Introduction to Web-scripting languages

Markup languages, Javascript and CGI

Philippe Martin
(philippe.martin@gu.edu.au)

Laboratory for Knowledge Visualisation and Ordering (KVO)
(http://www.int.gu.edu.au/kvo/)

Overview

- Markup languages for structuring documents or data
- Scripting languages for creating dynamic documents
- Client-side scripts with Javascript
- Server-side scripts via the Common Gateway Interface

File Edit View Go Communicator Help

Introduction to Web–scripting languages

This course is one of the Emerging Technologies (INT3004) courses
and occurs in semester 2, weeks 10, 11, 12 and 13.
(Lecture: wednesday 8–10am BUG 3.36, tutorial: friday 8–9am ENG 1.34).
Evaluation: project of 30 hours (to be done in 3 weeks),
1/3 of the notation for the Emerging Technologies course.

Wed. 22/09	General introduction Introduction to Javascript features and ressources Explanation of a Javascript program handling form elements Explanation of the Javascript project (to be completed by Wed. 20/10)
Fri. 24/09	Javascript exercises

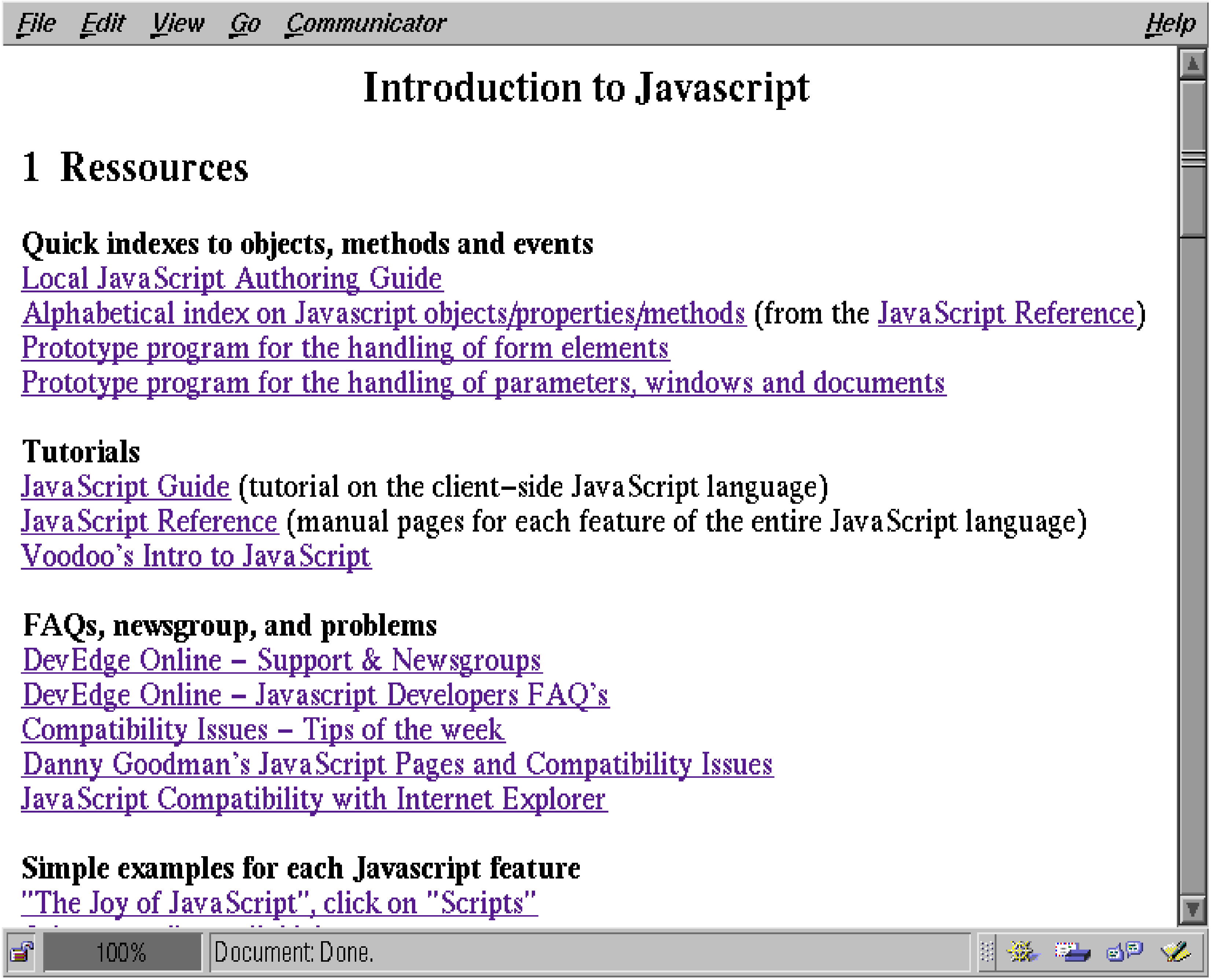
Wed. 06/10	The Get and Post methods to encode parameters Explanation of a Javascript program to encode/decode Get parameters and handles windows Security issues
Fri. 08/10	Exercises on window handling and parameter sending in Javascript

