# Lecture Notes 7
# Google Map APIs, JSON, JQuery, and AJAX

**Anand Tripathi**

**CSci 4131**

**Internet Programming**

## Topics

- Assignment 3 on Google Maps (Rohit and Samarth)
- Programming with Google Map APIs
- JQuery
  - JavaScript library
- JSON
  - JavaScript Object Notation
    - How to represent an object as string?
    - Conversion from object to string, and string to object
- AJAX
  - Asynchronous JavaScript and XML

## Google Map APIs

## Google Map APIs

- With Google Map APIs you can include maps in your web application.
- You can put markers on a map.
- You can attach  information with some locations on the map which pops-up in a new window on mouse clicks on some icons.
- You can show directions between points on the map.
- You can superimpose traffic flow data on a map.
- Many more features including places search, heat maps, etc.

## Google Map APIs

- Reference page for Google Map APIs
- https://developers.google.com/maps/documentation/javascript/tutorial
- How to include a Google map
- https://developers.google.com/maps/documentation/javascript/examples/map-simple
- How to place a marker
- https://developers.google.com/maps/documentation/javascript/examples/marker-simple
- How to write something in the information page
- https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple
- How to capture the event on a Google Map
- https://developers.google.com/maps/documentation/javascript/events
- How to use directions services of the Google Maps API
- https://developers.google.com/maps/documentation/javascript/examples/directions-simple
- How to use traveling modes in directions
- https://developers.google.com/maps/documentation/javascript/examples/directions-travel-modes

## Map Object and Map Options

- Google APIs provide a JavaScript class called Map which is used for display a map on a page.

```
var someMap = new google.maps.Map(
                        document.getElementById("map-canvas"),
                        mapOptions);
```

- In the document body you define a place holder for a map as follows:

```
<div id=“map-canvas” style=“width:100%; height: 100%”>    </div>
```

- Map option define:
  - latitude and longitude of the center of the map
  - Zoom level for displaying the map

## Map Options

```
function initialize() {
  var mapOptions =  {
       zoom: 8,
      center: new google.maps.LatLng(-34.397, 150.644)
   }
   var map = new google.maps.Map(
            document.getElementById('map-canvas'),
            mapOptions);
}
```

```
google.maps.event.addDomListener(window, 'load', initialize);
```

Another way to specify execution of initialization function on load:

```
<body onload="initialize()">   …….  </body>
```

**Latest way:**

```
<script async defer
src=https://maps.googleapis.com/maps/api/js?key=API_KEY&callback=initialize></script>
```

# Simple Map example

## *(old way of programming)*

**See https://developers.google.com/maps/documentation/javascript/examples/map-simple**

```
<head>
  <title>Simple Map</title>
  <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
  <script>
     var map;
     function initialize() {
       var mapOptions = {
             zoom: 8,
             center: new google.maps.LatLng(-34.397, 150.644)
        };
        map = new google.maps.Map(document.getElementById('map-canvas'),   mapOptions);
     }
     google.maps.event.addDomListener(window, 'load', initialize);
  </script>
</head>
<body>   <div id="map-canvas“style=“width:100%; height: 100%”></div>
 </body>
```

# Simple map example

## *(current way of programming)*

See: https://developers.google.com/maps/documentation/javascript/examples/map-simple

```
<body>
   <div id="map"></div>
   <script>
    var map;
    function initMap() {
         map = new google.maps.Map(
                       document.getElementById('map'),
                       {
                         center: {   lat: -34.397,   lng: 150.644   },
                         zoom: 8
                       }
                  );
      }  // end of function initMap
    </script>
    <script src="https://maps.googleapis.com/maps/api/js?key=API_KEY&callback=initMap"
        async defer></script>
  </body>
```

You need to create an API_KEY on google developers site and paste it here.

