

## Fall 2004 Handout 7

### URL and dates

The URL of the World Wide Web home page for this course has a tilde:

<http://i5.nyu.edu/~mm64/x52.9755>

<i>Lecture</i>	<i>Section 1 Thursday</i>
7	April 2
8	April 9
9	April 16
10	April 23
11	April 30

### Reference book

*JavaScript Unleashed, Second Edition* by Richard Wagner, et al. Sams.net Publishing, 1997; ISBN 1-57521-306-0. It's \$49.99 in aisle 9 downstairs at the NYU Main Bookstore (not the NYU Computer Bookstore), 18 Washington Place, (212) 998-4667, <http://www.bookc.nyu.edu/main>. Page numbers in the handouts refer to this book. The URL of the book is

[http://merchant.superlibrary.com:8000/  
catalog/mcp/PRODUCT/PAGE/15752/bud/1575213060.html](http://merchant.superlibrary.com:8000/catalog/mcp/PRODUCT/PAGE/15752/bud/1575213060.html)

A better choice would have been *JavaScript: The Definitive Guide, Second Edition* by David Flanagan. O'Reilly, 1997; ISBN 1-56592-234-4. The URL of the book is

<http://www.ora.com/catalog/jscript2/>

### Computer bookstores in Manhattan

Computer Bookworks (212) 385-1616  
78 Reade Street (near City Hall), between Broadway and Church Street  
[bookman3@mindspring.com](mailto:bookman3@mindspring.com)

Barnes and Noble (212) 807-0099  
105 Fifth Avenue at 18th Street  
<http://www.barnesandnoble.com/>

### Contact information

Mark Meretzky's home page: <http://i5.nyu.edu/~mm64/>

Home page for this course: <http://i5.nyu.edu/~mm64/x52.9555/>

Mark Meretzky's E-mail address: [mark.meretzky@nyu.edu](mailto:mark.meretzky@nyu.edu)

The system administrator's address is [comment@i5.nyu.edu](mailto:comment@i5.nyu.edu). For the NYU computer help desk, send email to [its.clientservices@nyu.edu](mailto:its.clientservices@nyu.edu), or call (212) 998-3333, or visit the ITS Client Services Center at <http://www.nyu.edu/its/helpdesk.html>. For problems with computer

accounts and passwords, send email to the accounts office at [its.accounts@nyu.edu](mailto:its.accounts@nyu.edu) or call (212) 998-3035, or visit <http://www.nyu.edu/its/accounts/stuacct.html>. For information about grades, incompletes, and NYU courses, including courses which I will teach next semester, call the School of Continuing and Professional Studies at (212) 998-7190. To contact me after the course is over, please send me email—don't phone.

### Homework, exams, grades

Your grade will depend on the homework you hand in to me on paper between the start of the course and 6:30 p.m. on the last class of the course. There will be no midterm or final. You get only one chance to hand in each problem. Hand in only the ones I assign in class, not all the problems in the Handouts.

Each assignment will be due one class after it is assigned. I can't predict when I will assign each assignment, since it depends on how fast the class goes. The class web page will list when each assignment is due. Incorrect or incomplete homework handed in on the last class will hurt your grade: it would be better to hand in nothing.

I will give you the answer to every assignment on the day it is due. You therefore get no credit for homework that I receive after that date or that is incorrect or incomplete. If you will be absent on the due date, hand it in early or mail it to me so that I receive it early:

Mark Meretzky  
NYU School of Continuing and Professional Studies  
Information Technologies Institute  
10 Astor Place, room 503 (between Broadway and Lafayette Street; closed after 6:00 p.m.)  
New York, NY 10003-6935

Put your real name, course number, and section number on the homework. Do not email me your homework. If you miss a class, send someone to tape it or take notes, and to drop and pick up your homework.

I will return each assignment to you one class after you give it to me, except for homework which you give me on or after the last class. I will give you back that homework only if you give me a self-addressed stamped envelope.

