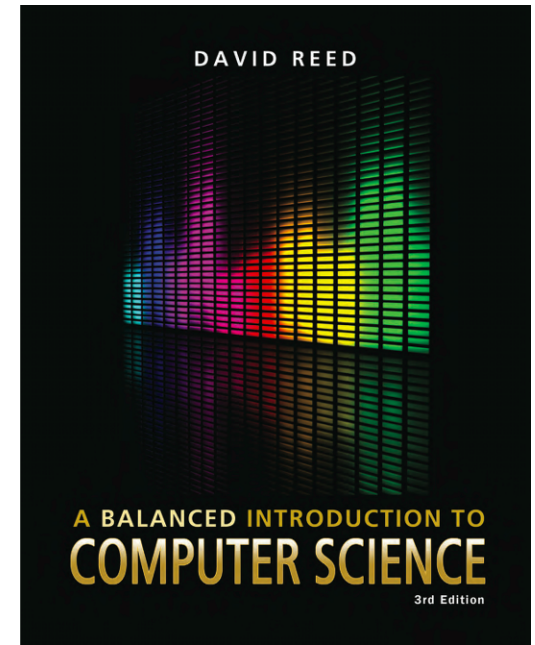


A Balanced Introduction to Computer Science, 3/E

David Reed, Creighton University

©2011 Pearson Prentice Hall
ISBN 978-0-13-216675-1



Chapter 4

JavaScript and Dynamic Web Pages

Static vs. Dynamic Pages



recall: a Web page uses HTML tags to identify page content and formatting information

HTML can produce only *static pages*

- static pages look the same and behave in the same manner each time they are loaded into a browser

in 1995, researchers at Netscape developed JavaScript, a language for creating *dynamic pages*

- Web pages with JavaScript can change their appearance:
 - ▣ over time (e.g., a different image each time that a page is loaded), or
 - ▣ in response to a user's actions (e.g., typing, mouse clicks, and other input methods)

Programming Languages



JavaScript is a *programming language*

- a *programming language* is a language for specifying instructions that a computer can execute
- each *statement* in a programming language specifies a particular action that the computer is to carry out
(e.g., changing an image or opening a window when a button is clicked)

some programming languages are general-purpose

- popular languages include C++, Java, J#

JavaScript was defined for a specific purpose: *adding dynamic content to Web pages*

- can associate JavaScript statements with certain HTML elements so that they react to actions taken by the user (e.g., a button click)

Event Handlers



JavaScript code is associated with HTML element using *event handler* attributes

- the ONMOUSEOVER attribute specifies the statement(s) that are to be executed when the mouse is moved over the element
- the ONMOUSEOUT attribute specifies the statement(s) that are to be executed when the mouse is moved off the element
- for example, can have an image that reacts to mouse movements:

```

```

the simplest type of action is to change the value of one of the other attributes of the element

- this is accomplished via a JavaScript *assignment statement*

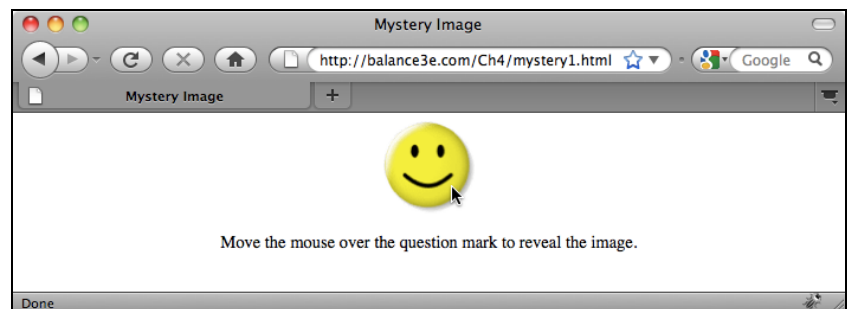
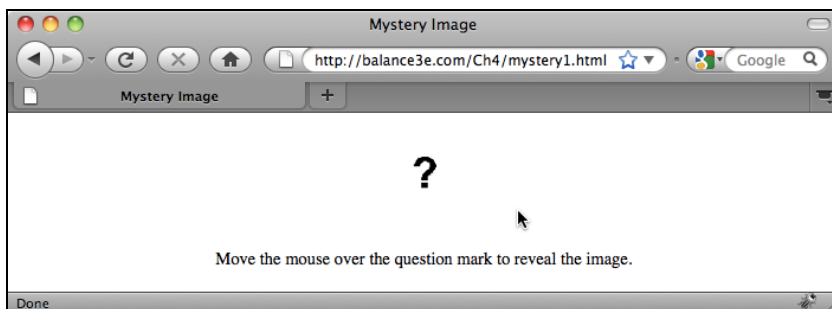
```
this.ATTRIBUTE_NAME = NEW_ATTRIBUTE_VALUE;
```