# LatLng Object

- You can construct an object that give a location (position) on map.
- position = new google.maps.LatLng(-34.397, 150.644)
- You can get the two coordinates using methods shown below

  position.lat()

  position.lng()
- A click event on a map has LatLng object for the event position on the map. You can read it in an event handler

```
google.maps.event.addListener  ( map, ‘click',
      function(event) { eventPosition = event.latLng; …. }
);
```

## Map Types

- Reference:
- https://developers.google.com/maps/documentation/javascript/3.exp/reference#MapTypeId
- google.maps.MapTypeId
  - ROADMAP
  - TERRAIN
  - SATELLITE
  - HYBRID

- setTilt - angle of aerial view

## Example of Map Type

```
var mapOptions = {
    zoom:19 ,
    center: myLatlng,
    mapTypeId : google.maps.MapTypeId.SATELLITE
  }
  var map = new
          google.maps.Map (document.getElementById('map-canvas'),
                                          mapOptions);
   map.setTilt(45);
```

- See this example of Keller Hall view  (tilt = 45)

## Superimposing Traffic View

- var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

- var trafficLayer = new google.maps.TrafficLayer();
- trafficLayer.setMap(map);

See this example

## Putting a Marker on a Map

**https://developers.google.com/maps/documentation/javascript/examples/marker-simple**

**Maker Object** has several properties:

- *position*: which is specified as an object containing latitude and longitude
- *map*: map object with which this marker is associated
- *title*: a string title which will show when mouse is on this marker
- *icon:* set this property to desired icon image other than the default when creating a new marker

  Example: icon" "icon.png"

- See reference page:
- https://developers.google.com/maps/documentation/javascript/reference#Marker

## Putting a Marker on a Map

**https://developers.google.com/maps/documentation/javascript/examples/marker-simple**

```
function initialize() {
  var myPosition = new google.maps.LatLng(-25.363882,131.044922);
  var mapOptions = {
    zoom: 4,
    center: myPosition
  }
  var map = new google.maps.Map(document.getElementById('map-canvas'),
mapOptions);

  var marker = new google.maps.Marker(
      {  position: myPosition,
         map: map,
         title: 'Hello World!' }    );
}
google.maps.event.addDomListener(window, 'load', initialize);
```

## Adding, Removing, Hiding Markers

see https://developers.google.com/maps/documentation/javascript/examples/marker-remove

```
var map;
var markers = [];

function initialize() {
  var haightAshbury = new google.maps.LatLng(37.7699298, -122.4469157);
  var mapOptions = {
    zoom: 12,
    center: haightAshbury,
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };

map = new google.maps.Map(document.getElementById('map-canvas'),
      mapOptions);

google.maps.event.addListener(map, 'click', function(event) {
    addMarker(event.latLng);
  });
```

## Adding, Removing, Hiding Markers

see https://developers.google.com/maps/documentation/javascript/examples/marker-remove

```
// Add a marker to the map and push to the array.
function addMarker(location) {
  var marker = new google.maps.Marker({
    position: location,
    map: map
  });
  markers.push(marker);
}


// Sets the map on all markers in the array.
function setAllMap(map) {
  for (var i = 0; i < markers.length; i++) {
    markers[i].setMap(map);
  }
}
```

## Adding, Removing, Hiding Markers

see https://developers.google.com/maps/documentation/javascript/examples/marker-remove

```
// Removes the markers from the map, but keeps them in the array.
function clearMarkers() {
  setAllMap(null);
}

// Shows any markers currently in the array.
function showMarkers() {
  setAllMap(map);
}

// Deletes all markers in the array by removing references to them.
function deleteMarkers() {
  clearMarkers();
  markers = [];
}

google.maps.event.addDomListener(window, 'load', initialize);
```

## Info Window on a Map

see: https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple

See the Info Window APIs under the Overlays section.

These  pop-up overlay windows are created by the Google Map API library functions

You can create a pop-up overlay window when a user clicks on some icon or marker.

