Tutorial 11 Notes

Events (slides 4-8)

* JavaScript programs run in response to events such as clicking an object on a form or closing a web page
* An event handler waits until the event occurs and then responds by running a function or command block to execute an action
* Event handlers can be added to a page element using the attribute <element onevent = “script”>
* Event handlers can also be defined as object properties using the command object.onevent = function;

Event handlers for the browser window

* onbeforeunload, oncopy, oncut, onerror, onload, onpaste, onresize, onunload (slide 9)
* window.onload = init runs init function when the page loads (slide 10)
* Apply the following event handler to respond to a mouse click (slide 12) object.onclick = function;

Event object (slides 15 and 16)

* An event object is an object that contains properties and methods associated with an event
* For example, the action of clicking a mouse button generates an event object containing information such as which mouse button was clicked, where in the page it was clicked, the time at which it was clicked, and so forth
* Event object properties and methods are listed in Figure 11-10 (slide 17)

Object properties (slides 19 to 22)

* Inline styles for each page element can be applied using the following style attribute <element style = “property:value”>
* The equivalent command in JavaScript is object.style.property = “value”;

Creating object selectors (slides 23 to 27)

* In JavaScript, to change the background color of all table cells, you must first define an object collection based on a CSS selector using the following querySelectorAll() method: document.querySelectorAll(selector) where selector is the CSS selector that the object collection is based on
* Once the object collection has been defined, change the background-color style of each td element by applying the backgroundColor property to the objects in the object collection
* Reference only the first element that matches a selector pattern using the following JavaScript method: document.querySelector(selector) where selector is a CSS selector

Mouse events (slide 28)

Mouse event object properties (slide 30)

Event model (slides 32 to 34)

* Describes how events and objects interact within the web page and web browser
* The process in which a single event is applied to a hierarchy of objects is part of the event model
* Once an event has been initiated, it propagates through the object hierarchy in three phases
* Capture phase: The event moves down the object hierarchy starting from the root element (the browser window) and moving inward until it reaches the object that initiated the event
* Target phase: The event has reached the target of the event object and no longer moves down the object hierarchy
* Bubbling phase: The event propagates up the object hierarchy back to the root element (browser window) where the propagation stops

Adding an event listener (slides 36 to 40)

* Only one function can be applied to an event handler at a time
* Unlike event handlers, more than one function can be applied to an event using event listeners
* Add an event listener to an object by applying the addEventListener() method
* object.addEventListener(event, function [, capture = false]);
* The event model allows to remove event listeners from the document by applying removeEventListener() method object.removeEventListener(event, function [, capture = false]);

Controlling event propagation (slides 41 to 43)

* Apply the following preventDefault() method to the event object to prevent the occurrence of the browser’s default actions: evt.preventDefault()

Keyboard events (slide 44)

Keyboard event properties (slide 46)

Modifier keys: Alt, Ctrl, Shift, and Command keys (slide 49)

Working with functions as objects (slides 55 and 56)

* Everything in JavaScript is an object, including functions
* Anything that can be done with an object can be done with a function, including storing a function as variable and storing a function as an object property

Function declarations and function operators (slides 57 to 59)

The following hello() function is created using the function declaration format:

function hello(){

 alert(“Welcome to Hanjie!”);}

Function operator: The definition of the function becomes the variable’s “value”

var hello = function (){

 alert(“Welcome to Hanjie!”);}

Displaying dialog boxes (slides 67 to 70)

* Alert dialog box can be created using the following alert() method: alert(text)where text is the message displayed in the alert dialog box
* When an alert dialog box is displayed, the execution of the program code halts until the user clicks the OK button in the dialog box
* JavaScript also supports confirm and prompt

**Hanjie Interactive Puzzle** (jpf\_hanjie.html, jpf\_grids1.js and jpf\_hanjie.js programs)