Menus, Forms, and Dialogs in XHTML/CSS/JavaScript

The *input* tag is the near-catch-all XHTML construct for defining form/dialog elements within a web page; the *type* attribute determines the element

- *button* is a button

- *text* is a standard text field

- *radio* is a radio button; use the *name* property to indicate which radio buttons are "grouped" together

- *check* is a check box

- *password* is a password field

Other form/dialog-related tags include the following:

- *select* defines either a drop-down menu or a scrolling list (the presence of the *size* attribute renders this element as a list, which makes sense, since only a scrolling list needs this information *a priori*)

- *option* tags within a *select* block define the individual choices within the drop-down menu or scrolling list

- *textarea* defines a multiline text area

All of these elements must be nested within a *form* tag, which delimits related sets of these user interface components; as with any XHTML element, defining an *id* attribute facilitates easy programmatic access

Simulating a Menu Bar

- Web page "menu bars" are typically *div* blocks with some form of CSS styling

- "Menu items" within these menu bars are either links (*a*), *div*s, or *span*s with an assigned *onclick*, *onmousedown*, or *onmouseover* event handler; this handler makes the corresponding "pull-down menus" appear

- These "pull-down menus" are also *div* blocks, CSS-styled to resemble menus, that are either toggled with the CSS *display* property or dynamically added to/removed from the XHTML document

- "Terminal" menu items are typically links to other pages, or else elements with additional *onclick* behavior

- Simulated menu bars are sufficiently common that CSS/ JavaScript libraries for implementing them are available

- Newer web browsers support visual effects such as fades, shadows, and/or animations, thus giving the user interface an extra level of, er, "dynamicness"

- Since, in the end, these types of "menus" are really just animated/coordinated blocks of XHTML, they do *not* have to be nested within a *form* block

- Popup (e.g., right-click) menus tend to be avoided, since they may clash with the web browser's own native popup menu

Web Server Activities

- Note that this discussion is limited to the content *within* a browser, *after* it has been delivered by a web server

- A fully-realized web application includes logic in the web server that:

  - Builds the web page dynamically

  - Binds menu/form/dialog elements within the web page to domain objects/controllers on the server

  - May customize the pages depending on the context or current user

- Ajax adds yet another wrinkle to web applications

  - Web pages can dynamically load new content from the web server, using the same URL mechanism

  - Loading a URL in this manner does *not* change the current address displayed by the web browser

  - The content is typically delivered as XML (thus the "x" in "Ajax"), which is then parsed in the web browser; in reality, the content can be anything, including images, plain text, or more XHTML

  - Incoming content is delivered via events as well, allowing the web page to continue "running" while waiting for data to arrive (thus the "A" for "asynchronous" in "Ajax")