You can set various properties such a content, width, position, zIndex.

https://developers.google.com/maps/documentation/javascript/reference#InfoWindow

## Info Window on a Map

see: https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple

```
function initialize() {
  var myLatlng = new google.maps.LatLng(-25.363882,131.044922);
  var mapOptions = {
    zoom: 4,
    center: myLatlng
  };
var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
var contentString =  “some context string to be displayed”;
var infowindow = new google.maps.InfoWindow({
      content: contentString
  });
var marker = new google.maps.Marker({
      position: myLatlng,
      map: map,
      title: 'Uluru (Ayers Rock)'
  });
google.maps.event.addListener(marker, 'click', function() {
    infowindow.open(map,marker);
  });
}
google.maps.event.addDomListener(window, 'load', initialize);
```

## Using directions service

see: https://developers.google.com/maps/documentation/javascript/examples/directions-simple

```
function initMap() {
  var directionsService = new google.maps.DirectionsService;
  var directionsDisplay = new google.maps.DirectionsRenderer;
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 7,
    center: {lat: 41.85, lng: -87.65}
  });
  directionsDisplay.setMap(map);

  var onChangeHandler = function() {
    calculateAndDisplayRoute(directionsService, directionsDisplay);
  };
  document.getElementById('start').addEventListener('change', onChangeHandler);
  document.getElementById('end').addEventListener('change', onChangeHandler);
}
```

## Using directions service

see: https://developers.google.com/maps/documentation/javascript/examples/directions-simple

```
function calculateAndDisplayRoute(directionsService, directionsDisplay) {
  directionsService.route(
    {
    origin: document.getElementById('start').value,         // LatLng object
    destination: document.getElementById('end').value, // LatLng object
    travelMode: google.maps.TravelMode.DRIVING
    },
    function(response, status) {
          if (status === google.maps.DirectionsStatus.OK) {
             directionsDisplay.setDirections(response);
           } else {
              window.alert('Directions request failed due to ' + status);
           }
     });
}
```

## Using traveling modes in directions

See: https://developers.google.com/maps/documentation/javascript/examples/directions-travel-modes

```
function calculateAndDisplayRoute(directionsService, directionsDisplay) {
  var selectedMode = document.getElementById('mode').value;
  directionsService.route({
    origin: {lat: 37.77, lng: -122.447},
    destination: {lat: 37.768, lng: -122.511},
    travelMode: google.maps.TravelMode[selectedMode]
  }, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    } else {
      window.alert('Directions request failed due to ' + status);
    }
  });
}
```

## JQuery

## JQuery

- JQuery is a JavaScript library containing in-built functions to simplify JavaScript programming
- It provides a large collection of functions.
- JQuery simplifies many common JavaScript programming tasks such as
  - DOM elements navigation and manipulation
  - Javascript events
  - Manipulating HTML and CSS elements and attributes
  - AJAX programming

# JQuery Library

- Using JQuery library
  - You can download JQuery library from
  - http://jquery.org/
  - This library needs to be included in the javascript code using the JQuery functions

```
<head>
<script type="text/javascript" src="jquery.js"></script>
</head>
```

  - You can also use the hosted JQuery library available from Google or Microsoft, instead of downloading

```
<head>
<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
</head>
```

## JQuery Syntax

- Basic syntax
  - $(<selector>).action()
  - e.g. $("p").hide() – hide all <p> elements
- Selector is used to select one or more HTML elements on which the specified method is to be applied
  - $(this) – selects current HTML element
  - $("element-type") – selects all elements of a particular type such as <p> or <div>
  - $("#id-name") – select all elements with the specified id
  - $(".class-name") – selects all elements with the given class

Example JQuery program

## JQuery Functions

- Manipulating HTML elements
  - Selecting HTML element(s) using selectors
  - Changing html text of the selected element(s)
    - $(“p”).html(“this is new text”)  //overwrites existing text
    - $(“p”).append(“this is new text”) //appends to the existing text
  - Modifying CSS attributes
    - $(“p”).css("background-color","yellow“);
- Events
  - Executing a given code when a particular event occurs
    - $("button").click(function() {..some code... } )
- Effects
  - Hide/show elements
    - $(“p”).hide