Wed. 13/10	Using the Post method for sending parameters to a CGI server Explanation of a C program to encode/decode Get and Post parameters
Fri. 15/10	Exercise on CGI servers in C and shell

Wed. 20/10	Introduction to some successors of HTML: XML, VRML, CSS and XSL
Fri. 22/10	Exercises with CSS (+ evaluation of the Javascript project ?)

URL of this document: http://meganesia.int.gu.edu.au/~phmartin/int3004/

100% Document: Done.



File Edit View Go Communicator Help

Introduction to Javascript

1 Ressources

Quick indexes to objects, methods and events
Local JavaScript Authoring Guide
Alphabetical index on Javascript objects/properties/methods (from the JavaScript Reference)
Prototype program for the handling of form elements
Prototype program for the handling of parameters, windows and documents

Tutorials
JavaScript Guide (tutorial on the client–side JavaScript language)
JavaScript Reference (manual pages for each feature of the entire JavaScript language)
Voodoo's Intro to JavaScript

FAQs, newsgroup, and problems
DevEdge Online – Support & Newsgroups
DevEdge Online – Javascript Developers FAQ's
Compatibility Issues – Tips of the week
Danny Goodman's JavaScript Pages and Compatibility Issues
JavaScript Compatibility with Internet Explorer

Simple examples for each Javascript feature
"The Joy of JavaScript", click on "Scripts"

100% Document: Done.

JavaScript	**Java**
Code integrated with, and embedded in HTML	Applets distinct from HTML (accessed from HTML pages)
Interpreted (not compiled) by client	Compiled bytecodes downloaded from server, executed on client
Dynamic binding. Object references checked at runtime	Static binding. Object references must exist at compile-time
Object-based. No distinction between types of objects. Properties and methods can be added to any object dynamically	Object-oriented. Inheritance through the class hierarchy. Classes and instances cannot have properties or methods added dynamically
Variable data types not declared	Variable data types must be declared
Cannot automatically write to hard disk	Cannot automatically write to hard disk

Tips

To force the reload of an HTML document after its modification, use <shift>Reload, not just Reload. If that still does not work, clear the caches with the "Preferences - Advanced" menu or/and exit from the browser.

If you made a bug in a Javascript program (e.g. an infinite loop) and your browser does not let you remove the caches via the menu, you'll have to do it by hand (on Unix, type: rm -r /.netscape/cache/*).

