Chapter 14 Web and Mobile Mapping

Michael P. Peterson, USA

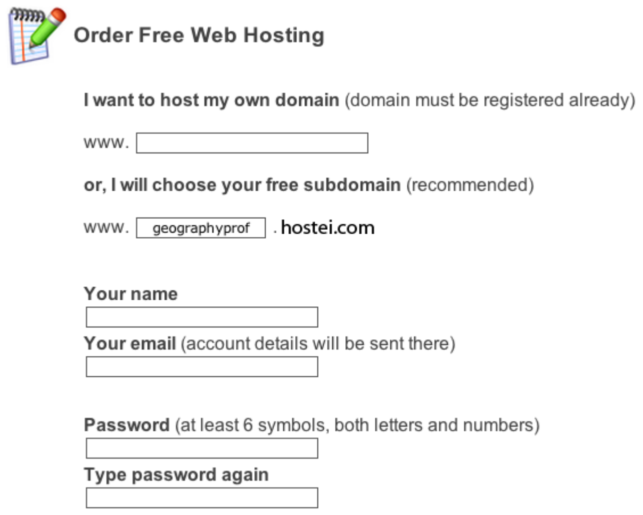
Abstract

Mapping has always been dependent on a set of tools, both for measuring the world and for making the map. Since 2005, a new set of online or cloud-based tools has been developed for making maps called Application Programmer Interfaces. These tools provide the additional benefit of being automatically available to anyone with an Internet connection. This chapter examines this new mash-up era in mapping and shows how to make and distribute maps using free mapping tools.

14.1 Introduction

It is difficult to overstate the importance of maps as a form of communication about the world. They help us understand both our surroundings and the space beyond our direct perception. Maps influence how we think about the world and how we act within it. They connect us to our environment. Each of us is a mapmaker, or cartographer, in the sense that we all make mental maps. Sometimes, we even need to draw these maps for others to help explain how to find a particular location.
The making of maps and the analysis of the underlying information have evolved into a science and are valuable skills for many different types of work. Many of the tools for mapping are now located in the cloud—a sophisticated system of hardware and software accessible through the Internet. Cloud-based mapping tools allow for very advanced forms of mapping. A further advantage of these online maps is that they can be easily made available to others.

Making maps in the cloud requires using a server. While it is possible to transform almost any



Order Free Web Hosting

I want to host my own domain (domain must be registered already)

www. [ ]

or, I will choose your free subdomain (recommended)

www. [geographyprof] . hostei.com

Your name

[ ]

Your email (account details will be sent there)

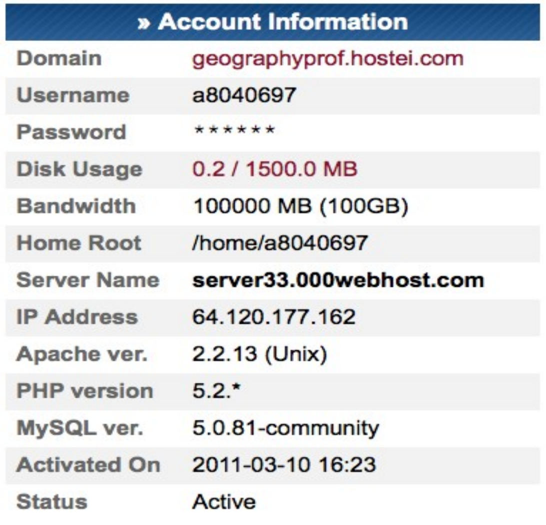
[ ]

Password (at least 6 symbols, both letters and numbers)

[ ]

Type password again

[ ]



» Account Information

Domain	geographyprof.hostei.com
Username	a8040697
Password	******
Disk Usage	0.2 / 1500.0 MB
Bandwidth	100000 MB (100GB)
Home Root	/home/a8040697
Server Name	server33.000webhost.com
IP Address	64.120.177.162
Apache ver.	2.2.13 (Unix)
PHP version	5.2.*
MySQL ver.	5.0.81-community
Activated On	2011-03-10 16:23
Status	Active

Figure 14.1a shows in the left panel the signup page for [http://000webhost.com.](http://000webhost.com) Account information is shown on the right. Choosing a subdomain is free. A charge is incurred for specifying a domain with a specific name, such as http://www.peterson.com but having the site choose a free domain like http://geographyprof.hostei.com is free. (© 2014 First Class Web Hosting)

Figure 14.1b shows the Account information.

computer into a server, it is easier and more secure to use an online hosting service. Cloud-hosting services such as Amazon Web Services and Microsoft Azure implement scalability in the sense that they can be upgraded to serve millions of simultaneous users. These services charge for disk space usage and server processing cycles.

Another option for hosting web pages is web-hosting services that only charge for disk storage. While these services are not as scalable, they are easier to use and offer free subdomain plans with up to 1,500 MB of disk space. Two such free web-hosting services are `000webhost.com` and `podserver.info`. The following sections describe how to create a website and serve maps using one of these free web-hosting services.

