram">
```

```
<CENTER>
```

```
First name:
```

```
<INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
```

```
Last name:
```

```
<INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
```

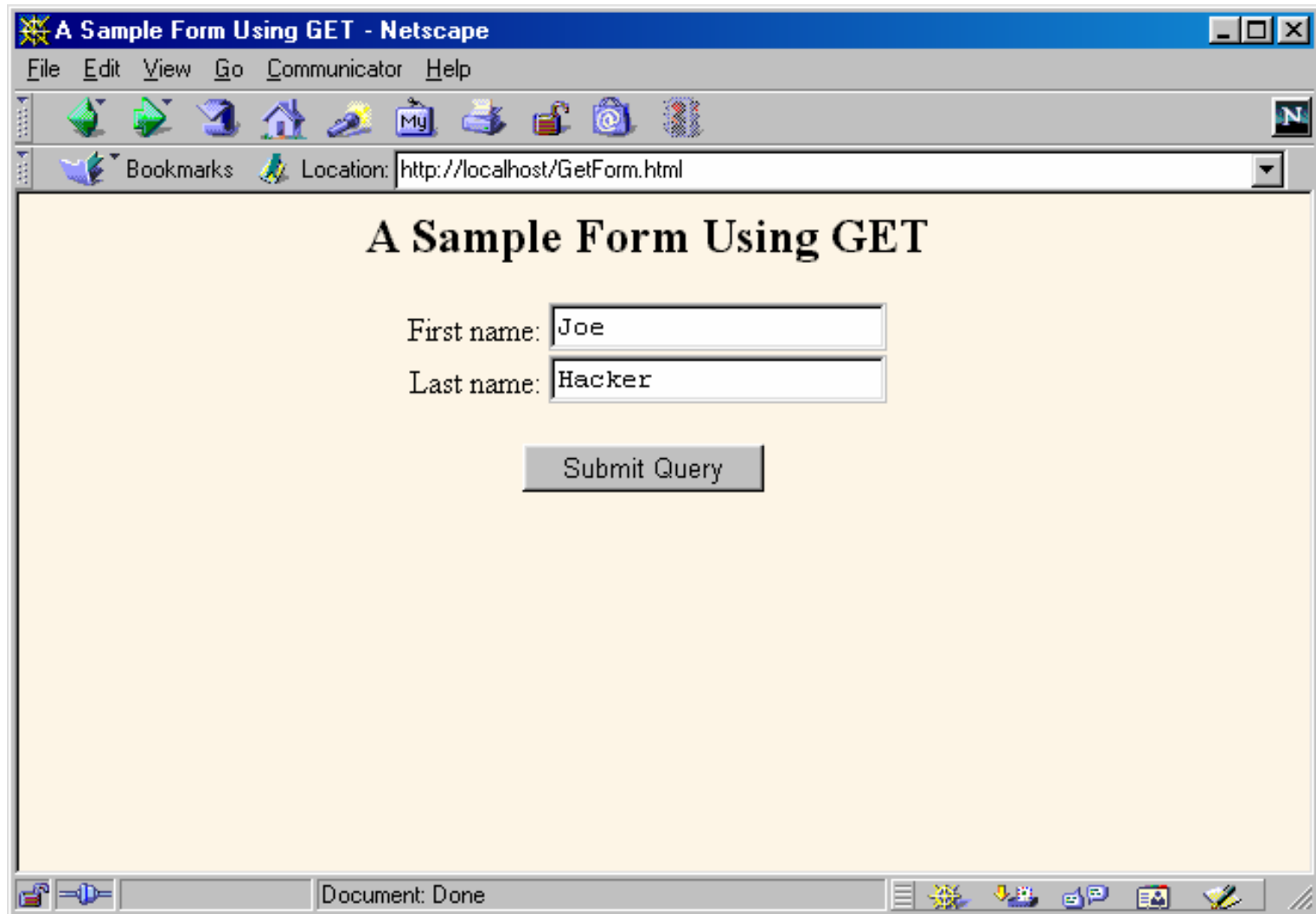
```
<INPUT TYPE="SUBMIT">
```