I will not give a grade to each individual homework, but I will correct every mistake you make. If there are no corrections, you did the homework perfectly. The only grade you will receive will be the one you get for the entire course. There is no way to predict this grade before the end of the course, since it partially depends on how well everyone else in the class does. You may also be penalized for gross absenteeism.

### Collaboration

You can collaborate with one or two other people on the same web page. In this case, you will all receive the same grade. In the real world you will program with other people, so I encourage you to do so now.

### The end of the course

I will not tell you your grade. I always give the grades to NYU the day after the last class of the course, or the day after NYU gives me the grading sheet, whichever comes last. I don't know what NYU will do with the grades or how long they will take to make them available to you. To hear your grade, call NYU TorchTone at (212) 998-7277 from a touch-tone phone. It will ask you for your nine-digit student ID number (usually your social security number) and your Personal Identification Number (or PIN, which is your birthdate in the format **MMDDYY**, e.g. **123174**). Or visit

<http://www.scps.nyu.edu/>

To request an Incomplete, fill out the Information Technologies Institute grading policy form and return it to me by the lecture day of class. I gave you this form on the first day of class. If you lost it, call (212) 998-7190 to get another, or print

**<http://i5.nyu.edu/~mm64/common/grading.html>**

Some students request an Incomplete just to extend their computer account.

To extend your i5.nyu.edu account if you have requested an Incomplete, fill out form ITS 775 at

**<http://www.nyu.edu/its/accounts/forms/request.extension.pdf>**

and bring it to me by the last day of class. After I sign it, take it to the address at the top of the form. Do not leave form ITS 775 with me.

To complete your incomplete, mail the URL of your page to me at the above address. Include your full name, social security number or NYU ID number, email address, the course and section number, and the year and semester when you took the course. Do not email me your late homework.

### Computer labs at NYU:

**<http://www.nyu.edu/its/labs/>**

(212) 998-3409	Room LC-8 Tisch Hall, two flights down
PC's	40 West 4th Street at Greene Street
printers:	<b>th_hp8150_1</b>
	<b>th_hp8150_2</b>
	<b>th_hp8150_3</b>

(212) 998-3457	14 Washington Place, one flight down
PC's	between Greene and Mercer Streets
printers:	<b>wp_hp8150_1</b>
	<b>wp_hp8150_2</b>

(212) 998-3421	Education Building, second floor
Macs	35 West 4th Street at Greene Street
printers:	<b>ed_hp8150_1</b>
	<b>ed_hp8150_2</b>
	<b>ed_hp8150_3</b>

(212) 998-3504	North Dorm, two flights down
PC's and Macs	75 Third Avenue (southeast corner of Third Avenue & 12th Street)
printers:	<b>nd_hp8100_2</b>

To get your plastic, magnetic NYU ID card, see

**<http://www.nyu.edu/nyucard/>**

### Your i5.nyu.edu account

Our computer is Sun 250 server running Solaris 8 (SunOS 5.8). Its Internet hostname is **i5.nyu.edu** and its IP version 4 address is **128.122.253.142**.

Your i5.nyu.edu login name is listed at

**<http://i5.nyu.edu/~mm64/common/students.html>**

It is the same as your NYU NetID used by your NYU DIAL or NYU Home account. It will be two or three *lowercase* letters (your initials) followed by one or more digits. In these Handouts, we'll assume your login name is **abc1234**.

Your i5.nyu.edu secret password is the same as your central NYU single sign-on password used by your NYU DIAL or NYU Home account. By default, it will be the first six digits of your NYU ID number (which is probably the same as your social security number). In these Handouts, we'll assume your password is **Bacall18?**.

Before using your login name and password for the first time, register at <http://start.nyu.edu/>. First time i5.nyu.edu users must leave the password field blank as they have not yet set their password. They will then be prompted to enter their social security number and birth date.