14.2 Servers in the Cloud

14.2.1 Making Space in the Cloud

Figure 15.1 depicts the sign-up page for the 000webhost.com provider and the resultant account information. Notice how a free subdomain is being requested under the hostei.com domain. While an address like http://geographyprof.com could be requested in the top line, it would incur a charge because it would represent a new domain. To obtain a free account, it is important to not register your own domain.

An e-mail address is needed to register for the free subdomain. The Account Information page shows that the web address that was assigned by the service provider is http://geographyprof.hostei.com (or http://64.120.177.162) and that 1,500 MB of free storage is available. The account information also shows the availability of the Apache web server, and other online tools including PHP and MySQL.

Included in most web-hosting services is a graphical interface to the services that are offered. This is called the control panel, or cPanel (see Figure 14.2). The tools handle e-mail, the editing of files, the scheduling of tasks, and account management. All of these tools represent open-source projects that are written and maintained by a small legion of programmers. File Manager is the most useful tool for managing files and building a website. MySQL and phpMyAdmin are used for building a database. Most web-hosting services use a similar cPanel to access server resources.

Figure 15.3 shows the File Manager window with access to tools for uploading and creating new files and directories (subfolders). Tools are also available to move, delete, and rename files. The file listing shows the name, type, and size of the file, while the Owner, Group, and Perms fields depict security settings. Mod Time indicates when the file was last modified. The files can be edited directly from this window by clicking “Edit” at the end of each file name.

The **public_html** folder is the directory from which all web files are served. If an html file is to be presented through a web page, it must reside in this folder. Usually, this folder contains a file called `index.htm` (or `index.php`) that is the first page that is accessed when the site is referenced. For example, if an address such as `http://geographyprof.hostei.com/CloudMapping/` is entered into a browser, the browser will look for a file called `index.htm` in a directory (folder) called `CloudMapping` that is itself located in the `public html` directory. That means that the following two addresses would display the same file:
`http://geographyprof.hostei.com/Online_Mapping/`
`http://geographyprof.hostei.com/Online_Mapping/index.htm`

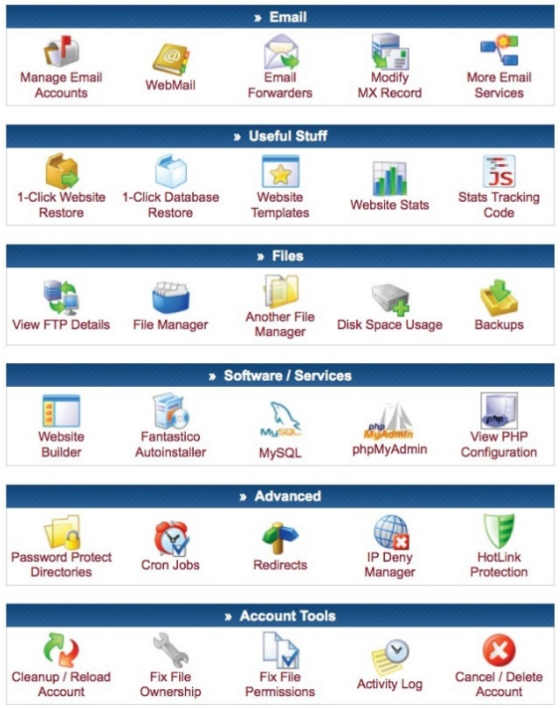
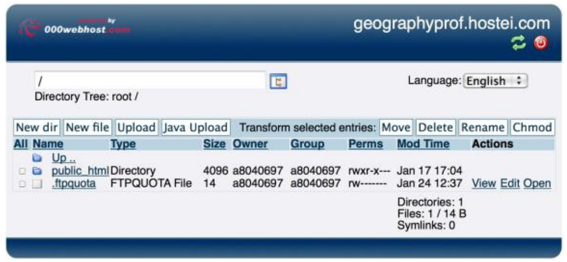


Figure 14.2 shows a standard webHosting control panel, called cPanel, which gives access to different tools. File Manager is the major program for uploading and editing files. (© 2014 First Class Web Hosting)



/
Directory Tree: root /
Language: English

New dir | New file | Upload | Java Upload | Transform selected entries: | Move | Delete | Rename | Chmod

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
	Up ..							
	public_html	Directory	4096	a8040697	a8040697	rwxr-x---	Jan 17 17:04	
	.ftpquota	FTPQUOTA File	14	a8040697	a8040697	rw-------	Jan 24 12:37	View Edit Open

Directories: 1
Files: 1 / 14 B
Symlinks: 0

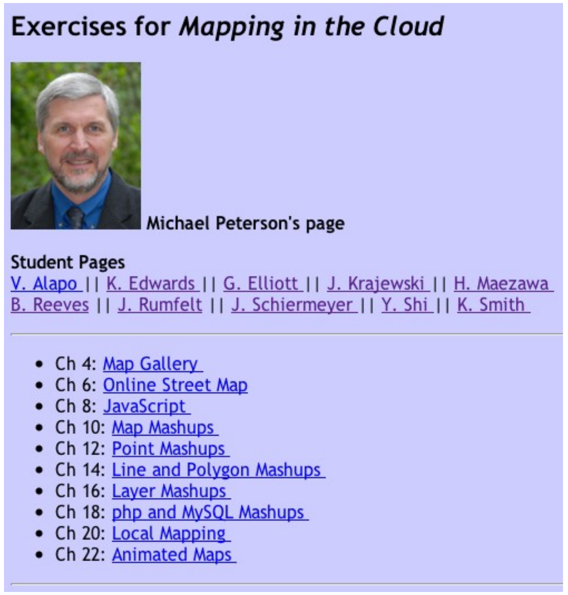
*Figure 14.3 shows the File Manager Window from an online hosting service. This service allows files to be created and edited. All files to be served through the web must be in the **public_html** directory/folder. (© 2014 First Class Web Hosting)*

Normally, this index.htm file serves as an entry point to the website and will have links to all other files in the directory.