```
</CENTER>
```

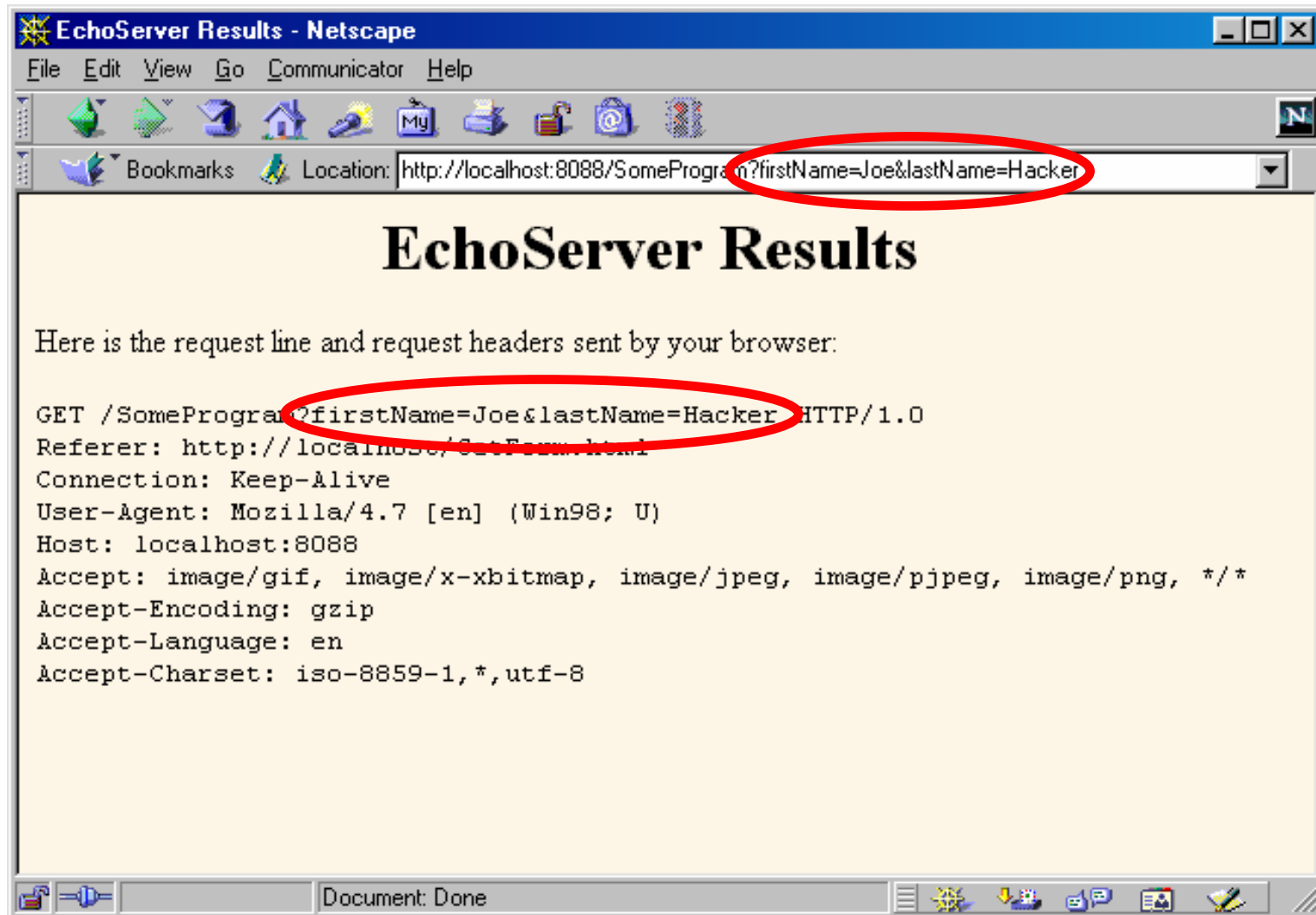
```
</FORM>
```

```
</BODY></HTML>
```

Initial Result



Submission Result



Sending Data with POST

...

