

Fall 2004 Handout 8

Copy a web page to i5.nyu.edu

To copy a web page whose URL is `http://www.host.com/directory/file.html` to the `public_html` subdirectory of the home directory of your account on `i5.nyu.edu`,

```
cd
cd public_html
pwd

lynx -source http://www.host.com/directory/file.html > file.html
ls -l file.html
-rw-----  1 abc1234  users          6469 Apr  9 07:52 file.html

chmod 644 file.html
ls -l file.html
-rw-r--r--  1 abc1234  users          6469 Apr  9 07:52 file.html
```

The URL of the new copy on `i5.nyu.edu` will then be

`http://i5.nyu.edu/~abc1234/file.html`, where `abc1234` is your `i5.nyu.edu` loginname.

Use the same `lynx` command to copy a file of any type to `i5.nyu.edu`: `.html`, `.gif`, `.jpg`, etc.

Assign a new value to a property of an object

```
1 <HTML>
2 <HEAD>
3 <TITLE>Document has moved (the old way).</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Document has moved (the old way).</H1>
7 The new URL is
8 <BR>
9 <a href = "http://www.hrm.org/">http://www.hrm.org/</a>
10 </BODY>
11 </HTML>
```

Instead of saying `window.location`, you can just say `location` (Flanagan pp. 176–177; Wagner p. 244). Assigning a new value to this property will point your browser at a new URL (Flanagan pp. 213–214; Wagner pp. 300–301).

```
1 <HTML>
2 <HEAD>
3 <SCRIPT TYPE = "text/javascript">
4 location = "http://www.hrm.org/";
5 </SCRIPT>
6 </HEAD>
7 </HTML>
```

▼ Homework 8.1: for people whose home page is on another machine

Create a home page for yourself on `i5.nyu.edu` if you have not already done so. Put the above `location` = into it to sweep a visitor directly to your home page on the other machine.



Multi-way if: Flanagan, pp. 71–72; Wagner, pp. 140–144

Instead of saying `window.prompt`, you can just say `prompt`. See Flanagan p. 13 for an example of the `parseInt isNaN` combination.

```

1 <HTML>
2 <HEAD>
3 <TITLE>A multi-way if</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>A multi-way if</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 var digits = prompt("Please type a number.", "");
10 var d = parseInt(digits);
11
12 if (isNaN(d)) {
13     document.write ("Sorry, '" + digits + "' is not a number.");
14 } else if (d == 4) {
15     document.write ("Four colly birds");
16 } else if (d == 3) {
17     document.write ("Three French hens");
18 } else if (d == 2) {
19     document.write ("Two turtledoves");
20 } else if (d == 1) {
21     document.write ("A partridge in a pear tree");
22 } else {
23     document.write ("The number " + d + " is out of range.");
24 }
25 </SCRIPT>
26 </BODY>
27 </HTML>

```

After line 10, we could add

```

1 if (digits == null) {
2     document.write ("You have to say something; you can't just press Cancel."
3 );

```

Loop example: Flanagan, pp. 74–75; Wagner, pp. 145–148

```

1 <HTML>
2 <HEAD>
3 <TITLE>100 Bottles of Beer on the Wall</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>100 Bottles of Beer on the Wall</H1>
7 <SCRIPT TYPE = "text/javascript">
8 for (var b = 100; b >= 1; b = b - 1) {
9     document.write(
10         "<P>", b, " bottles of beer on the wall\n",

```

```

11     "<BR>", b, " bottles of beer\n",
12     "<BR>If one of those bottles should happen to fall\n",
13     "<BR>", b - 1, " bottles of beer on the wall\n");
14 }
15 </SCRIPT>
16 </BODY>
17 </HTML>

```

The raw output begins

```

1 <P>100 bottles of beer on the wall
2 <BR>100 bottles of beer
3 <BR>If one of those bottles should happen to fall
4 <BR>99 bottles of beer on the wall
5 <P>99 bottles of beer on the wall
6 <BR>99 bottles of beer
7 <BR>If one of those bottles should happen to fall
8 <BR>98 bottles of beer on the wall
9 <P>98 bottles of beer on the wall