The `index.htm` file will have a relatively simple structure—a title line followed by links to all of the assignments. This file could have a picture of the website owner and links to the websites of other students, as shown in Figure 14.4. The corresponding code shows how a picture is inserted using the

<img src=*filename*>

tag. The links to the student pages are separated by two vertical lines (“||”). The code for the entire index file can be obtained by searching for “Peterson Mapping in the Cloud.”


Exercises for *Mapping in the Cloud*

Michael Peterson's page

Student Pages
V. Alapo || K. Edwards || G. Elliott || J. Krajewski || H. Maezawa
B. Reeves || J. Rumfelt || J. Schiermeyer || Y. Shi || K. Smith

- Ch 4: Map Gallery
- Ch 6: Online Street Map
- Ch 8: JavaScript
- Ch 10: Map Mashups
- Ch 12: Point Mashups
- Ch 14: Line and Polygon Mashups
- Ch 16: Layer Mashups
- Ch 18: php and MySQL Mashups
- Ch 20: Local Mapping
- Ch 22: Animated Maps

```
<html>
<head></head>
<body bgcolor="#CCCCFF">
<h2> Exercises for <i>Mapping in the
Cloud</i> </h2>
<img src=peterson.jpg height=150><b>
Michael Peterson's page
</b><p>
<p>
<b> Student Pages </b><br>
<a href=http://victoriaA.site88.net> V.
Alapo </a> ||
<a href=http://mapsarefuntoo.web44.net> K.
Edwards </a> ||
<br> <hr>
<ul>
<li>
Ch 4: <a
href=http://maps.unomaha.edu/onlinemapping/
code04.zip>
Map Gallery </a><br>
<li>
Ch 6: <a href=code06.zip> Online Street
Map</a><br>
<li>
Ch 8: <a
href=http://maps.unomaha.edu/onlinemapping/
code08.zip>
JavaScript </a><br>
</a><br>
</ul>
<hr>
</body>
</html>
```

Figure 14.4 shows an example of an index.htm file that includes a picture, links to all other students, and links to the assignments. The code shows how to display a picture using the img tag, links to the pages for the other students, and to the assignments.

14.2.2 HTML

HTML is the building block of the web. It is the language that makes it possible to present information through web pages. It is also a container for scripting languages such as JavaScript and PHP. Some knowledge of HTML is necessary for presenting maps through the Internet.

HTML consists of tags that define the layout of the page (Willard 2009). It includes simple text codes, surrounded by the "<" and ">" delimiters, that specify how the document will appear in the browser. Other HTML tags also create links to documents or display a graphic file. An ordinary text editor can be used to enter the file (Notepad on Windows or TextEdit with Macintosh, with appropriate settings). Unlike word processors, these programs are intended for entering unformatted text. Once the files are created, they may be opened with a browser such as Explorer, Firefox or Chrome.

All HTML files begin with the **"<html>"** tag and end with the same code prefaced with a slash (e.g., "**</html>**"). The slash in front of the end code indicates that the HTML coding is finished. Technically, all HTML tags have a beginning and end, with the end tag indicated with "/." For example, the **<h1>** command is used to begin header text—larger text used for titles—and the **</h1>** code stops the header text format.

We can now display a map with the **img** tag. All file names that are referenced with img must end with GIF, JPG (or JPEG) or PNG as these are the common file types for the different browsers. The **img "src"** option is used to specify the file's URL address. In the example below, notice the image file designator **".gif"** at the end of the URL. The **img** command also includes a number of options that can be used to alter the size of the image or change its placement on the page. Like **hr, img** does not have a standard end tag. In strict XHTML formatting, it is written as

**<img src=map.png /><img src="http://maps.unomaha.edu/OnlineMapping/Chapter4/MapExample1.gif"
width="500" height="389">**

The **<embed>** tag is used to display graphic files that are not in the GIF, JPEG, or IMG format. Examples include Adobe's PDF, Flash, SVG, and QuickTime.