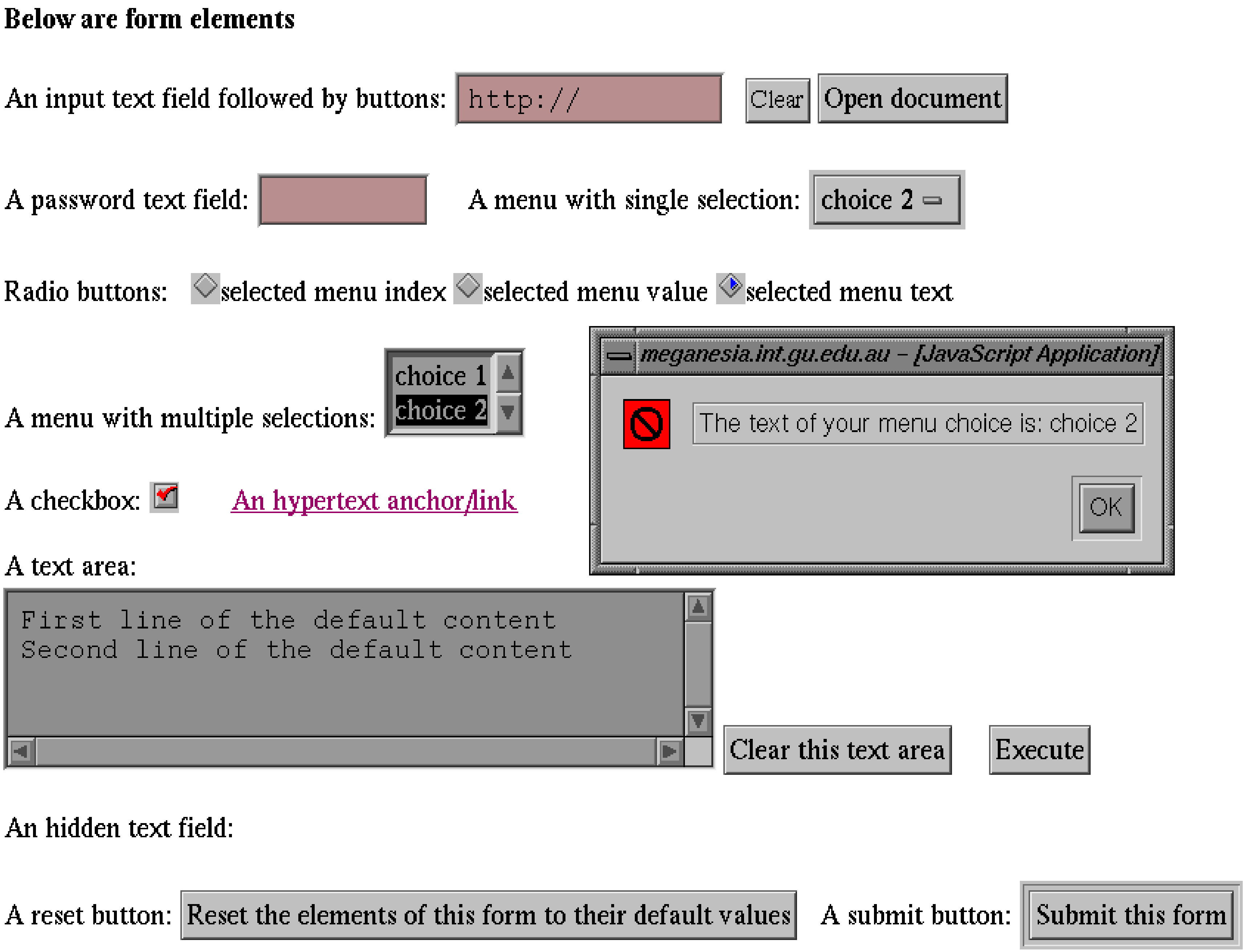
When creating hyperlinks, use relative URLs. To refer to the file index.html in a directoty, use the name of this file, do not just put the path of the directory (some Web servers do not understand this abbreviation).

```
<html><head><title>Basic example of HTML and Javascript file</title>
<!-- If Javascript is enabled the content of this document is:
Hello Netscape's user! This sentence is not generated via Javascript.
 -->

<script language="javascript">
<!--
var WelcomingMessage = "<p>Hello ";

function /*void*/ WelcomeTheUser ()
{ document.write(WelcomingMessage + navigator.appName + "'s user!\n");
}
//End of script -->
</script>
</head>

<body bgcolor="white">

<script language="javascript">
<!--
WelcomeTheUser ();
//end of script -->
</script>

This sentence is not generated via Javascript.
</body>
</html>
```

