

Internet Fundamentals & Introduction to Web Technologies

Course: IT (044615)

Lecture: 6

**Dynamic Documents
With JavaScript**

Dr. Ramez Hajislam

6.1 Introduction

- Using DOM, JavaScript can change the document in which it is embedded
- Elements can be move
- Style can be changed
- Visibility can be changed

6.2 Element Positioning

- CSS provides powerful tools to position elements in a web page
- The position property specifies the position mode
 - Value is absolute or relative or static
- The left and top properties specify element position
 - A positive value of top pushes the element down
 - A positive value of left pushes the element to the right
 - A negative value of top pushes the element up
 - A negative value of left pushes the element to the left

6.2 Absolute Positioning

- Absolute position specifies where an element appears relative to the containing element
- Example absPos.html uses absolute positioning to overlay two pieces of text
 - The width property is used to control the overlapping
- Example absPos2.html illustrates the positioning of text relative to a containing element

6.2 Static and Relative Positioning

- In static position mode, the default, elements are placed left to right, top to bottom
 - The top and left properties are ignored for static positioning
- Relative position mode allows changing position relative to where the element would be with static positioning
- This could be used to create superscripts or subscripts by moving text up or down relative to its normal position
- Example relPos.html illustrates positioning large text so that it looks to be centered with the remaining text

6.3 Moving Elements

- JavaScript code can move elements by changing the top and left properties
 - Note that the position mode has to be relative or absolute for this to work

6.3 Moving Elements

- Example mover.html illustrates dynamically placing elements
 - Text input fields are provided to enter the x and y coordinates desired for the displayed image
 - An image element has an id attribute and style to specify it as absolute position
 - An event handler on a button gets values from the text fields and uses those as parameters to a JavaScript function
 - The function gets a style node from the image element (variable dom)
 - The top and left properties of the style element are changed (note the px appended as a unit)

6.4 Element Visibility

- Example showHide.html illustrates hiding and showing an element by manipulating the visibility property
- The JavaScript code accesses the style node for the image element (variable dom)
- The visibility property of the style node is altered to change the visibility of the element

6.5 Changing Colors and Fonts

- Colors and font properties can be manipulated through the style property of an element

6.5 Changing Colors

- Example `dynColors.html` illustrates setting background and foreground colors
- The `change` event is used which triggers a change depending on which text box was used
- Note `this` is used to refer to the input tag triggering the event

6.5 Changing Fonts

- The dynLink example illustrates changing font properties using JavaScript
- A mouseover event on a link causes the font to change
- A mouseout event on the same link causes the font to change back to the original

6.6 Dynamic Content

- By manipulating the DOM tree representing the document, the document content can be changed
- The dynValue.html example illustrates dynamic content by changing the content of a text area when the mouse moves over other components
- The mouseover and mouseout events are used to control this
- The value property of the 'help' box is used to change the content

6.7 Stacking Elements

- The z-index style property can be used to govern the layering of elements in the display
 - If two elements both cover a part of the window, the element with the higher z-index value will cover the other one
 - Think of a artist painting the document content on the screen. Elements with lower z-index are painted before those with higher z-index
- The stacking.html example illustrates manipulating the z-index property dynamically

6.8 Locating the Mouse Cursor

- An event object created for a mouse related event has properties that give the coordinates of the mouse position at the time of the event
 - clientX and clientY give the position in pixels relative to the upper left-hand corner of the browser window
 - screenX and screenY give the mouse position relative to the upper left-hand corner of the screen
- The event object is available as an object named event
 - In Mozilla/Firefox the object can be accessed by passing it as a parameter to the event handler
 - In Internet Explorer, the object is a global
- The where.html example illustrates these points

6.9 Reacting to a Mouse Click

- The anywhere.html example is another example using mouse location information
- The example uses mousedown and mouseup events to drive the action
- The position and visibility of an element are manipulated by the event handler

6.10 Slow Movement of Elements

- JavaScript provides methods to time activities
- `setTimeout` will execute some JavaScript code at one time in the future
- `setInterval` will execute a JavaScript function at evenly spaced times in the future

6.10 Function setTimeout

- Takes two parameters
 - A string containing valid JavaScript code (similar to an event attribute value)
 - A non-negative number
- The call to this function immediately returns
- The numeric parameter specifies a number of milliseconds
- After that number of milliseconds, the JavaScript code is executed

6.10 Function setInterval

- This function takes a variable number of parameters
 - The first parameter is the name of a function, similar to what is used to register an event by assigning to a node property
 - The second parameter is a number, a number of milliseconds
 - The remaining parameters, if any, are used as parameters to the call of the function listed first
- This function call returns immediately
- Thereafter, at an interval given by the second parameter, the function is called over and over

6.10 Example moveText.html

- A string of text is moved from one position to another
- `setTimeout` is used to move the text in incremental steps
 - The function executed by `setTimeout`, `moveText`, will call `setTimeout` again provided the text is not in its final position
 - The call to `setTimeout` includes the current text position as parameters to `moveText`
- The text position is encoded as strings with units, as required by CSS, so this text must be decoded to numerical form to perform arithmetic

6.11 Dragging and Dropping Elements

- Example dragNDrop.html illustrates dynamically modifying event handlers in order to provide dragging and dropping functionality
- The grabber method is invoked on mousedown on an element
 - The grabber method assigns handlers for the target for the mouseup and mousemove events
 - The grabber method also determines coordinates for the target element so that it can be properly placed when the mouse moves
- The mouseup event handler, dropper, undoes the event registrations
- The mousemove event handler moves the target element