The format is identical to that of the **<img>** option:

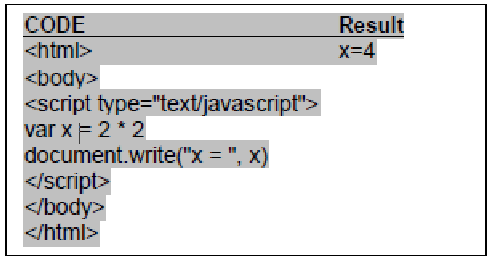
<embed src="http://maps.unomaha.edu/Cloud_Mapping/Chapter4/MapExample4.pdf" width="500" height="389">

One of the most used tags in HTML is the anchor that is usually used with a text string to create a hypertext link. It can also be used with the **img** tag to create a hyperimage link. The format of the anchor command is as follows:

<a href=http://maps.unomaha.edu/Cloud_Mapping/Chapter4/MapExample4.pdf> Click for PDF file</a>

14.2.3 JavaScript

By itself, HTML is simply a page formatting language. In combination with JavaScript, an HTML page can execute computer code (W3Schools.com 2011). Once relegated to geeks, programming is now being viewed as a form of expression, as the “amplification of thinking,” and a necessary skill. This “coding as literacy” concept is promoting new ideas about the importance of being able to program. Online programming sites like CodeAcademy are becoming increasingly popular. Khan Academy has introduced a similar suite of free online programming exercises. The following examples provide a brief introduction to JavaScript and how it can be used to call other functions.

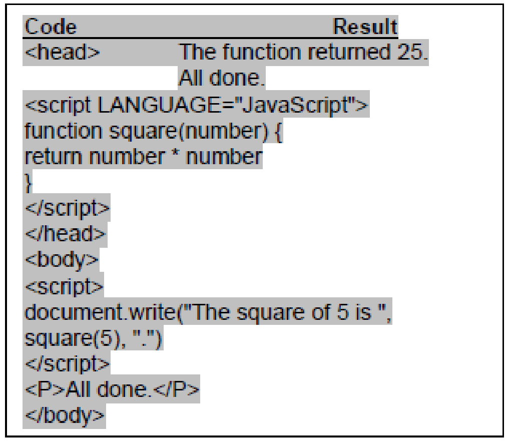


```
CODE                                   Result
<html>                                 x=4
<body>
<script type="text/javascript">
var x = 2 * 2
document.write("x = ", x)
</script>
</body>
</html>
```

Figure 14.5 shows a calculation performed with JavaScript within the body of an HTML file.

Functions are the fundamental building blocks of JavaScript. A function is a procedure—a set of statements that performs a specific task. Functions are generally defined in the head section of an HTML document. This assures that all functions are defined before any content is displayed. The example in Figure 15.6 defines a simple function in the head section of an HTML document. The function is then called in the body of the document.

The function **square** takes one argument, called *number*, and the function consists of one statement, **return** `number * number`,which indicates it should return the argument of the function multiplied by itself. The return statement specifies the value that is returned by the function.

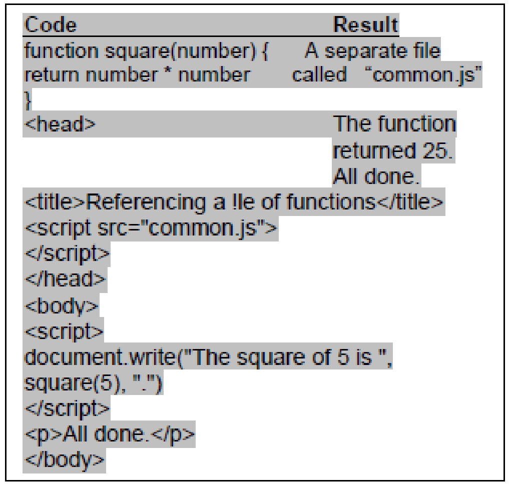


```
Code                                   Result
<head>              The function returned 25.
                    All done.
<script LANGUAGE="JavaScript">
function square(number) {
return number * number
}
</script>
</head>
<body>
<script>
document.write("The square of 5 is ",
square(5), ".")
</script>
<P>All done.</P>
</body>
```

Figure 14.6 shows a function that squares the number passed to it by a call to the function.

The external file that contains JavaScript functions can be on the same computer as the HTML file, as implemented in Figure 14.7, or it can be on another computer or server. This is how Application Programming Interface (API) code is distributed. One reference to a library of API code makes it possible for a webpage designer to access thousands of functions.

Rather than embed the JavaScript code directly in the HTML file, either in the body or the head, it is possible to place the JavaScript functions within a separate file. The SRC attribute of the **<script>** tag specifies the external file where the JavaScript code can be found. Figure 14.8 shows the external file called *common.js* and how it is referenced in the head section of an HTML document. The external JavaScript file may contain multiple functions but no HTML code.



```
Code                                   Result
function square(number) {      A separate file
return number * number        called  "common.js"
}
<head>                                 The function
                                       returned 25.
                                       All done.
<title>Referencing a !le of functions</title>
<script src="common.js">
</script>
</head>
<body>
<script>
document.write("The square of 5 is ",
square(5), ".")
</script>
<p>All done.</p>
</body>
```

Figure 14.7 shows a function that is placed into an external document, common.js. The function is then referenced from the HTML file with <script src=“common.js”>

Figure 14.8 shows how the Google Maps API code is referenced. To aid in debugging, the Google Maps API code will work locally on a computer without the need to transfer the code to a server. But, in order for others to see the map, the code must reside on a server.

```
Code
<head>
<title>Google Maps JavaScript API
Example</title>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sen
sor=false">
</script>
</head>
```

Figure 14.8 shows the implementation of a call to the Google Maps API that accesses a large number of mapping-related functions. Here, the sensor is set to "false." The value of the sensor would be set to "true" if a mobile device is used that can provide the current position.