```
<BODY BGCOLOR="#FDF5E6">
```

```
<H2 ALIGN="CENTER">A Sample Form Using POST</H2>
```

```
<FORM ACTION="http://localhost:8088/SomeProgram"  
      METHOD="POST">
```

```
<CENTER>
```

```
First name:
```

```
<INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
```

```
Last name:
```

```
<INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
```

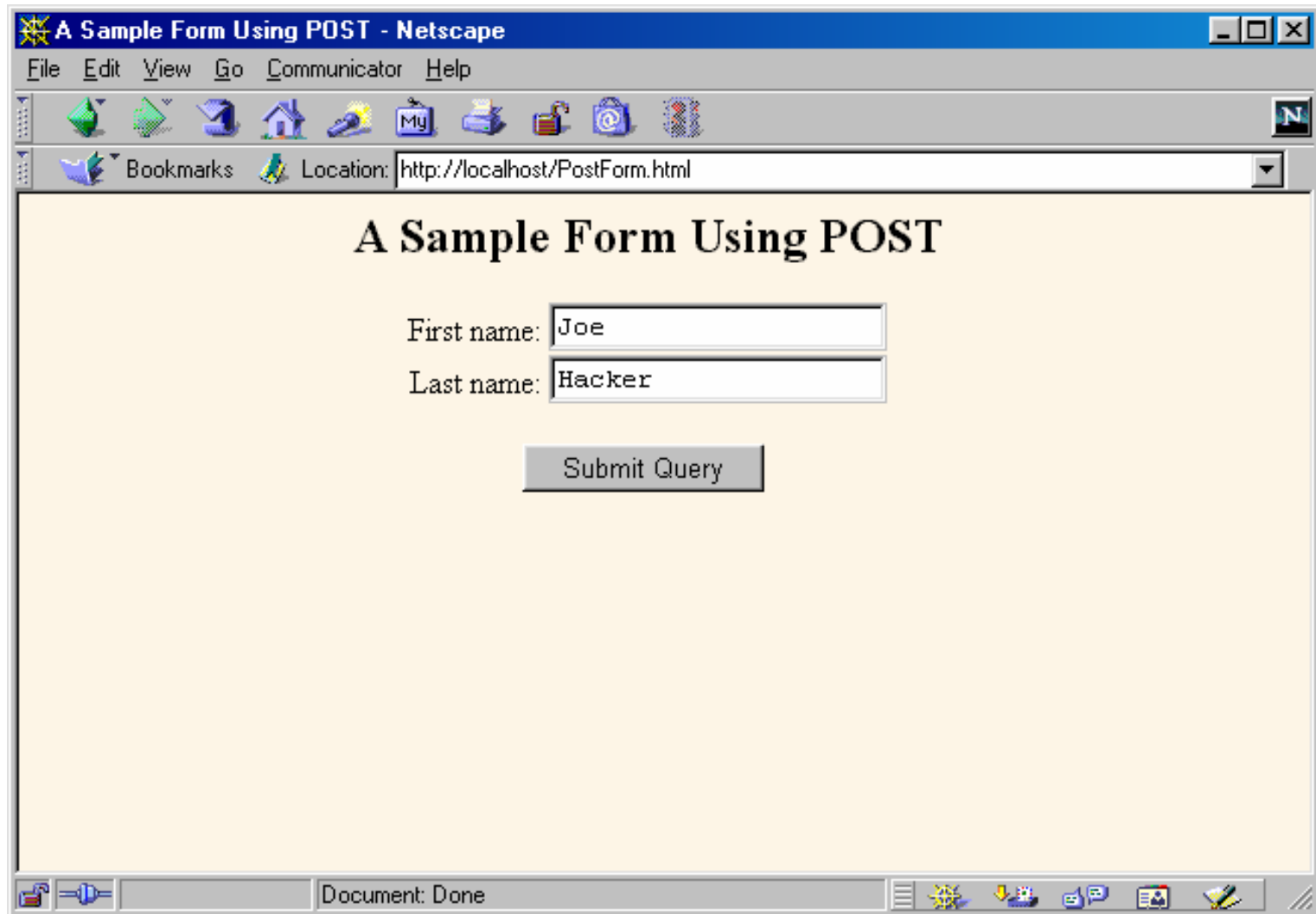
```
<INPUT TYPE="SUBMIT">
```