Javascript example

```
This paragraph about the current environment is generated via Javascript.
  userAgent: Mozilla/4.05C-SGI [en] (X11; I; IRIX64 6.5 IP27);
  appCodeName: Mozilla;
  appVersion: 4.05C-SGI [en] (X11; I; IRIX64 6.5 IP27);
  appName: Netscape;
  language: en;
  platform: IRIX6.2;
  plugins: [object PluginArray];
  mimeTypes: [object MimeTypeArray];
  screen: [object Screen];  history: [object History];
  parent frame: [object Window];
  document title: Example of HTML and Javascript file; URL:
http://meganesia.int.gu.edu.au/~phmartin/int3004/javascript/formPrototype.html
```

This paragraph about the object MyCar is also generated via Javascript.
make: Ford; color: white; year: 1990; model: Mustang; 0: ?; [object Object]: ?; wheels: 4;


Below are form elements
An input text field followed by buttons:
http://
Clear
Open document
A password text field:
A menu with single selection:
choice 2
Radio buttons:
selected menu index
selected menu value
selected menu text
meganesia.int.gu.edu.au – [JavaScript Application]
The text of your menu choice is: choice 2
OK
A menu with multiple selections:
choice 1
choice 2
A checkbox:
An hypertext anchor/link
A text area:
First line of the default content
Second line of the default content
Clear this text area
Execute
An hidden text field:
A reset button:
Reset the elements of this form to their default values
A submit button:
Submit this form

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN">
<html>
<head><title>Example of HTML and Javascript file</title>

<meta name="Author" content="Philippe Martin">
<meta name="keywords"  content="Javascript example">
<meta name="description" content="Example of HTML and Javascript file">

<link rel="Author" rev="Made" href="mailto:philippe.martin@gu.edu.au">
<link rel="Previous" href="../index.html">

<script language="javascript">
<!-- //note: Netscape has problems if this comment is on the previous line
```

```
//*********** 1) GLOBAL VARIABLES ************

var Boolean_globalVariable= true;
var String_globalVariable =
      "Line 1\nLine 2 \
       Line 3. Two quotes: ''. An escaped backquote: \" \
       Line 4. Special characters: \b\f\n\r\t";
var Number_globalVariable = 3.14;
var AnotherGlobalVariable = null;
var ChildWindow = null;

var MyCar = new Car("Ford");
MyCar.year=1990; MyCar["model"]="Mustang";MyCar[0]="?"; MyCar[MyCar]="?";


//*********** 2) APPLICATION-DEPENDENT FUNCTIONS ************

function /*Car*/ Car (aMake)  //Javascript objects are associative arrays
{ this.make= aMake; this.color="white";
  //to declare the method toString():  this.toString= MyObject_toString
  return this;
}
Car.prototype.wheels=4;

function /*String*/ MyObject_toString ()
{ return slotsOfObject (this); }
```

```
//************ 3) COOKIE HANDLING FUNCTIONS ************
//cf. developer.netscape.com/docs/manuals/communicator/jsguide4/cookies.htm

function /*void*/ registerCookie (name, value)
{ var today= new Date();  var expirationDate= new Date();
  expirationDate.setTime(today.getTime()+1000*60*60*24*365);//in a year
  setCookie(name,value,expirationDate);
}

function /*void*/ setCookie (name, value, /*optional*/expirationDate)
{ if (expirationDate)
     document.cookie= name + "=" + escape(value) +
                      "; expires=" + expirationDate.toGMTString();
  else document.cookie= name + "=" + escape(value);
} //escape() encodes in hexadecimal characters such as ';', ',' and spaces

function /*String*/ getCookie (name)
{ if (document.cookie.length <= 0) return ""; //no cookie
  var cookieBeginning= document.cookie.indexOf(name+"=");
  if (cookieBeginning == -1) return ""; //no such cookie
  var valueBeginning = cookieBeginning + name.length + 1;
  var valueEnd= document.cookie.indexOf(";",valueBeginning);
  if (valueEnd == -1) valueEnd= document.cookie.length;
  return unescape(document.cookie.substring(valueBeginning,valueEnd));
}
```