```

The newlines

Without the `\n`'s, the raw output would be

```
1 <P>100 bottles of beer on the wall<BR>100 bottles of beer<BR>If one of those bottles should
```

But the `\n`'s have no effect on the rendered output that you see in your browser and are therefore usually omitted from JavaScript programs. To make the raw output easier to read, however, I'll insert newlines into it even though the program does not output them.

```
document.writeln("hello");
```

does the same thing as

```
document.write("hello\n");
```

Loop example: output the COLOR attribute for the tag

```

1 <HTML>
2 <HEAD>
3 <TITLE>The COLOR attributes of the &lt;FONT&gt; tag</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>The
7 <A HREF =
8 "http://www.w3.org/TR/REC-html40/present/graphics.html#edef-color-FONT">COLOR</A>
9 attribute of the
10 <A HREF =
11 "http://www.w3.org/TR/REC-html40/present/graphics.html#edef-FONT">&lt;FONT&gt;</A>
12 tag</H1>
13 <SCRIPT TYPE = "text/javascript">
14 for (var r = 31; r <= 255; r = r + 32) {
15     var red = r.toString(16);
16
17     //Make sure that red is two digits.
18     if (red.length == 1) {
19         red = "0" + red;
20     }

```

```

21
22     document.write("<BR><FONT COLOR = #", red, "0000>",
23         "Pack my box with five dozen liquor jugs.</FONT>");
24 }
25 </SCRIPT>
26 </BODY>
27 </HTML>

```

The raw output is

```

1 <BR><FONT COLOR = #1f0000>Pack my box with five dozen liquor jugs.</FONT>
2 <BR><FONT COLOR = #3f0000>Pack my box with five dozen liquor jugs.</FONT>
3 <BR><FONT COLOR = #5f0000>Pack my box with five dozen liquor jugs.</FONT>
4 <BR><FONT COLOR = #7f0000>Pack my box with five dozen liquor jugs.</FONT>
5 <BR><FONT COLOR = #9f0000>Pack my box with five dozen liquor jugs.</FONT>
6 <BR><FONT COLOR = #bf0000>Pack my box with five dozen liquor jugs.</FONT>
7 <BR><FONT COLOR = #df0000>Pack my box with five dozen liquor jugs.</FONT>
8 <BR><FONT COLOR = #ff0000>Pack my box with five dozen liquor jugs.</FONT>

```

Loop example: output HEIGHT and WIDTH attributes for the tag

The original image is 123 pixels high, 98 pixels wide. `p` stands for “percent”.

```

1 <HTML>
2 <HEAD>
3 <TITLE>The HEIGHT and WIDTH attributes of the &lt;IMG&gt; tag</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>The
7 <a href =
8 "http://www.w3.org/TR/REC-html40/struct/objects.html#edef-height-IMG">HEIGHT</a>
9 and
10 <a href =
11 "http://www.w3.org/TR/REC-html40/struct/objects.html#edef-width-IMG">WIDTH</a>
12 attributes of the
13 <a href =
14 "http://www.w3.org/TR/REC-html40/struct/objects.html#edef-IMG">&lt;IMG&gt;</a>
15 tag</H1>
16 <SCRIPT TYPE = "text/javascript">
17 for (var p = 100; p > 0; p = p - 20) {
18     document.write('<IMG SRC = "~/mm64/markface.gif"',
19         "HEIGHT = ", Math.round(123 * p / 100),
20         " WIDTH = ", Math.round(98 * p / 100), ">");
21 }
22 </SCRIPT>
23 </BODY>
24 </HTML>

```

The raw output is

```

1 <IMG SRC = "~/mm64/markface.gif" HEIGHT = 123 WIDTH = 98>
2 <IMG SRC = "~/mm64/markface.gif" HEIGHT = 98 WIDTH = 78>
3 <IMG SRC = "~/mm64/markface.gif" HEIGHT = 74 WIDTH = 59>
4 <IMG SRC = "~/mm64/markface.gif" HEIGHT = 49 WIDTH = 39>
5 <IMG SRC = "~/mm64/markface.gif" HEIGHT = 25 WIDTH = 20>