```
</CENTER>
```

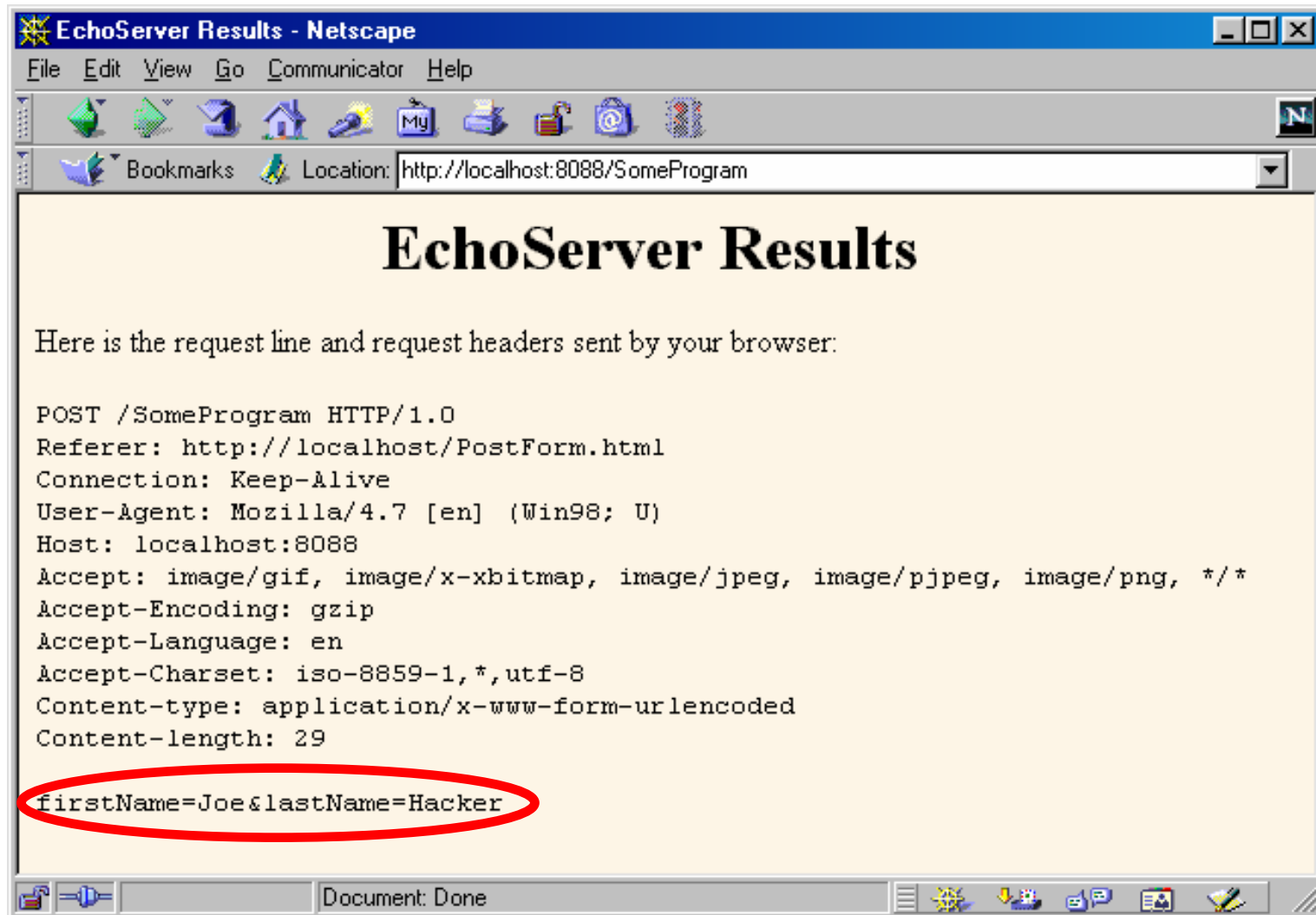
```
</FORM>
```

```
</BODY></HTML>
```

Initial Result



Submission Result



The screenshot shows a Netscape browser window titled "EchoServer Results - Netscape". The address