To change your password to a more colorful one, e.g., **bogart!** or **Bacall18?**, go back to <http://start.nyu.edu/>.

### The “secure shell” ssh

NYU will be your access provider if you don’t already have one. Go to one of the computer labs and get the ITS NYU-NET CD-ROM containing the software for DIAL/PPP and the installation documentation. See <http://www.nyu.edu/its/faq/>. Your NetID is the same as your login name.

To log into **i5.nyu.edu**, you have to run a program named **ssh**. It may be contained in another program named **PuTTY** or **putty.exe**. Mark’s web page

<http://i5.nyu.edu/~mm64/>

tells how to get it.

(1) On Windows, run **putty.exe**. A window named **PuTTY Configuration** will appear. Type **i5.nyu.edu** as the host name, and select the radio button for the protocol **SSH**. The port number should be 22. Then press **Open**. Dismiss any **PuTTY Security Alert** window that may appear.

```
login as: abc1234
password: Bacall18?
```

(2) On Macintosh OSX, launch the application **Terminal** to get Unix shell window. Pull down the **Font** menu and select a pleasant font. Then give the command

```
ssh abc1234@i5.nyu.edu
```

(3) On other Macs, launch the application **MacSSH Telnet**. A window named **Open Connection...** will appear. Type **i5.nyu.edu** as the host name, and check the checkbox for **Secure Shell**. Then press **Connect**.

If it doesn’t work, try again but say **128.122.253.142** instead of **i5.nyu.edu**.

### After logging in

When you are finished logging in, you will see the prompt. To verify that you are really logged in, run simple programs such as

```
1$ date
2$ cal 1 2004
```

*Press RETURN after each command line.  
Need space before each command line argument.*

### Log out

```
1$ exit
```

*or logout on other systems*

If the terminal window is still open, pull down the **File** menu and select **Exit** or **Quit** to close it.

### What can go wrong: a guide to the special keys

(1) If you accidentally type **control-z**, it will say **Stopped**. Type

```
1$ fg
```

*Bring the most recently stopped program back into the foreground.*

to start things up again. If it says **You have stopped jobs** when you try to log out, type **fg** to give your stopped job a chance to finish. Repeat if necessary.

(2) Press **backspace**, **delete**, or  to erase the last character typed. If your  or backspace key doesn’t work, type

2\$ `stty erase` ←

(with a space before the ←) and press **RETURN**. As a last resort, see if **control-h** will backspace.

(3) To kill a long program, type **control-c** on a terminal or a PC. You may have to type it several times.

(4) **Control-s** will freeze the screen; unfreeze it with **control-q**. Similarly, **Hold Screen** on a terminal or **Scroll Lock** on a PC will freeze the screen. Unfreeze it with another **Hold Screen** or **Scroll Lock**.

(5) Uppercase **control-U** will freeze the screen on a PC if you're running the programs **vi**, **view**, or **less**. You'll have to quit out of **telnet** and start again.

(6) Never press **Caps Lock** in this class: almost everything we type will be lowercase. Don't confuse

- (6a) the lowercase letter **l**, the uppercase letter **I**, and the digit **1**
- (6b) the lowercase letter **o**, the uppercase letter **O**, and the digit **0**
- (6c) the diagonal slash **/** and the backslash **\**
- (6d) the single quote **'**, the double quote **"**, and the back quote **`**
- (6e) the dash **-**, the underscore **\_**, and the tilde **~**
- (6f) the left parentheses **(**, the left curly brace **{**, and the left square bracket **[**
- (6g) the right parentheses **)**, the right curly brace **}**, and the right square bracket **]**
- (6h) the vertical bar (pipe symbol) **|**, the colon **:**, and the exclamation point **!**
- (6i) any uppercase letter and the corresponding lowercase letter.