Mystery Image Page



```
1. <!doctype html>
2. <!-- mystery1.html                      Dave Reed -->
3. <!-- This page changes an image source on mouseover. -->
4. <!-- ===== -->
5.
6. <html>
7.   <head>
8.     <title> Mystery Image </title>
9.   </head>
10.
11.   <body>
12.     <div style="text-align:center">
13.       
16.     <p>
17.       Move the mouse over the question mark to reveal the image.
18.     </p>
19.   </div>
20. </body>
21. </html>
```

- initially, the image displays a '?'
- when mouse moves over, SRC attribute is assigned to happy face
- when mouse leaves, SRC attribute is assigned back to '?'



HTML/JavaScript Strings



a string is a sequence of characters enclosed in quotes

- strings appear in HTML attribute assignments (e.g., file name for SRC attribute)
- strings also appear in JavaScript assignments (e.g., attribute assignment within ONMOUSEOVER)

```

```

here, JavaScript assignments (gray text) are nested inside the HTML strings assigned to the event handlers

technically, strings can be represented using either single or double quotes

- to avoid confusion, we will always use double quotes for HTML strings; single quotes for JavaScript strings



Other Dynamic Attributes

can similarly change other element attributes using JavaScript assignments

- HEIGHT attribute of an IMG sets the height of the image (in number of pixels)
- WIDTH attribute of an IMG sets the width of the image (in number of pixels)

```

```

here, moving the mouse over the image expands it from 85x85 pixels to 200x200 pixels

similarly, moving the mouse away reduces it back to 85x85 pixels

an event handler attribute can be assigned a series of attribute assignments (separated by semicolons)

- multiple assignments are executed in the order they appear

onclick Event Handler



another common event handler is the ONCLICK attribute

- specifies JavaScript statement(s) to be executed when the mouse is clicked on the element

```

```

```
1. <!doctype html>
2. <!-- mystery2.html                      Dave Reed -->
3. <!-- This page changes an image source on a click. -->
4. <!-- ===== -->
5.
6. <html>
7.   <head>
8.     <title> Mystery Image </title>
9.   </head>
10.
11.   <body>
12.     <div style="text-align:center">
13.       
15.     <p>
16.       Click on the question mark to reveal the image.
17.     </p>
18.   </div>
19. </body>
20. </html>
```


Input buttons



the `onClick` attribute is often associated with buttons

- a button is an HTML element that appears as a labeled rectangle/oval

Click for free money!

a button is a specific type of input element

```
<input type="button" value="BUTTON_LABEL"  
onClick="CODE_TO_BE_EXECUTED_WHEN_MOUSE_CLICKS_ON_BUTTON">
```

typically, buttons are used to initiate actions on other elements

- e.g., click on a button to change the SRC or HEIGHT/WIDTH of an IMG
- need to be able to identify the element (as opposed to this.)

```
document.getElementById('ELEMENT_ID')
```

Button Version



```
1. <!doctype html>
2. <!-- mystery3.html                                Dave Reed -->
3. <!-- This page changes an image source on button clicks. -->
4. <!-- ===== -->
5.
6. <html>
7. <head>
8.   <title> Mystery Image </title>
9. </head>
10.
11. <body>
12.   <div style="text-align:center">
13.     
14.   <p>
15.     <input type="button" value="Show Image"
16.       onclick="document.getElementById('mysteryImg').src=
17.         'http://balance3e.com/Images/happy.gif';">
18.     <input type="button" value="Hide Image"
19.       onclick="document.getElementById('mysteryImg').src=
20.         'http://balance3e.com/Images/mystery.gif';">
21.   </p>
22. </div>
23. </body>
24. </html>
```



each button
click changes
the SRC
attribute of the
element with ID
mysteryImg

note: the IDs must match exactly,
including the same capitalization

Dynamic Text

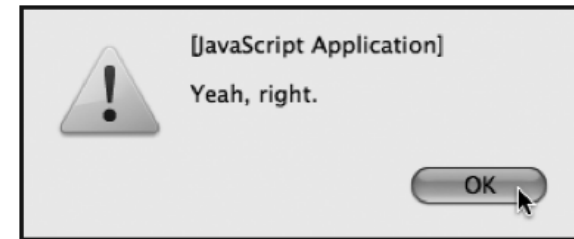


to display text within a page, there are 2 main options:

1. `alert` statement: will display a text message in a separate alert window

```
<input type="button" value="Click for free money!"  
onClick="alert('Yeah, right.');">
```

- ❑ alert windows are useful when you want to display a short (1-line) message
- ❑ can be annoying to the user since the pop-up window must be closed by clicking the OK button



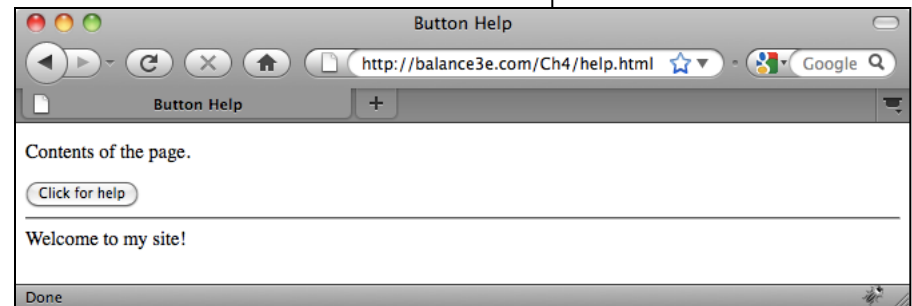
2. `INNERHTML` attribute: text elements (`DIV`, `P`, `SPAN`) have an attribute that specifies their text content
 - ❑ you can change the text displayed in one of these elements by assigning a new string to its `INNERHTML` attribute
 - ❑ advantages:
 - ✓ the text is integrated into the page (not in a separate window)
 - ✓ it can accommodate long messages
 - ✓ it can incorporate HTML formatting (e.g., paragraphs using `P`, bold using `B`, ...)

Help Page

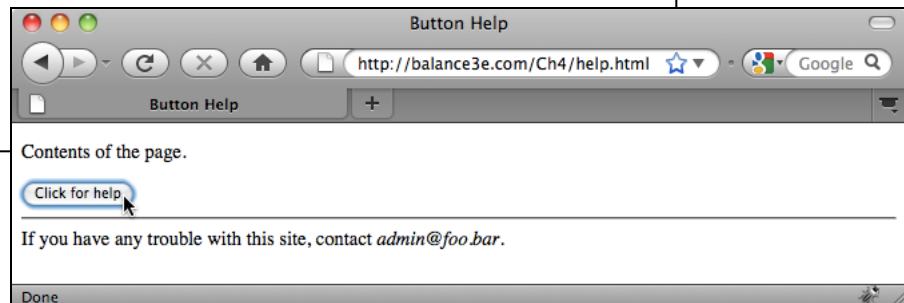


```
1. <!doctype html>
2. <!-- help.html                                     Dave Reed -->
3. <!-- Web page that displays a help message when a button is clicked. -->
4. <!-- ===== -->
5.
6. <html>
7.   <head>
8.     <title> Button Help </title>
9.   </head>
10.
11.   <body>
12.     <p>
13.       Contents of the page.
14.     </p>
15.     <input type="button" value="Click for Help"
16.           onclick="document.getElementById('outputDiv').innerHTML=
17.             'If you have any trouble with this site, ' +
18.             'contact <i>admin@foo.bar</i>.';">
19.
20.     <hr>
21.     <div id="outputDiv">
22.       Welcome to my site!
23.     </div>
24.   </body>
25. </html>
```

initially,
outputDiv
displays a
welcome message



when the button
is clicked, that
message is
replaced with
different text



Error Messages



an error in the format of an HTML or JavaScript statements is known as a *syntax error*