```

Loop through an array of words: Flanagan pp. 121–131; Wagner pp. 386–392;

```
1 <HTML>
2 <HEAD>
3 <TITLE>Loop through an array of words</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Loop through an
7 <a href =
8 "http://developer.netscape.com/docs/manuals/communicator/jsref/core1.htm#1080770">array</a>
9 of words</H1>
10 <SCRIPT TYPE = "text/javascript">
11 var a = new Array(
12     "monkey",    //0
13     "rooster",  //1
14     "dog",       //2
15     "pig",       //3
16     "rat",       //4
17     "ox",        //5
18     "tiger",    //6
19     "hare",      //7
20     "dragon",    //8
21     "snake",    //9
22     "horse",    //10
23     "sheep"     //11
24 );
25
26 document.write (
27     "The length of the array is ", a.length, ".",
28     "<BR>The first element of the array is ", a[0], ".",
29     "<BR>The second element of the array is ", a[1], ".",
30     "<BR>The third element of the array is ", a[2], ".");
31
32 for (var i = 0; i < a.length; i = i + 1) {
33     document.write("<BR>", a[i]);
34 }
35
36 var d = new Date();
37 var y = d.getYear() + 1900;
38 document.write("<P>", y, " is the year of the ", a[y % a.length], ".");
39 </SCRIPT>
40 </BODY>
41 </HTML>
```

The raw output of the **for** loop is

```
1 <BR>monkey
2 <BR>rooster
3 <BR>dog
4 <BR>pig
5 <BR>rat
6 <BR>ox
7 <BR>tiger
8 <BR>hare
9 <BR>dragon
10 <BR>snake
```

```
11 <BR>horse
12 <BR>sheep
```

To fix (or at least postpone) the year 2000 problem, change the above line 39 to

```
1 var y = d.getYear();
2
3 if (y >= 50) {
4     //year in the range 1950 to 1999 inclusive
5     y = y + 1900;
6 } else {
7     //year in the range 2000 to 2049 inclusive
8     y = y + 2000;
9 }
```

If they hadn't invented arrays, line 39 would have to be written

```
1 if (y % a.length == 0) {
2     document.write("<P>", y, " is the year of the monkey.");
3 } else if (y % a.length == 1) {
4     document.write("<P>", y, " is the year of the rooster.");
5 } else if (y % a.length == 2) {
6     document.write("<P>", y, " is the year of the dog.");
7 } else etc.
```

Sort an array

NEXT TIME: Sort the above array into alphabetical order with

```
1 a.sort();
```

Then sort in order of increasing or decreasing length; reverse alphabetical order.

Nested loops

```
1 <HTML>
2 <HEAD>
3 <TITLE>Nested loops</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Nested loops</H1>
7 <SCRIPT TYPE = "text/javascript">
8 var meat = new Array("Beef", "Chicken", "Pork", "Shrimp");
9 var veggie = new Array(
10     "Black Bean Sauce",
11     "Broccoli",
12     "Garlic Sauce",
13     "Mixed Chinese Vegetables",
14     "Snow Peas"
15 );
16
17 for (var m = 0; m < meat.length; ++m) {
18     for (var v = 0; v < veggie.length; ++v) {
19         if (v == 0) {
20             document.write("<HR>");
21         } else {
22             document.write("<BR>");
23         }
24     }
25 }
```

```

24     document.write (meat[m], " with ", veggie[v]);
25     }
26 }
27 </SCRIPT>
28 </BODY>
29 </HTML>

```

The raw output of the `for` loop begins

```

1 <HR>Beef with Black Bean Sauce
2 <BR>Beef with Broccoli
3 <BR>Beef with Garlic Sauce
4 <BR>Beef with Mixed Chinese Vegetables
5 <BR>Beef with Snow Peas
6 <HR>Chicken with Black Bean Sauce
7 <BR>Chicken with Broccoli