14.3 Google Maps API

Introduced soon after Google Maps in 2005, the Google Maps Application Programming Interface consists of a series of functions that control the appearance of the map, including its scale and location, and any added information in the form of points, lines or areas and associated descriptions. The use of Google Maps API is essentially free, provided the site does not charge for access. Google places a limitation on the number of maps that can be served. A site cannot generate more than 25,000 map loads a day for 90 consecutive days. A map load is one map displayed with the Google Maps API. Once loaded, the degree to which a user interacts with a map has no impact on the map load number. It would be extremely difficult for the average user of the Google Maps API to exceed 25,000 map loads. Even if a site were to go "viral," it would need to sustain 25,000 map loads per day for 90 consecutive days before the limit would be reached. Usage limits can be placed on a site so that it does not exceed that number. If the site consistently exceeds 25,000 maps a day, Google would require you to register your site and pay US$0.50 for every 1,000 map views beyond this limit. For example, if your Google mapping page served 100,000 maps a day for 90 consecutive days, you would be charged $37.50 (75,000÷1,000 x 0.5) a month after the initial 90-day period.

Specialized Google Maps API web services have additional usage limits, including:

- Directions—provides directions in text form—limited to 2,500 a day;
- Distance Matrix—returns travel distance and time—limited to 100 elements per query and 2,500 a day;
- Elevation—elevation at points—limited to 2,500 requests per day where each request returns up to 512 elevations;
- Geocoding—converts a street address to latitude and longitude—limited to 2,500 a day;
- Places—returns business establishments and other points of interest around a point—requires an API key and is limited to 1,000 requests a day.

A Google Maps API key is a numeric code that registers your site with Google. It is not needed for normal applications and would only be required if the usage limits are exceeded or the Places web service is used.

The example in Figure 14.9 shows the JavaScript code and API calls for displaying a simple map that is centered at a specific location. The zoom level, which can range from 0 to 21, is set to 15 under `myOptions`. The center is defined with a specific latitude and longitude value, and the ROADMAP option is selected to define the map style. All of the API calls are made in the **initialize** function. This function is called **onload** within the body of the HTML file.

```
<html>
<head>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
function initialize(){
var latlng = new google.maps.LatLng(46.810928, -
90.817981);
var myOptions = {
zoom: 15,
center: latlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map=new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
}
</script>
<title></title>
</head>
<body onload="initialize()">
<div id="map_canvas" style="width:600px; height:300px">
</div>
</body>
</html>
```

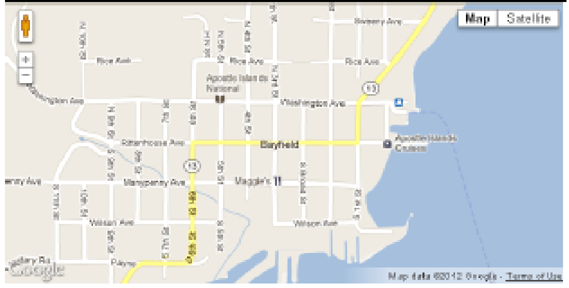


Figure 14.9 shows a basic Google map request. The center of the map can be changed along with the zoom level and map type. (© 2014 Google)

A simple change can be made to this code by substituting new latitude and longitude values. Determining the latitude/longitude for a specific point can be done in a number of different ways:

- In Google Maps with a right-click (control + click on a Mac) and selecting "What's here?" from the longitude of the point will appear in the top line of the Google Maps window;

- A right-click with MapQuest displays the values in a pop-up window;
- In Bing Maps, the latitude and longitude are displayed with a right-click;
- To display the coordinates in the decimal degrees format with Google Earth, select Tools/Options and click on the decimal degrees option;
- Finally, there are a number of online utilities. Searching for "Finding latitude and longitude" will lead you to a site. Most of these sites use Google Maps, including this example: **http://findlatituteandlongitude.com.**

Another change to the basic Google Map is the type or style of map that is displayed. Google offers four views:

- **MapTypeId.ROADMAP** displays the default road map view;
- **MapTypeId.SATELLITE** displays Google Earth satellite images;
- **MapTypeId.HYBRID** displays a mixture of normal and satellite views;
- **MapTypeId.TERRAIN** displays a physical map based on terrain information.

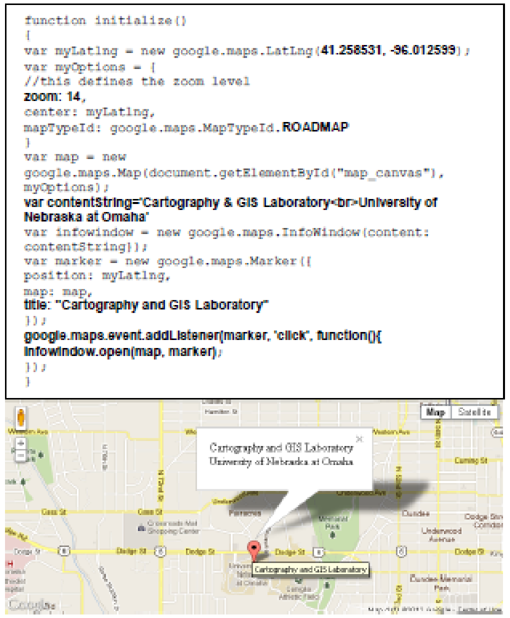
The initial zoom level can also be changed. A value of "0" would draw a zoomed-out, small-scale map. As the zoom level number increases, so too does the scale of the map. The upper value varies for different parts of the world. Generally, 20 levels of details are always available. Some parts of the world have more than 20 zoom levels.