- some syntax errors are ignored by the browser
 - e.g., misspelling an HTML tag name
- most JavaScript syntax errors will generate an error message
 - *Internet Explorer*: must configure the settings to pop up error messages
 - *Firefox*: must view the Error Console to see error messages
- recognizing common errors allows you to avoid them in the future

Error: unterminated string literal

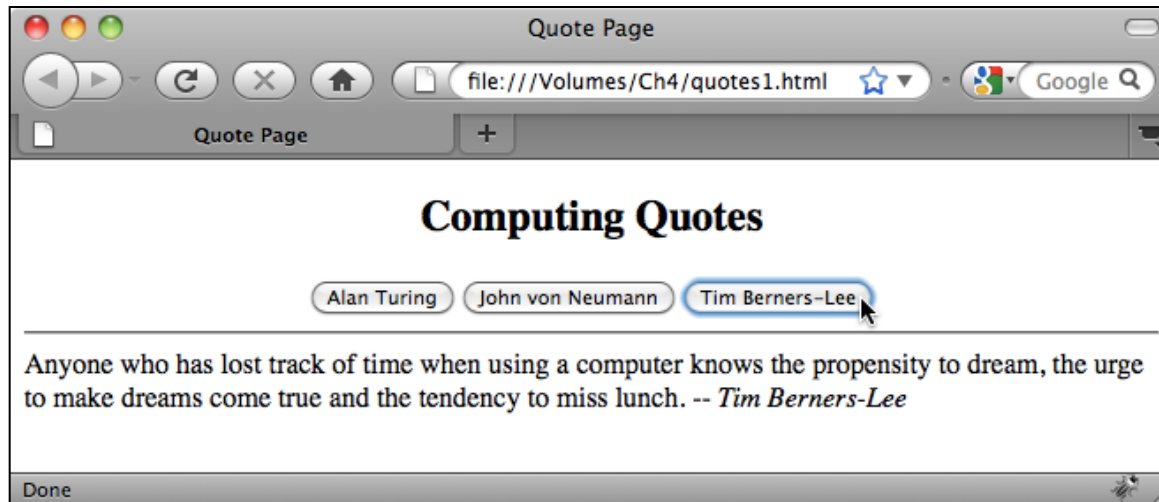
```
alert('This example is illegal because the string is broken  
      across lines');
```

Error: missing) after argument list OR

Error: missing ; after statement

```
alert('This example is illegal because '  
      'there is not a + connecting the pieces');
```

Example: Quotations Page

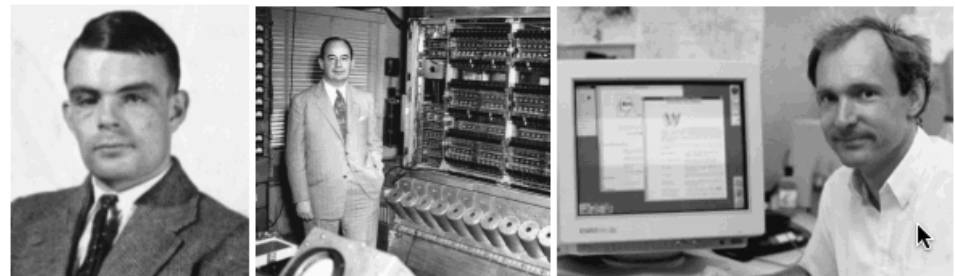


suppose we want to develop a page that displays favorite quotations in a page division

could use:

- a button for each quote, clicking the button displays the quote
- an image for each quote, moving the mouse over the image displays the quote

Computing Quotes



Anyone who has lost track of time when using a computer knows the propensity to dream, the urge to make dreams come true and the tendency to miss lunch. -- Tim Berners-Lee

Done

Apostrophes in Strings



if you want to display text that contains apostrophes, must be careful

```
<input type="button" value="welcome"
      onclick="document.getElementById('outputDiv').innerHTML=
               'welcome to my page. I'm glad you're here.';">
```

- will cause an error, since the apostrophe in I'm will be interpreted as the closing quote in the string

to display an apostrophe in text, must precede it with an *escape character* (i.e., a backslash)

```
<input type="button" value="welcome"
      onclick="document.getElementById('outputDiv').innerHTML=
               'welcome to my page. I\'m glad you\'re here.';">
```

- the escape character tells the browser that the apostrophe that follows is part of the string, not the closing quote