### ▼ Homework 7.1: get three people's email addresses (not to be handed in)

During the intermission, get the name, phone number, and email address (i.e., login name) of three other people in the class whom you could call when you're absent to find out what we did and what the homework was. To get a balanced spectrum of opinion, pick one person who is close to you in age and gender, one person who is remote from you in age and gender, and one discretionary person.



### Set up your i5.nyu.edu account

Let's assume that your i5 prompt is **1\$**. To create a **public\_html** subdirectory of your home directory, containing an **index.html** file,

```
2$ /home1/m/mm64/public_html/x52.9749/makehome
```

The URL of your home page will then be

```
http://i5.nyu.edu/~abc1234
```

where **abc1234** is your i5 loginname.

### DocumentRoot, UserDir, and DirectoryIndex

Here are excerpts from the Hyperterxt Transport Protocol Dæmon configuration file **/usr/local/apache-1.3.14/conf/httpd.conf**. The documentation for this file is at

```
http://www.apache.org/docs/mod/directives.html
```

```
1 #If a URL specifies no filename, use this filename by default.
2 #If the directory has no file with this name, do an ls -l of the
3 #directory instead.
4
5 DirectoryIndex index.html
6
7 #In a URL, a user's loginname with a ~ in front of it stands
8 #for the following subdirectory of that user's home directory.
```

```

9
10 UserDir public_html
11
12 #The following is automatically prefixed to the directory specified
13 #in a URL, unless the specified directory begins with a user's
14 #loginname with a ~ in front of it.
15
16 DocumentRoot "/usr/local/apache-1.3.14/htdocs"
17
18 #In the URL of a gateway, the following short string stands
19 #for the much longer string.
20
21 ScriptAlias /cgi-bin/ "/usr/local/apache-1.3.14/cgi-bin/"

```

### How to get back to your public\_html subdirectory

Put all your `.html` files in the `public_html` subdirectory of your home directory, not in your home directory.

```

1$ cd                               Go to your home directory.
2$ cd public_html                   Go one level down.
3$ pwd                               See where you are.
   /home1/a/abc1234/public_html
4$ ls -l | more                       press space bar for more, or q to quit

```

### After you create a file with pico

```

1$ pico newfile.html

2$ ls -l newfile.html
-rw-----  1 abc1234      users      9 Apr  2 12:27 newfile.html

3$ chmod 644 newfile.html

4$ ls -l newfile.html
-rw-r--r--  1 abc1234      users      9 Apr  2 12:27 newfile.html

```

### The pair of <SCRIPT> tags

```

<SCRIPT>                               JavaScript is the default.
<SCRIPT LANGUAGE = "JavaScript">       Deprecated by the WWW Consortium.
<SCRIPT TYPE = "text/javascript">     How the Consortium wants it.

```

Better yet, you can set up a default `SCRIPT`ing language, e.g., `text/javascript`, `text/vbscript`, `text/tcl`, etc.:

```

<HEAD>
<META http-equiv = "Content-Script-Type" content = "text/javascript">
</HEAD>

<SCRIPT>

```

## Objects, methods, and properties

Every JavaScript statement ends with a semicolon.

<code>document.write("hello");</code>	<code>write</code> is a method of the object <code>document</code> .
John, eat cake!	
Subject, verb direct object!	
<code>document.write("good", "bye");</code>	comma between arguments
<code>document.lastModified</code>	<code>lastModified</code> is a property of the object <code>document</code> .

## The document object and its write method

```

1 <HTML>
2 <HEAD>
3 <TITLE>The document object and its write method</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>The document object and its write method</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 document.write ("Four score and seven years ago...");
10
11 </SCRIPT>
12 </BODY>
13 </HTML>
```