```

▼ Homework 8.2: make a style sheet

Write a pair of nested `for` loops to output a short sentence in every combination of font and size. The outer loop should loop through the names of several fonts. (Store the names of the fonts in an **Array** named `font`.) The inner loop should loop through the possible sizes.

```
<FONT FACE = "HELVETICA" SIZE = 1>Pack my box with five dozen liquor jugs.</FONT>
```



Multiple cookies with arrays and for loops

```

1 <HTML>
2 <HEAD>
3 <TITLE>Multiple cookies</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Multiple cookies</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 var first; //the user's first name
10 var bg; //the user's favorite background color
11 var n; // the number of this visit
12
13 if (document.cookie == "") {
14     //This is the user's first visit.
15     first = prompt ("New user please register. What is your first name?", "");
16     bg = prompt ("What is your favorite background color?", "red");
17     n = 1;
18
19     document.cookie = "first=" + first;
20     document.cookie = "bg=" + bg;
21     document.cookie = "n=1";
22 } else {
23     //This is not the user's first visit.
24     document.write (
25         "<P>For your information, document.cookie now contains",
26         "<BR>", document.cookie,
27         "<P>Here are the contents of each element of the array c:"
28     );

```

```

29
30   var c = document.cookie.split(";");
31
32   for (var i = 0; i < c.length; i = i + 1) {
33       //Remove the leading blank, if any, from c[i].
34       if (c[i].charAt(0) == " ") {
35           c[i] = c[i].substring(1);
36       }
37       document.write ("<BR>", c[i]);
38
39       var pair = c[i].split("=");
40       if (pair[0] == "first") {
41           first = pair[1];
42       } else if (pair[0] == "bg") {
43           bg = pair[1];
44       } else if (pair[0] == "n") {
45           n = parseInt(pair[1]) + 1;
46           document.cookie = "n=" + n;
47       } else {
48           document.write ("<BR>unknown cookie ", pair[0]);
49       }
50   }
51 }
52
53 document.bgColor = bg;
54 document.write ("<P>Welcome, ", first, ", to visit number ", n, ".");
55 </SCRIPT>
56 </BODY>
57 </HTML>

```

At the start of Mark's second visit, `document.cookie` contains

```
first=Mark; bg=green; n=1
```

Note the space after each semicolon.

`c[0]` contains `first=Mark`

`c[1]` contains `bg=green`

`c[2]` contains `n=1`

To display an ordinal number,

```

1 document.write ("Welcome, ", first, ", to your ", n);
2
3 if (n % 10 == 1 && n % 100 != 11) {
4     document.write ("st");
5 } else if (n % 10 == 2 && n % 100 != 12) {
6     document.write ("nd");
7 } else if (n % 10 == 3 && n % 100 != 13) {
8     document.write ("rd");
9 } else {
10    document.write ("th");
11 }
12
13 document.write ("visit.");

```

In the real world, cookies are even more complicated for three reasons:

- (1) You have to make sure that the user's first name and favorite background color contain no blanks, semicolons, equal signs, etc. See the **escape** and **unescape** functions.
- (2) Since a browser can save only a limited number of cookies, you would normally put all three pieces of information into one big cookie.
- (3) The cookies shown above expire when you turn off your browser.