```
//*********** 4) GENERAL BASIC FUNCTIONS ***********

function /*String*/ slotsOfObject (anObject)
{ var result="";
  for (var slot in anObject)
    result += slot + ": " + anObject[slot] + ";\n  ";
  return result;
}

function /*String*/ selectedTextInSelectArray (selArray)
{ return selArray[selArray.selectedIndex].text;
}
function /*String*/ selectedValueInSelectArray (selArray)
{ return selArray[selArray.selectedIndex].value;
}

function /*String*/ selectedTextInDoc(doc)
{ if (navigator.appName=="Netscape") return doc.getSelection();
  return doc.selection; //for MSIE4
}
```

```
function /*Natural*/ asciiCodeOf(c) { return c.charCodeAt(0); }
function /*bool*/ isSpace(c)
{ return ((c==' ') || (c=='\n') || (c=='\r') || (c=='\t') || (c=='\f')); }

function /*bool*/ isIn (str1,str2) { return (str2.indexOf(str1) >= 0); }
function /*bool*/ isLower (str) { return (str==str.toLowerCase()); }
function /*bool*/ isUpper (str) { return (str==str.toUpperCase()); }
//functions parseInt(str), parseFloat(str), isNaN(str)

function /*Number*/ regExprIn (/*String*/regularExpression, str)//->index/-1
{ var reg = new RegExp(regularExpression);  return reg.search(str);
}//developer.netscape.com/docs/manuals/communicator/jsref/corea3.htm#1158210
//or: developer.netscape.com/docs/manuals/communicator/jsguide4/expr.htm

function /*bool*/ isInStrings (str,strArray)
{ for (i in strArray) { if (strArray[i]==str) return true; }
  return false;
}
```

```
function /*bool*/ tr () //trace function with a variable number of arguments
{ var result="";
  var nbArg = tr.arguments.length;
  for (var i=0; i<nbArg; i++)  result += tr.arguments[i] + ";\n";
  return confirm(result);
}
function /*bool*/ error (str) { return confirm(str+"."); } //error message
function /*void*/ reload() {  history.go(0); }

function /*void*/ goThereInTime(url,milliseconds)
{ setTimeout("self.location.href="+url, milliseconds); }

//End of script -->
</script>
</head>
```

```
<body bgcolor="white">
<!--  onLoad="alert('this document has now been loaded');"
      onUnLoad="alert('this document is unloading');"
  -->
<h2 align="center">Javascript example</h2>

<NOSCRIPT>
<p><b>Note: this document uses Javascript which is not enabled in
your browser.<br>
You need to get Netscape Navigator 2.0 (or later) or Internet Explorer 3.0
(or later). You should also enable JavaScript by on the Advanced page of
the Preferences dialog box.</b></p></br>
</NOSCRIPT>
```

```
<script language="javascript">
<!--
document.write("<pre>This paragraph about the current environment is " +
               "generated via Javascript.\n  ");
document.write(slotsOfObject(navigator));
document.write("screen: "+ screen+ ";  history: "+window.history+";\n  ");
document.write("parent frame: "+window.parent+";\n  ");
document.write("document title: "+document.title+"; URL:\n");
document.write(window.location+"\n</pre>\n");

document.write("<p>This paragraph about the object MyCar is also " +
               "generated via Javascript.<br>\n");
document.write(slotsOfObject(MyCar));
//end of script -->
</script>
```

```
<FORM name="myForm" action="http://somewhere/someProgram.cgi" method="Post"
      target="resultWindow"
      onSubmit="return confirm('Do you really want to send the parameters '
                                    + 'in this form to ' + this.action);">