## More elaborate arguments for the write method

```

1 <HTML>
2 <HEAD>
3 <TITLE>The document object and its write method</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>The document object and its write method</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 document.write ("Four score and seven years ago ");
10 document.write ("our fathers brought forth on this continent a new nation, ");
11 document.write ("conceived in liberty, ");
12
13 //Can give more than one argument to document.write().
14 //Arguments can contain HTML tags.
15
16 document.write ("and dedicated to the proposition ",
17     "that all men are created equal.",
18     "<P>",
19     "Now we are engaged in a great civil war, ",
20     "testing whether that nation ",
21     "or any nation so conceived and so dedicated can long endure.");
22
23 </SCRIPT>
24 </BODY>
25 </HTML>
```

Concatenate strings: Flanagan, pp. 41, 57; Wagner, pp. 125–126

```
1 document.write("cup", "cake");
2 document.write("cup" + "cake");
3 document.write("cupcake");
4
5 var sentence = "Your browser is " + navigator.appName + ".";
6 document.write(sentence);
7
8 document.write ("hello ", 4, 5);
9 document.write ("hello " + 4 + 5);
10 document.write ("hello " + (4 + 5));
```

Display the properties of various objects.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Objects and their properties</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Objects and their properties</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 document.write (
10     "<P>This document last modified (GMT): ", document.lastModified,
11     "<BR>This document's URL: ", document.URL,
12     "<BR>I got here from: ", document.referrer,
13
14     "<P>Browser: ", navigator.appName,
15     "<BR>Running on: ", navigator.appVersion,
16     "<BR>", navigator.plugins.length, " plug-ins",
17
18     "<P>Current window width in pixels: ", window.innerWidth,
19     "<BR>Screen width in pixels: ", window.screen.width
20 );
21
22 </SCRIPT>
23 </BODY>
24 </HTML>
```

Change some of the properties of an object.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Change the properties of an object</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Change the properties of an object</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 document.bgColor = "red";
10 //Move mouse into window to see the status.
11 window.defaultStatus = "The background color of this document is red.";
12
13 </SCRIPT>
```



```
14 </BODY>
15 </HTML>
```

NEXT TIME: change a property and go to a new URL: "document has moved". Move it up to here from later handout.

**Create a new object and store it in a variable.**

```
1 <HTML>
2 <HEAD>
3 <TITLE>Create a new object</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Create a new object</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 var christmas = new Date(1997, 11, 25, 9, 0, 0);
10 document.write (
11     "<P>Christmas morning: ", christmas.toLocaleString(),
12     "<BR>The hour is: ", christmas.getHours()
13 );
14
15 var now = new Date();
16 document.write (
17     "<P>The current time is: ", now.toLocaleString(),
18     "<BR>The current hour is: ", now.getHours()
19 );
20
21 </SCRIPT>
22 </BODY>
23 </HTML>
```

**A variable can contain a number or string instead of an object.**

```
1 <HTML>
2 <HEAD>
3 <TITLE>A variable can contain a number or string.</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>A variable can contain a number or string.</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 //Create a variable and put a number into it.
10 var i = 10;
11 var j = 20;
12 var k = i + j;
13 document.write("The sum of ", i, " and ", j, " is ", k, ".");
14
15 //Change the contents of a variable.
16 i = 40;
17 document.write("<BR>The new value of i is ", i, ".");
18
19 //The computer concentrates first of the stuff to the right of the =.
20 i = i + 1;
21 document.write("<BR>The new value of i is ", i, ".");
```

```
22
23 var s = "Four score"
24 document.write("<BR>The value of s is ", s, ".");
25 </SCRIPT>
26 </BODY>
27 </HTML>
```

#### Pop up a dialog box: confirm, prompt, alert

```
1 <HTML>
2 <HEAD>
3 <TITLE>Dialog boxes</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Dialog boxes</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 //Don't need the leading window.
10
11 var true_or_false = window.confirm ("The confirm method.  Is my ballon OK?");
12 var n = window.prompt ("The prompt method.  What's your name?", "John Doe");
13 window.alert ("The alert method.  Is my triangle OK?");
14
15 document.write(n, " thinks it is ", true_or_false, " that my ballon is OK.");
16
17 </SCRIPT>
18 </BODY>
19 </HTML>
```