Event handlers in a client-side imagemap

```
1 <HTML>
2 <HEAD>
3 <TITLE>X52.9755 Spring 1998 Section 1 (Thursday) Class Photo</TITLE>
4 <META http-equiv = "Content-Script-Type" content = "text/javascript">
5 </HEAD>
6 <BODY>
7 <H1>JavaScript <a href = "/~mm64/x52.9755">X52.9755</a></H1>
8 <H2>Spring 1998, Section 1 (Thursday) Class Photo</H2>
9 Thursday, April 2, 1998
10 <BR>
11 <A HREF = "http://www.nyu.edu/infocenter/campus2/ws/region4.html">48
12 Cooper Square</A>,
13 room 208
14 <P>
15 Click on the tip of each person's nose.
16 Mark's <A HREF = "/~mm64/#touch">home page</A>
17 tells how to make your own touch-sensitive imagemap.
18 <P>
19 <MAP NAME = "map">
20
21 <AREA
22 SHAPE = CIRCLE
23 COORDS = 645,190,10
24 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?lqc4436"
25 onMouseOver = "status = 'Louise Covitt lqc4436'; return true;"
26 onMouseOut = "status = ''; return true;">
27
28 <AREA
29 SHAPE = CIRCLE
30 COORDS = 596,154,10
31 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?sqs8467"
32 onMouseOver = "status = 'Sam Sultan sqs8467'; return true;"
33 onMouseOut = "status = ''; return true;">
34
35 <AREA
36 SHAPE = CIRCLE
37 COORDS = 531,144,10
38 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?js559"
39 onMouseOver = "status = 'James Scranton js559'; return true;"
40 onMouseOut = "status = ''; return true;">
41
42 <AREA
43 SHAPE = CIRCLE
44 COORDS = 501,115,10
45 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?bk290"
46 onMouseOver = "status = 'Brian Kortland bk290'; return true;"
```

```
47 onMouseOut = "status = ''; return true;">
48
49 <AREA
50 SHAPE = CIRCLE
51 COORDS = 410,126,10
52 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?cgy204"
53 onMouseOver = "status = 'Chris J Yellen cgy204'; return true;"
54 onMouseOut = "status = ''; return true;">
55
56 <AREA
57 SHAPE = CIRCLE
58 COORDS = 381,188,10
59 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?cdc212"
60 onMouseOver = "status = 'Christopher D Cooper cdc212'; return true;"
61 onMouseOut = "status = ''; return true;">
62
63 <AREA
64 SHAPE = CIRCLE
65 COORDS = 360,128,10
66 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?ans210"
67 onMouseOver = "status = 'Allison N Strandwitz ans210'; return true;"
68 onMouseOut = "status = ''; return true;">
69
70 <AREA
71 SHAPE = CIRCLE
72 COORDS = 306,217,10
73 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?jc529"
74 onMouseOver = "status = 'Janetara Currie jc529'; return true;"
75 onMouseOut = "status = ''; return true;">
76
77 <AREA
78 SHAPE = CIRCLE
79 COORDS = 287,181,10
80 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?fsk202"
81 onMouseOver = "status = 'Francis S Kilroy fsk202'; return true;"
82 onMouseOut = "status = ''; return true;">
83
84 <AREA
85 SHAPE = CIRCLE
86 COORDS = 214,125,10
87 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?jaf226"
88 onMouseOver = "status = 'Jimmy A Fountain jaf226'; return true;"
89 onMouseOut = "status = ''; return true;">
90
91 <AREA
92 SHAPE = CIRCLE
93 COORDS = 219,199,10
94 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?mer221"
95 onMouseOver = "status = 'Mary E Radmore mer221'; return true;"
96 onMouseOut = "status = ''; return true;">
97
98 <AREA
99 SHAPE = CIRCLE
100 COORDS = 139,189,10
```

```
101 HREF = "http://members.aol.com/nickt15/project.html"
102 onMouseOver = "status = 'Nicholas Turi nt232'; return true;"
103 onMouseOut = "status = ''; return true;">
104
105 <AREA
106 SHAPE = CIRCLE
107 COORDS = 101,219,10
108 HREF = "http://www.stern.nyu.edu/~awong8/"
109 onMouseOver = "status = 'Alice So Kam Wong asw1'; return true;"
110 onMouseOut = "status = ''; return true;">
111
112 <AREA
113 SHAPE = CIRCLE
114 COORDS = 160,341,10
115 HREF = "mailto:jtl2@is6.nyu.edu"
116 onMouseOver = "status = 'James T Lap mailto:jtl2@is6.nyu.edu'; return true;"
117 onMouseOut = "status = ''; return true;">
118
119 <AREA
120 SHAPE = CIRCLE
121 COORDS = 280,284,10
122 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?hs317"