<p><br><strong>Below are form elements</strong>

<p>An input text field followed by buttons:
  <input type="TEXT" name="aTextField" size=15 value="http://"
         onSelect="alert('the text is: ');"
         onChange="alert('a change has been made');">
    <!-- onFocus ="alert('the focus is on the text field);"
         onBlur  ="alert('the focus is out of the text field);" -->

   <small>
  <input type="BUTTON" value="Clear"
         onClick="this.form.aTextField.value='';"></small>

  <input type="BUTTON" value="Open document"
         onClick="ChildWindow= window.open(this.form.aTextField.value,'');">

<p>A password text field:
  <input type="PASSWORD" name="aPasswordField" size=9 value="">
     
```

```
A menu with single selection:
<SELECT name="aMenu"
        onChange="alert('a change has been made');">
   <!-- onFocus ="alert('the focus is on the menu');"
        onBlur  ="alert('the focus is out of the menu');" -->
   <option value="a">          choice 1
   <option value="b" selected>choice 2
   <option value="c">          choice 3
</SELECT>

<p>Radio buttons:  
  <input type="RADIO" name="aRadioChoice" checked
        onClick="var sel= this.form.aMenu.selectedIndex;
                 alert('The index of your menu choice is: ' + sel);
                ">selected menu index
  <input type="RADIO" name="aRadioChoice"
        onClick="var sel=selectedValueInSelectArray(this.form.aMenu);
                 alert('The value of your menu choice is: ' + sel);
                ">selected menu value
  <input type="RADIO" name="aRadioChoice"
        onClick="var sel=selectedTextInSelectArray(this.form.aMenu);
                 alert('The text of your menu choice is: ' + sel);
                ">selected menu text
```

```
<p>A menu with multiple selections:
  <SELECT name="aMultiSelectionMenu" multiple size=2
          onChange="alert('a change has been made');">
     <option value="a">          choice 1
     <option value="b" selected>choice 2
     <option value="c">          choice 3
  </SELECT>    


<p>A checkbox:
  <input type="CHECKBOX" name="aCheckbox" checked
         onClick="alert('This checkbox is ' +
                         (this.checked ? 'checked' : 'unchecked') );">
       

  <a href="aURL"
     onClick="return confirm('Do you really want to go to'+this.href);"
     onMouseOver="window.status='You are over a link'; return true"
  >An hypertext anchor/link</a>
```

```
<p>A text area: <br>
  <TEXTAREA NAME="aTextArea" ROWS=4 COLS=40
         onSelect="alert('This event generally does not occur');">
First line of the default content
Second line of the default content
  </TEXTAREA>
  <input type="BUTTON" value="Clear this text area"
         onClick="this.form.aTextArea.value=''">    
  <input type="BUTTON" value="Execute"
         onClick="eval(this.form.aTextArea.value)">

<p>An hidden text field:
  <input type="HIDDEN" name="aHiddenField" value="aParameter">
<p>A reset button:
  <input type="RESET"
         value="Reset the elements of this form to their default values">
  A submit button:
   <input type="SUBMIT" value="Submit this form">
</FORM>
</body></html>
```

Project: refinement of a knowledge retrieval tool

Handling a simple knowledge representation language

Example: "The cat Tom is white and is on a blue mat"

```
[the cat Tom, color: the white,
              on: (a mat, color: the blue)]
```

```
[Cat, subtype of: Animal] //a cat is an animal
```

```
spec [an animal, color: the white]
```

Knowledge Retrieval Tool

KB http://meganesia.int.gu.edu.au/~phmartin/int3004/javascript/pr Clear Open

Query Direct query mode | Display the queries before the results

--- Query examples --- Clear Submit

Ressources

- The existing FCGquery
- The grammar of the FCG language
- A KB with various kinds of knowledge
- Examples of queries

Features to add to FCGquery

- Parsing of the whole grammar
- Specialisation of types and FCGs
- Path retrieval
- Implementation of aliases