## Example JQuery program

```
<html>
<head>   <title>  JQuery Example  </title>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
</head>
<body>
<p id="para1" >  This is the first paragraph.  </p>
<p id="para2" >  This is the second paragraph.  </p>
<input type="button"  value="Replace text of all paragraphs" onClick='$("p").html("this is new
text")'  > <br/>
<input type="button"  value="Replace text seccnd paragraph" onClick='$("p:nth-
child(2)").html("paragraph 2 text changed")'  > <br/>
<input type="button"  value="Append text seccnd paragraph" onClick='$("p:nth-
child(2)").append(" we will continue this next year.")'  > <br/>
<input type="button"  value="Change font color to red" onClick='$("p").css("color", "red")'  >
<br/>
<input type="button"  value="Hide all paragraphs" onClick='$("p").hide()'  > <br/>
<input type="button"  value="Show all paragraphs" onClick='$("p").show()'  > <br/>
<input type="button"  value="Show photo" onClick='$("#image1").attr("src", "lotus-1.jpg") '  >
<br/>
<img id="image1" src="" />
</body>
</html>
```

## AJAX

# AJAX

**Chapter 16 of Deitel book**

- Asynchronous JavaScript and XML
- It provides an asynchronous model to fetch some object over the Internet using the HTTP protocol and selectively update an object in the browser’s document object.
- One does not need to reload an entire page again just to get some part of the page to be updated with some new content.

See example 16-5 here

## AJAX Model

- It defines XMLHttpRequest object to send an HTTP request to a server.
- With this object an asynchronous event handler is registered as a callback function when a response from the server is received.
- The callback function appropriately updates the browser window’s document object to selectively update the displayed information.

## Methods of XMLHttpRequest Object

- open -- It initializes the request object with two <u>mandatory parameters</u>:
  - Method name as GET and POST
  - URL
  - Optional boolean parameter when “true” means that the request is to be made asynchronous. This is the default value.
- send -- it sends the request to the server.
  - One optional parameter which specifies the data to be POSTed. Null by default.

## Methods of XMLHttpRequest Object

- setRequestHeader -- It sets the request header:
  - Two parameters: header and value
- getResponseHeader -- it retrieves the value for a specified response header.
- getAllResponseHeaders
  - Returns an array of all response headers
- abort

## Properties of XMLHttpRequest Object

- onreadystatechnage This stores the callback function
- status - HTTP response status e.g. 200
- readyState
  - Keeps track of progress in the callback function
  - Values are 0..4

  0 means uninitialized request

  1 means request is loading

  2 means request is loaded

  3 means server is sending data

  4 means request has been completed

# AJAX Example

**See example 16-5 here from Deitel’s**

```
<head>
  <style type="text/css">
    .box { border: 1px solid black;
           padding: 10px }
  </style>
  <title>Switch Content Asynchronously</title>
  <script type = "text/javascript" language = "JavaScript">
    var asyncRequest; // variable to hold XMLHttpRequest object
```

# AJAX Example

**See example 16-5 here**

```
// set up and send the asynchronous request.
 function getContent( url )   {
   // attempt to create the XMLHttpRequest and make the request
   try    {
     asyncRequest = new XMLHttpRequest();
        // create request object  register event handler
     asyncRequest.onreadystatechange = stateChange;
     asyncRequest.open( 'GET', url, true ); // prepare the request
     asyncRequest.send( null );
           // send the request, arg is null   - no data
           // this argument is used in POST requests
   } // end try
   catch ( exception )  {
     alert( 'Request failed.' );
   } // end catch
 } // end function getContent
 function stateChange()  {

        if ( asyncRequest.readyState == 4 && asyncRequest.status == 200 )
        {
          document.getElementById( 'contentArea' ).innerHTML =
                asyncRequest.responseText; // places text in contentArea
        } // end if
      } // end function stateChange

      // clear the content of the box
      function clearContent()  {
        document.getElementById( 'contentArea' ).innerHTML = '';
      } // end function clearContent

</script>
```

```
<body>
  <h1>Mouse over a book for more information.</h1>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/cpphtp6.jpg"
    onmouseover = 'getContent( "cpphtp6.html" )'
    onmouseout = 'clearContent()'/>

  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/iw3htp4.jpg"
    onmouseover = 'getContent( "iw3htp4.html" )'
    onmouseout = 'clearContent()'/>

  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/jhtp7.jpg"
    onmouseover = 'getContent( "jhtp7.html" )'
    onmouseout = 'clearContent()'/>
```

```
   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vbhtp3.jpg"
      onmouseover = 'getContent( "vbhtp3.html" )'
      onmouseout = 'clearContent()'/>

   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vcsharphtp2.jpg"
      onmouseover = 'getContent( "vcsharphtp2.html" )'
      onmouseout = 'clearContent()'/>

   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/chtp5.jpg"
      onmouseover = 'getContent( "chtp5.html" )'
      onmouseout = 'clearContent()'/>

   <div class = "box" id = "contentArea">&nbsp;</div>
</body>
```