#### The if statement does one of several alternatives.

```
1 <HTML>
2 <HEAD>
3 <TITLE>The if statement</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>The if statement</H2>
7 <SCRIPT TYPE = "text/javascript">
8
9 var now = new Date();
10
11 if (now.getHours() < 12) {
12     document.write ("Good morning.");
13 }
14
15 //Two alternatives:
16 if (now.getHours() < 12) {
17     document.write ("<BR>Good morning.");
18 } else {
19     document.write ("<BR>Good evening.");
20 }
21
22 //Three alternatives:
23 if (now.getHours() < 12) {
```

```
24     document.write("<BR>Good morning.");
25 } else if (now.getHours() < 18) {
26     document.write("<BR>Good afternoon.");
27 } else {
28     document.write("<BR>Good evening.");
29 }
30
31 if (navigator.appName == "Netscape") {
32     document.write("<BR>I'm glad you're using Netscape.");
33 } else {
34     document.write("<BR>This document would look better with Netscape.");
35 }
36
37 if (parent == null) {
38     document.write("<BR>This document is not in a frame.");
39 } else {
40     document.write("<BR>This document is one frame in a larger window.");
41 }
42
43 </SCRIPT>
44 </BODY>
45 </HTML>
```

NEXT TIME: `if` with `parseInt` and `isNaN`.

An exercise in which you have to write an `if` statement.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Monday's Child is Fair of Face</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Monday's Child is Fair of Face</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 var now = new Date();
10
11 //Monday's child is fair of face,
12 //Tuesday's child is full of grace,
13 //Wednesday's child is full of woe,
14 //Thursday's child has far to go,
15 //Friday's child is loving and giving,
16 //Saturday's child has to work for its living--,
17 //But the child that's born on the Sabbath day
18 //Is fair and wise and good and gay.
19
20 //0 for Sunday, 1 for Monday, 2 for Tuesday, etc.
21 if (now.getDay() == 0) {
22     document.write("But the child that's born on the Sabbath day",
23         "<BR>Is fair and wise and good and gay.");
24 }
25
26 </SCRIPT>
27 </BODY>
28 </HTML>
```

Loops do the same thing over and over.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Looping up the New York State Thruway</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Looping up the New York State Thruway</H1>
7 <TABLE BORDER>
8 <TR>
9 <SCRIPT TYPE = "text/javascript">
10
11 for (var m = 108; m >= 68; m = m - 10) {
12     document.write(
13         "<TD>",
14         "<PRE>",
15         "Albany    ", m,
16         "<BR>Montreal ", m + 220,
17         "<BR>Buffalo  ", m + 280,
18         "</PRE>",
19         "</TD>"
20     );
21 }
22
23 </SCRIPT>
24 </TR>
25 </TABLE>
26 </BODY>
27 </HTML>
```

Use a loop to mass produce buttons.

```
1 <HTML>
2 <HEAD>
3 <TITLE>Mass produce radio buttons</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Mass produce buttons</H2>
7 <FORM>
8 <SCRIPT TYPE = "text/javascript">
9
10 for (var i = 0; i <= 9; i = i + 1) {
11     document.write(
12         "<BR>",
13         "<INPUT TYPE = BUTTON VALUE = ", i, ">"
14     );
15 }
16
17 </SCRIPT>
18 </FORM>
19 </BODY>
20 </HTML>
```

**Send a cookie to a client to remember his or her preferences.**

Do not run this example until we discuss it in class.

```

1 <HTML>
2 <HEAD>
3 <TITLE>Send a cookie to the client.</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Send a cookie to the client.</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 if (document.cookie == "") {
10     var bg = prompt ("Choose a background color, red or green:", "red");
11     document.cookie = "bg=" + bg;
12     document.bgColor = bg;
13 } else if (document.cookie == "bg=red") {
14     document.bgColor = "red";
15 } else if (document.cookie == "bg=green") {
16     document.bgColor = "green";
17 } else {
18     document.write("Unexpected cookie: ", document.cookie);
19 }
20
21 </SCRIPT>
22 </BODY>
23 </HTML>