bar displays "http://localhost:8088/SomeProgram". The main content area shows the following text:

```
EchoServer Results

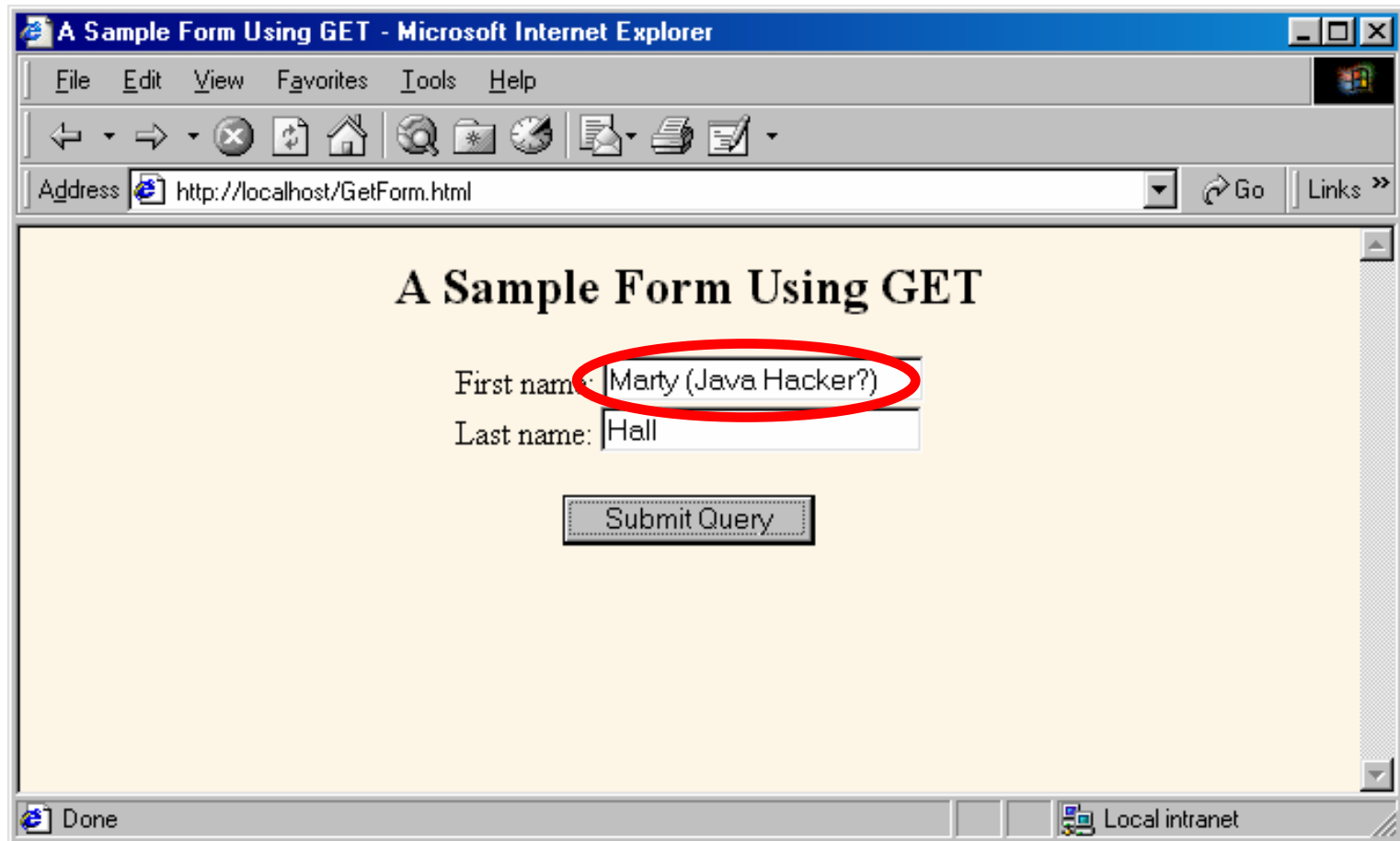
Here is the request line and request headers sent by your browser:

POST /SomeProgram HTTP/1.0
Referer: http://localhost/PostForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1, *, utf-8
Content-type: application/x-www-form-urlencoded
Content-length: 29

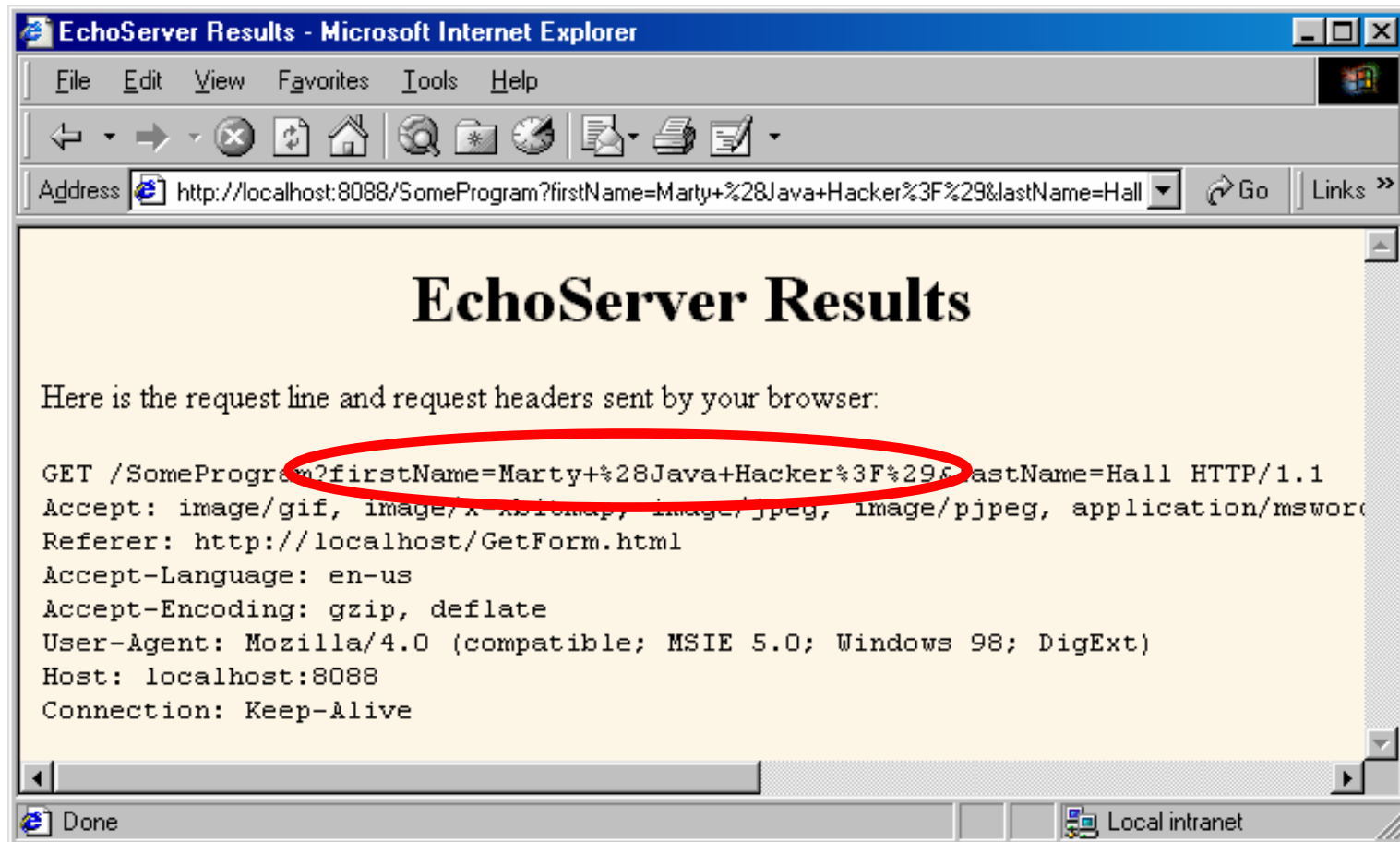
firstName=Joe&lastName=Hacker
```

The text "firstName=Joe&lastName=Hacker" is circled in red.

URL Encoding: Original Form



URL Encoding: Result



Text Controls

- **Textfields**

- `<INPUT TYPE="TEXT" NAME="..." ...>`
 - `VALUE` can give original value

- **Password Fields**

- `<INPUT TYPE="PASSWORD" NAME="..." ...>`
 - *Always use POST*

Enter Password:

- **Text Areas**

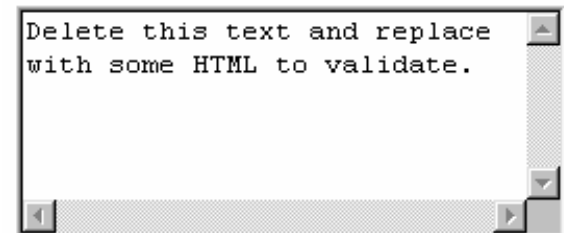
- `<TEXTAREA NAME="..."`
`ROWS="..." COLS="...">`

...

`</TEXTAREA>`

- Interpretation of regular HTML tags turned off between `<TEXTAREA...>` and `</TEXTAREA>`