## AJAX programming using JQuery

- Provides a simple ‘load’ method to asynchronously fetch data and modify a DOM element
  - **$(selector).load(url,*data,callback*)**

```
$("button").click(function(){
   $("div").load('test1.txt');
 });
```

- No need of preparing and sending XMLHttpRequest, load method performs the required task of sending and fetching data asynchronously

See Modified version of example 16-5 using JQuery

## Ajax with JQUERY

- See modified version of example 16-5 using JSON
- This is the original exam show descriptions of the books.

- JQuery provides a convenient and much simpler abstraction to perform Ajax operation.

- It provides a simple function

    load(url)

```
<head>
  <title>Switch Content using Ajax with JQuery?</title>
  <style type="text/css">
    .box { border: 1px solid black;    padding: 10px }
  </style>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <script type = "text/javascript" language = "JavaScript">

    function getContent(url) {
      $("#contentArea").load(url);
    }

    function clearContent()
    {    document.getElementById( 'contentArea' ).innerHTML = '';
    } // end function clearContent
  </script>  </head>
```

```
<body>
   <h1>Mouse over a book for more information.</h1>
   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/cpphtp6.jpg"
      onmouseover = 'getContent( "cpphtp6.html" )'
      onmouseout = 'clearContent()'/>
   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/iw3htp4.jpg"
      onmouseover = 'getContent( "iw3htp4.html" )'
      onmouseout = 'clearContent()'/>
   <img src =
      "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/jhtp7.jpg"
      onmouseover = 'getContent( "jhtp7.html" )'
      onmouseout = 'clearContent()'/>
```

```
 <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vbhtp3.jpg"
    onmouseover = 'getContent( "vbhtp3.html" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vcsharphtp2.jpg"
    onmouseover = 'getContent( "vcsharphtp2.html" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/chtp5.jpg"
    onmouseover = 'getContent( "chtp5.html" )'
    onmouseout = 'clearContent()'/>
  <div class = "box" id = "contentArea">&nbsp;</div>
</body>
```

## JSON

## JSON

- JSON – JavaScript Object Notation
- A syntax for storing and communicating text data similar to XML
- JSON is lightweight than XML – smaller than XML and easier to parse.
  - Efficient for communicating data to small handheld devices such as smartphones

## JSON Syntax

- JSON data is stored as name-value pairs
- Supports *arrays* – collection of name-value pairs and *objects* – collection of arrays and name-value pairs
- A name-value pair is stored as “name”: “value”
  - e.g. “firstname”:”John”
  - Values can be numbers, strings, booleans or an object or array
- Objects are denoted by { }
  - e.g {“firstname”:”John”,”lastname”:”Doe”}
- Arrays are denoted by [ ]

## JSON Example

- An array of 3 employee records, each record is an object with two name-value pairs – firstname and lastname

```
{
  "employees": [
  { "firstName":"John" , "lastName":"Doe" },
  { "firstName":"Anna" , "lastName":"Smith" },
  { "firstName":"Peter" , "lastName":"Jones" }
  ]
}
```

## JSON with JavaScript

- JSON data can be converted into a JavaScript object by parsing the JSON text data.