```

**Package statements into a function.**

—Source code on the Web at

<http://i5.nyu.edu/~mm64/x52.9749/javascript/function.html>

```

1 <HTML>
2 <HEAD>
3 <TITLE>Package statements into a function.</TITLE>
4 </HEAD>
5 <BODY>
6 <H1>Package statements into a function.</H1>
7 <SCRIPT TYPE = "text/javascript">
8
9 function chorus()
10 {
11     document.write (
12         "<P>Secret Agent Man, Secret Agent Man",
13         "<BR>They've given you a number, and taken 'way your name.",
14         "<BR>Secret Agent Man, Secret Agent Man",
15         "<BR>They've given you a number, and taken 'way your name."
16     );
17 }
18
19 document.write (
20     "<P>There's a man who lives a life of danger",
21     "<BR>To everyone he meets he stays a stranger",
22     "<BR>With every move he makes, another chance he takes",
23     "<BR>Odds are he won't live to see tomorrow."

```

```
24 );
25
26 chorus();
27
28 document.write (
29     "<P>Beware of pretty faces that you find",
30     "<BR>A pretty face can hide an evil mind",
31     "<BR>Be careful what you say, or you'll give yourself away",
32     "<BR>Odds are he won't live to see tomorrow."
33 );
34
35 chorus();
36
37 document.write (
38     "<P>Swinging on the Riviera one day",
39     "<BR>Lying in a Bombay alley the next day",
40     "<BR>Don't let the wrong words slip while kissing persuasive lips",
41     "<BR>Odds are he won't live to see tomorrow."
42 );
43
44 chorus();
45
46 </SCRIPT>
47 </BODY>
48 </HTML>
```

A form that runs a gateway.

http://i5.nyu.edu/~mm64/x52.9749/javascript/pizzacgi.html

```

<HTML>
<HEAD>
<TITLE>A form that runs a gateway</TITLE>
</HEAD>
<BODY>
<H1>A form that runs a gateway</H1>
<SCRIPT TYPE = "text/javascript">

</SCRIPT>

<HR>
<FORM METHOD = POST ACTION = "/cgi-bin/cgiwrap/~mm64/pizzacgi">

<P>
How many slices @ $1.75 each?
<INPUT TYPE = TEXT SIZE = 3 NAME = "n">

<P>Select only one kind of cheese:
<BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "regular" CHECKED>Regular
<BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "goat">Goat
<BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "velveeta">Velveeta
<BR><INPUT TYPE = RADIO NAME = "cheese" VALUE = "gouda">Gouda

<P>Select as many toppings as you want. 50&cent; each:
<BR><INPUT TYPE = CHECKBOX VALUE = 1 NAME = "pepperoni">Pepperoni
<BR><INPUT TYPE = CHECKBOX VALUE = 1 NAME = "mushroom">Mushroom
<BR><INPUT TYPE = CHECKBOX VALUE = 1 NAME = "sausage">Sausage
<BR><INPUT TYPE = CHECKBOX VALUE = 1 NAME = "olives">Olives

<P>
<INPUT TYPE = SUBMIT VALUE = "Order the slice(s).">
<INPUT TYPE = RESET VALUE = "Start again.">
</FORM>
<HR>
</BODY>
</HTML>

```

The gateway is a Perl program:

### Event handlers

This form runs the JavaScript onClick and onChange event handlers before it runs its gateway.

The gateway is a simpler Perl program:

### Exercise

Make the above form display an **alert** message if more than 20 slices have been ordered. Simply put an **if** statement in the **compute** function. Better yet, instead of displaying an alert message, you can travel to a new URL by changing the value of the property **window.location**.

□