Enter some HTML:

A screenshot of a text area control. The text inside the text area is "Delete this text and replace with some HTML to validate." The text area has a scroll bar on the right side and a small icon in the bottom right corner.

Push Buttons

- **Submit Buttons**

- `<INPUT TYPE="SUBMIT" ...>`
 - Use `NAME` if you have multiple buttons
 - Use `VALUE` to change button's label

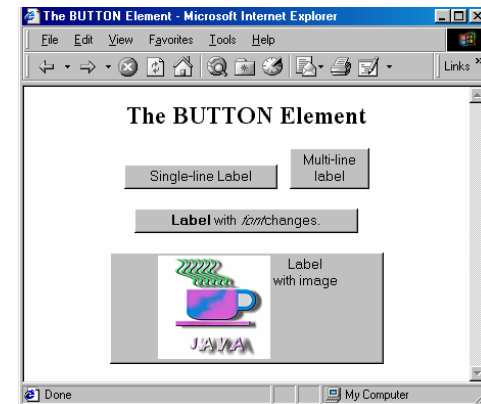


- **Reset Buttons**

- `<INPUT TYPE="RESET" ...>`
 - Use `VALUE` to change button's label

- **JavaScript Buttons**

- `<INPUT TYPE="BUTTON"`
`onClick="someJavaScriptFunction()" ...>`



- **Fancy Buttons**

- `<BUTTON TYPE="SUBMIT" ...>HTML</BUTTON>`
 - Internet Explorer and Netscape 6 only

Using Multiple Submit Buttons