  JSON String representation ➔ JavaScript object

  var data = JSON.parse(<JSON text data>);

- JavaScript object can be covered to a JSON string

  JavaScript object ➔ JSON String representation

  var jsonString = JSON.stringify (someObject)

- Communicating data from server to client
  - JSON can be used to fetch data from server, similar to fetching html or xml files.
  - JSON data is stored in files using extension '.json'
  - The MIME type for JSON text is "application/json"

## See example   Example of encoding an object  into a string.

```
<!DOCTYPE html> <html>
<body> <h2>Encode Object to JSON String</h2>
<p> JSON String Content:<br/>
<span >Name: John Tel: 111-222-3333</span><br/>
<span >Name: Ann Tel: 111-222-3333</span><br/>
<span >Name: Peter Tel: 111-222-3333</span><br/> </p>
<script>
   var test = { }; // Object
   test['employees'] = []; // Array
    var item = {};
         item['Name'] = 'John'; item['Tel'] ='111-222-3333'; test['employees'].push(item);
    var item2 = {};
           item2['Name'] = 'Ann'; item2['Tel'] ='111-222-3333'; test['employees'].push(item2);
    var item3 = {};
        item3['Name'] = 'Peter'; item3['Tel'] ='111-222-3333'; test['employees'].push(item3);
    var   json = JSON.stringify(test);
    alert(json); </script>
</body>
</html>
```

## See example **Example of decoding a JSON string object to JavaScript object**

```
<html>
<body>
<h2>Create Object from JSON String</h2>
<p>   Employees Content:<br/>
<span id="content"></span><br/>  </p>

<script>
var txt = '{"employees":[{"Name":"John","tel":"111-222-3333"},{"Name":"Anna","tel":"111-
      222-3333"},{"Name":"Peter","tel":"111-222-3333"}]}';

obj = JSON.parse(txt);
document.getElementById("content").innerHTML=txt;
For  (  var i = 0;  i<obj.employees.length;  i++  )   {
    var employee = obj.employees[ i ] ;
    alert("NAME: " +employee.Name+", TEL: "+employee.tel );
}
</script>
</body>
</html>
```

## JSON with JavaScript - Example

- In this example, JSON data is name-value pairs containing book details such as price and publication date
  - e.g. {"price":“$190.9","date":"Nov 2008"}
  - for each book a json file is created containing above JSON data

See modified version of example 16-5 using JSON

## Example

cppht6.json file has the following JSON data:

```
{"BOOK":{"price":"$70.99","date":"Nov 1999"}}
```

## JSON with JavaScript - Example

```
function getContent( url) {
    try {
         asyncRequest = new XMLHttpRequest();
         asyncRequest.onreadystatechange = stateChange;
         asyncRequest.open( 'GET', url, true ); // prepare the request
         asyncRequest.setRequestHeader("Accept","application/json",
                        "charset=utf-8");  //specify that request is for JSON data
         asyncRequest.send( null ); // send the request
     } catch ( exception ) {
         alert( 'Request failed.' );
     }
 }
```

## JSON with JavaScript - Example

```
 function stateChange() {
      if ( asyncRequest.readyState == 4 && asyncRequest.status == 200
)
      {
          var bookData = asyncRequest.responseText;
         var obj = JSON.parse ( bookData );
          var price = obj.BOOK.price;
          var pubdate = obj.BOOK.date;
         document.getElementById( 'contentArea' ).innerHTML
                        = " Price = "+price+"  Publication Date = "+pubdate
;
      } // end if
}
```

```
<body>
  <h1>Mouse over a book for more information.</h1>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/cpphtp6.jpg"
    onmouseover = 'getContent( "cpphtp6.json" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/iw3htp4.jpg"
    onmouseover = 'getContent( "iw3htp4.json" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/jhtp7.jpg"
    onmouseover = 'getContent( "jhtp7.json" )'
    onmouseout = 'clearContent()'/>
```

```
 <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vbhtp3.jpg"
    onmouseover = 'getContent( "vbhtp3.json" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/vcsharphtp2.jpg"
    onmouseover = 'getContent( "vcsharphtp2.json" )'
    onmouseout = 'clearContent()'/>
  <img src =
    "http://test.deitel.com/examples/iw3htp4/ajax/thumbs/chtp5.jpg"
    onmouseover = 'getContent( "chtp5.json" )'
    onmouseout = 'clearContent()'/>
  <div class = "box" id = "contentArea">&nbsp;</div>
</body>
```