14.4 Point, Line, Area, and Layer Mash-ups

All maps are composed of points, lines and areas. In addition, multiple maps can be combined as individual layers, a function that is the basis of geographic information systems. The examples in this section show how these elements can be added to the Google Map.

14.4.1 Points

The default Google marker is an upside-down raindrop symbol, but a large number of alternative icons are available. It is even possible to design symbols because they are simply 32x32 pixel images in the PNG format. Markers can be static or interactive. The major interactive type of marker is one that is clickable.



```
function initialize()
{
var myLatlng = new google.maps.LatLng(41.258531, -96.012599);
var myOptions = {
//this defines the zoom level
zoom: 14,
center: myLatlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
}
var map = new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
var contentString='Cartography & GIS Laboratory<br>University of
Nebraska at Omaha'
var infowindow = new google.maps.InfoWindow({content:
contentString});
var marker = new google.maps.Marker({
position: myLatlng,
map: map,
title: "Cartography and GIS Laboratory"
});
google.maps.event.addListener(marker, 'click', function(){
infowindow.open(map, marker);
});
}
```

Map
Satellite
Cartography and GIS Laboratory
University of Nebraska at Omaha
Cartography and GIS Laboratory
Dodge St
Cass St
Crossroads Mall Shopping Center
Memorial Park
Dundee
Underwood Avenue
Dundee Memorial Park
Cuming St
Hamilton St

*Figure 14.10 shows an example of a clickable marker. The **contentString** text variable is defined in HTML. (©2014 Google)*

In the example in Figure 14.10, the variable **contentString** is defined with text formatted in HTML. This is associated with an **infoWindow** variable that is subsequently associated with **google.maps.event.addListener**. When the user clicks on the marker, the text is displayed in a pop-up bubble. The HTML for this bubble could incorporate a picture or even a video using the **img** or **embed** tags.

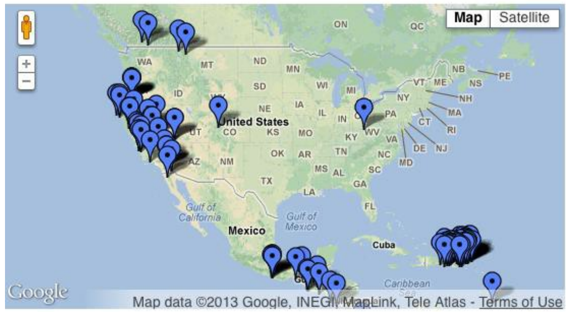
In cases where a large number of points need to be mapped that may be frequently updated, a web format called Really Simple Syndication (RSS) is often used. There are many advantages to RSS. Publishers of RSS feeds benefit by syndicating content automatically while consumers benefit through timely updates of mapped information. A standardized file format allows the information to be published once and viewed using many different programs.

KML, Keyhole Markup Language, is a format used for describing two- and three-dimensional space that was originally developed for Google Earth. It is now an open standard officially named the OpenGIS® KML Encoding Standard (OGC KML) and is maintained by the Open Geospatial Consortium (OGC). The format specifies features such as placemarks, images, polygons, and 3D models. Places are always specified with latitude and longitude. A large number of KML files are available through the Internet.

The **google.maps.KmlLayer** function reads a KML-formatted RSS feed specified with an HTTP address. Maps made in this way are usually displayed very fast. The trade-off is that there is less control over the look of the actual map because symbols are defined in the KML file.

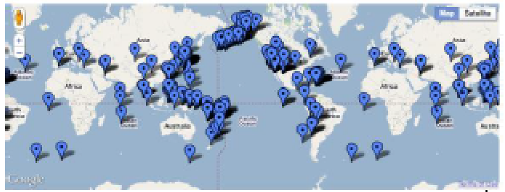
The example in Figure 14.11 shows an application of an RSS feed for the depiction of earthquakes. This particular KML feed is updated daily and shows earthquakes in the past seven days. Each marker is clickable and provides more information about the earthquake event. The code shows how to make a map for just a certain part of the world.