```
<CENTER>
Item:
<INPUT TYPE="TEXT" NAME="Item" VALUE="256MB SIMM"><BR>
<INPUT TYPE="SUBMIT" NAME="Add"
      VALUE="Add Item to Cart">
<INPUT TYPE="SUBMIT" NAME="Delete"
      VALUE="Delete Item from Cart">
</CENTER>
```

Item:

Add Item to Cart

Delete Item from Cart

Check Boxes

- **Format**

- `<INPUT TYPE="CHECKBOX" NAME="..." ...>`
 - The `CHECKED` attribute makes it initially checked
 - **Name/value pair sent only if checkbox is checked when form is submitted**

- **Example code**

```
<P>
```

```
<INPUT TYPE="CHECKBOX" NAME="noEmail" CHECKED>
```

```
Check here if you do <I>not</I> want to  
get our email newsletter
```

- **Example result**

Check here if you do *not* want to get our email newsletter

Radio Buttons

- **Format**

- `<INPUT TYPE = "RADIO" NAME = "... " VALUE = "... " ... >`

- All radio buttons in a group should have same NAME
 - Only one button in a group can be pressed; pressing a different one causes previous one to pop out

- **Example**

```
<DL>
  <DT>Credit Card:
  <DD><INPUT TYPE="RADIO" NAME="creditCard"
        VALUE="visa">
    Visa
  <DD><INPUT TYPE="RADIO" NAME="creditCard"
        VALUE="mastercard">
    Master Card
    ...
</DL>
```

Credit Card:

- Visa
- Master Card
- Java Smart Card
- American Express
- Discover

Combo Boxes

- **Format**

- SELECT gives NAME
- OPTION gives VALUE

Favorite language:

- **Example**

Favorite language:

Favorite language:

```
<SELECT NAME="language">  
  <OPTION VALUE="c">C  
  <OPTION VALUE="c++">C++  
  <OPTION VALUE="java" SELECTED>Java  
  <OPTION VALUE="lisp">Lisp  
  <OPTION VALUE="perl">Perl  
  <OPTION VALUE="smalltalk">Smalltalk  
</SELECT>
```

C
C++
Java
Lisp
Perl
Smalltalk

List Boxes

- **Format**

- Identical to combo boxes, but specify MULTIPLE

- **Example**

Languages you know:


```
<SELECT NAME="language" MULTIPLE>  
  <OPTION VALUE="c">C  
  <OPTION VALUE="c++">C++  
  <OPTION VALUE="java" SELECTED>Java  
  <OPTION VALUE="lisp">Lisp  
  <OPTION VALUE="perl" SELECTED>Perl  
  <OPTION VALUE="smalltalk">Smalltalk  
</SELECT>
```

Languages you know:



Other Controls and Options

- **File upload controls**
 - Lets user select a file and send it to the server
- **Server-side image maps**
 - User clicks on an image and form gets submitted.
 - Form data gets sent as *name.x=x-pos&name.y=y-pos*
- **Hidden fields**
 - Preset NAME and VALUE sent with form submission..
- **Grouping Controls**
 - FIELDSET lets you visually group forms.
 - Internet Explorer and Netscape 6 only.
- **Tab order control**
 - TABINDEX (Internet Explorer and Netscape 6 only)

Summary

- **General process**

- FORM uses ACTION to specify base URL
- Input elements each have a NAME
- User enters values
- When form submitted, URL is
baseURL?name1=value1&name2=value2&...
- For POST requests, name/value pairs sent on separate line
(not part of URL)

- **Textfields**

- <INPUT TYPE="TEXT" ...>

- **Submit Buttons**

- <INPUT TYPE="SUBMIT" ...>