123 onMouseOver = "status = 'Heather Stone hs317'; return true;"
124 onMouseOut = "status = ''; return true;">
125
126 <AREA
127 SHAPE = CIRCLE
128 COORDS = 360,302,10
129 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?mm64"
130 onMouseOver = "status = 'Mark Stuart Meretzky mm64'; return true;"
131 onMouseOut = "status = ''; return true;">
132
133 <AREA
134 SHAPE = CIRCLE
135 COORDS = 403,269,10
136 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?erl206"
137 onMouseOver = "status = 'Elizabeth R Lanphier erl206'; return true;"
138 onMouseOut = "status = ''; return true;">
139
140 <AREA
141 SHAPE = CIRCLE
142 COORDS = 478,265,10
143 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?ren205"
144 onMouseOver = "status = 'Roger E Norvan ren205'; return true;"
145 onMouseOut = "status = ''; return true;">
146
147 <AREA
148 SHAPE = CIRCLE
149 COORDS = 575,261,10
150 HREF = "/cgi-bin/cgiwrap/~mm64/classphoto?eqh5517"
151 onMouseOver = "status = 'Eli Hernandez eqh5517'; return true;"
152 onMouseOut = "status = ''; return true;">
153
154 <AREA
```

```
155 SHAPE = CIRCLE
156 COORDS = 463,186,10
157 HREF = "http://www.torrelazur.com/sf/kelly/index.htm"
158 onMouseOver = "status = 'Kelly Lynch kql6531'; return true;"
159 onMouseOut = "status = ''; return true;">
160
161 <AREA
162 SHAPE = CIRCLE
163 COORDS = 463,186,10
164 HREF = "/cgi-bin/donothing"
165 onMouseOver =
166     "status = 'Unknown person at coordinates 463,186'; return true;"
167 onMouseOut = "status = ''; return true;">
168
169 </MAP>
170
171 <A HREF = "/cgi-bin/imagemap/~mm64/x52.9755/001981.map">
172 <IMG
173     SRC = "001981.jpg"
174     ISMAP
175     USEMAP = "#map"
176 ></A>
177
178 <H2>Not in photo, hidden, or unidentified</H2>
179 <OL>
180 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?ya213">ya213</a>
181 Yasser Akhtar
182 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?ef231">ef231</a>
183 Emily Field
184 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?mh395">mh395</a>
185 Mark Hayden
186 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?aj8">aj8</a>
187 Adrian Jones
188 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?nk263">nk263</a>
189 Nathaniel Kanner
190 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?bl227">bl227</a>
191 Barbara Lynn
192 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?am26">am26</a>
193 Ann McDermott
194 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?jjm9351">jjm9351</a>
195 Joseph J Matazzoni
196 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?rcm4093">rcm4093</a>
197 Richard C Miller
198 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?kcm206">kcm206</a>
199 K. C Mukherjee
200 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?ejr2">ejr2</a>
201 Elizabeth Reece
202 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?cws210">cws210</a>
203 Colin W Shearwood
204 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?bks1">bks1</a>
205 Brian K Stefans
206 <li><a href = "/cgi-bin/cgiwrap/mm64/classphoto?lw247">lw247</a>
207 Lisa Weinstein
208 </OL>
```

```
209 </BODY>
210 </HTML>
```

Event handlers in a form

```
1 <HTML>
2 <HEAD>
3 <TITLE>Event handlers in a form</TITLE>
4 <META http-equiv = "Content-Script-Type" content = "text/javascript">
5 </HEAD>
6 <BODY>
7 <H1>Event handlers in a form</H1>
8 <HR>
9
10 <FORM
11 NAME = "pizzaform"
12 METHOD = POST
13 ACTION = "/cgi-bin/cgiwrap/~mm64/onclick">
14
15 <h2>Order a slice of pizza</h2>
16
17 <P>Select only one kind of cheese:
18 <BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "regular" CHECKED>Regular
19 <BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "goat">Goat
20 <BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "velveeta">Velveeta
21 <BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "gouda"
22 onClick = "
23     if (this.checked == true && document.pizzaform.olives.checked == true) {
24         alert ('Olives do not go with gouda.');
```

- The gateway is a Perl program named `/home1/m/mm64/public_html/cgi-bin/onclick:`