Earthquakes in the past week



```
var georssLayer = new
google.maps.KmlLayer('http://earthqu
ake.usgs.gov/earthquakes/
catalogs/eqs7day-M2.5.xml');
```

Earthquakes in the past week



```
var ctaLayer = new var ctaLayer = new
google.maps.KmlLayer('http://earthqu
ake.usgs.gov/earthquakes/
catalogs/eqs7day-
M2.5.xml',{preserveViewport:true});
ctaLayer.setMap(map);
```

*Figure 14.11 displays an RSS feed defined in KML format from the United States Geological Survey. Each icon pinpoints the earthquake and, when clicked, describes the event. The bottom map is displayed using a basic call to a KML layer that ignores the defined center and zoom level and therefore duplicates most of the world. The top map is displayed with the **{preserveViewport: true}** option and ctaLayer that applies the user-defined center and zoom level. (© 2014 Google)*

14.4.2 Lines

The google maps Polyline function is used to draw lines with the Google Maps API. In Figure 14.12, the Polyline function connects points that have been previously defined. Options for controlling the appearance of the line include strokeColor, strokeOpacity, and strokeWeight. As always, an appropriate center and zoom level must also be defined. The center could be the midpoint of the line itself.

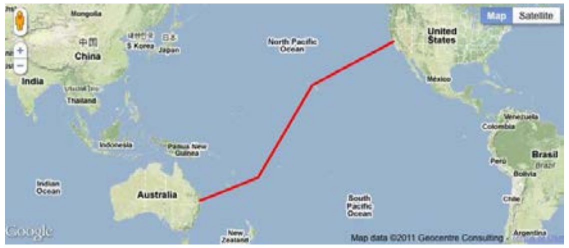


Figure 14.12 shows a line made from three elements consisting of four points. (© 2014 Google)

The shortest distance between two points on a map because the map has been projected from the spherical earth. On most projections, including the Mercator that is used by all major online mapping services, the shortest distance on the ground is represented as a curve—seemingly a longer distance between two locations. The great circle is defined as the shortest distance between two points that divides the Earth into two equal hemispheres. It is supported in the Google Maps API through the *geodesic: true* polyline option (see Figure 14.13).

```
var flightPlanCoordinates = [
new google.maps.LatLng(37.772323, -
122.214897),
new google.maps.LatLng(21.291982, -
157.821856),
new google.maps.LatLng(-18.142599, 178.431),
new google.maps.LatLng(-27.46758, 153.027892)
];
var flightPath = new google.maps.Polyline({
path: flightPlanCoordinates,
strokeColor: "#FF0000",
strokeOpacity: 1.0,
strokeWeight: 3
```

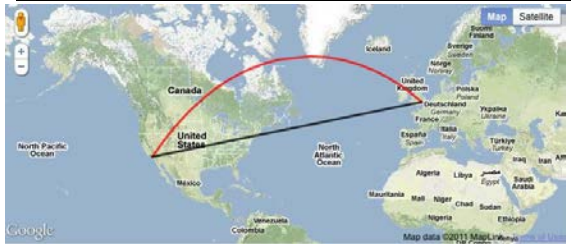


Figure 14.13 shows the *geodesic: true polyline option connecting two points through the great circle, the shortest distance between two locations on the sphere. It appears as a longer line on this map because of the projection. (© 2014 Google)*

14.4.3 Areas

A polygon may be viewed as a line that closes upon itself. It consists of a series of points, with the last point always equal to the first. The two additional attributes that need to be defined for **google.maps.Polygon** are the shading and opacity of the interior area.

```
<script type="text/javascript">
function initialise() {
var myLatLng=new
google.maps.LatLng(24.886436490787712,-
70.2685546875);
var myOptions = {
zoom: 5,
center: myLatLng,
mapTypeId: google.maps.MapTypeId.TERRAIN
};
var map = new
google.maps.Map(document.getElementById("map_c
anvas"),
myOptions);
var triangleCoords = [
new google.maps.LatLng(25.774252, -80.190262),
new google.maps.LatLng(18.466465, -66.118292),
new google.maps.LatLng(32.321384, -64.75737),
new google.maps.LatLng(25.774252, -80.190262)
];
var bermudaTriangle = new
google.maps.Polygon({
paths: triangleCoords,
strokeColor: "#FF0000",
strokeOpacity: 0.8,
strokeWeight: 2,
fillColor: "#FF0000",
fillOpacity: 0.35
});
bermudaTriangle.setMap(map);
}
</script>
```

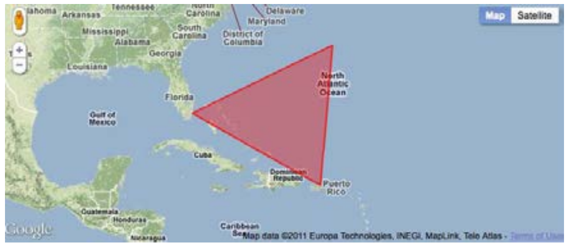


Figure 14.14 The Google Polygon function draws a closed shape. Options include strokeColor, strokeOpacity, strokeWeight, fillColor and fillOpacity. (© 2014 Google)

Figure 14.14 shows the Bermuda Triangle in the Atlantic Ocean. Four points are defined to depict the triangle. These points are loaded into an array named **triangleCoords**. This array is then passed to **google.maps.Polygon**. Parameters include the **strokeColor, strokeOpacity, stroke-Weight, fillColor** and **fillOpacity.**

14.4.4 Layers

To this point, we have overlaid points, lines, and areas that were defined as vectors of latitude and longitude. Now, we overlay a raster image or picture that could be an air photo, satellite image and scanned map. The advantage of overlaying an image is that the overlay can be done quickly. No major conversion or drawing is necessary to place the information because the underlying map is in the same format. Raster files can be overlaid as single entities or divided into tiles to precisely match the tiles of the underlying map.

