

Topic 1: HTML and JavaScript

1. HTML and its extensions
2. JavaScript programming
3. Elements of JavaScript

Exporting from MS Word to HTML

- To insert a hyperlink, select text, right click, and fill in form with URL
- To save as HTML, use *SaveAs*, select *Web Page*
- HTML file may be uploaded to a web server
- Note that while pictures may be embedded in Word files, they may not in HTML files

HTML tags

- Delimited by <, >
- Tags surround text to format it
- *Examples:*
 - **This <i> is </i> HTML**
This *is* HTML
 - **<title> 63.120 </title>**
(Designates text as a page title that appears in title bar in browser)
 - **<p> ... </p>** designates a paragraph
 - **<hr>** (generates horizontal rule)
 - **Search**
(hyperlink [Search](http://www.cnn.com))

Hyperlinks

- When user clicks on *link text* of a hyperlink (e.g., blue underlined text)
 - Browser places current HTML file on a *stack* maintained by browser
 - Browser downloads Internet file referenced by hyperlink
 - Browser displays referenced file, whether HTML, PDF, or other
 - Clicking “Back” button returns browser to HTML file at top of stack

HTML features

- Text links (hypertext)
- Graphic links (e.g., buttons)
- *Hot spots*: areas used as links
- *Forms*: Enable collection of data from user uploaded to server and processes by CGI script
- *Tables*: grids of cells for layout
- *Frames*: areas that scroll independently of the rest of page

Extensions to HTML

- Dynamic HTML: enables animation, mousover effects
- XML (Extensible Markup Language): supports structuring of data with tags, e.g., for fields of a database
- Scripting languages: JavaScript, VBScript, PerlScript
- Client-side scripts are embedded in HTML code, server-side scripts generate custom HTML files

2. Programming in JavaScript

- *Motivation:* Working with IT means thinking abstractly about data and operations
- Design, coding, testing of solutions is part of learning problem solving
- *JavaScript:* A procedural language, interpreted on browsers, with all the power of languages like C, C++, Java
- Compiled (offline translation) vs. interpreted (simultaneous translation) code
- Use tags:
`<script language = "Javascript"> ... </script>`
- The JavaScript text between these tags will execute when browser displays HTML file

Computer Science I Using Java David Keil 1/08

Event-driven programming

- *Browsers* and most other apps are *interactive*: alternate input and output
- *Command-line environments*: URL line in browser, Google prompt, DOS or UNIX prompt
- *GUI events*: Windows/Mac/Xwindows
- *Features*: Icons, menus, dialog boxes, windows, buttons, scrollers, check boxes
- *Common feature*: User generates *events*, e.g., clicks, drags, keystrokes, timeouts
- One form of interaction in browser: hyperlinks
- Another is embedding of event-based JavaScript programs in HTML files

Computer Science I Using Java David Keil 1/08

Simple JavaScript example

```
<html>  <!-- hello.htm -->
  <head><title>DK-Hello</title></head>
  <body>63.120 says Hello!
    <script language="JavaScript">
      alert("hello");
    </script>
  </body>
</html>
```

- Displays “hello” in an alert box (a kind of dialog)
- **alert** is a JavaScript function (a kind of procedure)
- JavaScript may be used after **script** tag

Computer Science I Using Java David Keil 1/08

Simple button

```
<html> <!--button.htm-->
<head><title>63120 Hello</title></head>
<body>
  <input type=button  value = "Hello"
    onClick = 'alert("Hi")'>
</body></html>
```

- This code displays “Hi” when “Hello” button is pressed
- **<input>** tag defines an input button object
- *Event handler*: code that specifies application’s response to a particular event, such as user click on a button

Computer Science I Using Java David Keil 1/08

Counting button clicks

```
<html> <head><title>yes-no counter</title></head>
<body>
  /* count-yes.htm Displays Yes, No buttons for user to click,
  counts # clicks on each. Four 'event-handlers' specify the
  program's response to four different input events: Yes, No,
  Stats, Reset. When user presses 'Yes' button, variable
  'num_yes' is incremented. */
  <script language="JavaScript">
    var num_yes=0, num_no=0; // Variables
  </script>
  <td><input type=button value = "Yes"
            onClick = 'num_yes = num_yes + 1'></td>
  <td><input type=button value = "No"
            onClick = 'num_no = num_no + 1'></td>
  <td><input type=button value = "Stats"
            onClick = 'alert("Yes: "+num_yes + " No: "+num_no)'></td>
  <td><input type=button value = "Reset"
            onClick = 'num_yes = num_no = 0'></td>
</body>
</html>
```

Computer Science I Using Java David Keil 1/08₁

Text input/output

```
<head><title>Input echo</title></head>
<body>
  <form name="Input"><table>
    <!-- Display prompt and get input:-->
    <td>Enter your user name:
    <!-- Generate input-box:-->
      <input type=text name=user_name
            value="" size = 15> </td>
    <!-- Wait for button-press, display message:-->
    <td><input type=button value="Done"
            onClick = 'alert("Hello " + user_name.value)'>
      </td>
    <!-- Assigning a value to onClick defines
    JavaScript response to clicking button-->
  </table></form>
</body>
```

Computer Science I Using Java David Keil 1/08₂

3. Elements of JavaScript

JavaScript variables

- Variables have *name*, *type*, *value*
- Must declare a variable to use it: **var a;**
- Valid *types* (interpretation of bits):
- Number: **var a=2, b-3;**
- String: **var name = "Bill";**
- Truth value: **var greater = (a > b);**
- *Assignment* can change variable's value:
a = b * 4;

Computer Science I Using Java David Keil 1/08₃

Expressions

- *Expressions* have *values*
- Expressions are *literals*, *variables*, or *function calls*, or may be formed with *operators* (+, -, *, /, %)
- *Boolean* expressions have truth values and may be built with *relational operators* (==, !=, <, >, <=, >=) or *logical operators* (!, &&, ||)

Computer Science I Using Java David Keil 1/08₄

Statements

- *Statements* are *executable* (or are variable declarations)
- Statements include
 - *assignment* **a = 4;**
 - *if, if ... else*
 - *while,*
 - compound statements { in braces }
- *Simple* statements, e.g., assignment, terminate with “;”

Computer Science I Using Java David Keil 1/08₅

Sources

- J. Parsons, D. Oja. *Computer Concepts*, 9th ed. Thomson, 2007.
- L. Snyder. *Fluency with Information Technology*. Addison Wesley, 2006.

Computer Science I Using Java David Keil 1/08

16