The example in Figure 14.15 shows a map that has been scanned and saved in the JPEG format. The latitude and longitude of the southwest and northeast corners have been estimated and are then defined using **imageBounds**. These coordinates are combined with the address of the image in the **oldmap** object.

```
function initialise() {
var newark = new google.maps.LatLng (40.740, -
74.18);
var imageBounds = new
google.maps.LatLngBounds(
new google.maps.LatLng (40.712216,-74.22655),
new google.maps.LatLng (40.773941,-74.12544));
var myOptions = {
zoom: 12,
center: newark,
mapTypeId: google.maps.MapTypeId.ROADMAP
}
var map=new
google.maps.Map(document.getElementById("map_c
anvas"),myOptions);
var oldmap = new google.maps.GroundOverlay(
"http://www.lib.utexas.edu/maps/historical/newark_nj_192
2.jpg",
imageBounds);
oldmap.setMap(map);
}
```

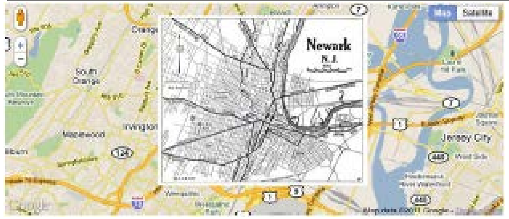


Figure 14.15 shows an overlay of a scanned map in the JPEG format. Map of Newark, NJ, courtesy of the University of Texas Libraries, The University of Texas at Austin. (© 2014 Google)

14.5 Mobile Mapping

Location-aware devices are becoming increasingly common. Virtually all mobile phones can now be located within a few meters. Smartphones have the additional ability to display their current location on a map. Tablet computers based on Apple's iOS and Google's Android can usually do the same but with the added benefit of displaying a much larger image.

There are many different types of mobile devices and many different ways of determining location. To provide a standardized approach, the World Wide Web Consortium (W3C) created a freely available geolocation API. Supported by nearly all browsers, the API uses multiple methods to find the location of the computer or mobile device (Svennerberg 2010, P. 235).

The Global Positioning System (GPS) is one method of determining location, but it only works with an unobstructed view of the sky. In urban areas, the most common method of determining location is triangulation based on Wi-Fi and cell tower signals. Location software developed by the Boston-based company, Skyhook, uses a massive reference network comprised of the known locations of over 250 million Wi-Fi access points and cellular towers. To develop the database, Skyhook deployed drivers to survey every single street, highway, and alley in tens of thousands of cities and towns worldwide, scanning for Wi-Fi access points and cell towers and plotting their precise geographic locations.

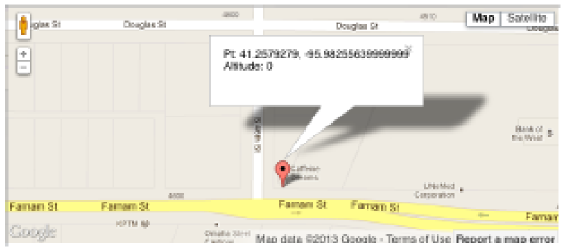
Mapping the current location of a device through a browser using the W3C API is shown in Figure 14.16. The

"navigator.geolocation.getCurrentPosition (function(position)"

statement resolves the current position of the device. If the position cannot be determined by GPS, the API uses the triangulation method based on a wireless network. This example presents an infowindow at the current location.

The example in Figure 14.16 replaces the infowindow with a clickable marker.

The contentString for the bubble text displays the latitude and longitude.



Pt: 41.2579279, -95.98255639999999
Altitude: 0

```
if(navigator.geolocation) {
browserSupportFlag = true;
navigator.geolocation.getCurrentPosition(funct
ion(position) {
initialLocation = new google.maps.LatLng
(position.coords.latitude,position.coords.longitude);
contentString = "Pt:
"+position.coords.latitude+",
"+position.coords.longitude ;
map.setCenter(initialLocation);
infowindow.setContent(contentString);
infowindow.setPosition(initialLocation);
var marker = new google.maps.Marker({
position: initialLocation,
map: map,
title:"Hello World!"
});
google.maps.event.addListener(marker, 'click',
function() {
infowindow.open(map,marker);
});
}, function() {
handleNoGeolocation(browserSupportFlag);
});
}
}
```

Figure 14.16 shows the text displayed. In the clickable bubble is the latitude and longitude of the current point of the mobile device. The "+" in the contentString statement is used to concatenate the numbers into a text string. (© 2014 Google)

14.6 Conclusions

We live in an amazing time for mapping. Within a matter of 20 years, from the 1970s to the 1990s, maps changed from static objects on paper to interactive presentations delivered through an electronic network. In the years since then, maps have become even more interactive in the sense of being able to add information—both thematic information through mash-ups and the editing of the underlying base map. The exercises in this chapter provided an introduction to the new world of mapping through the Internet. The tools that were introduced can be used to make very sophisticated types of maps.

References

Google Maps JavaScript API V3 Basics (2011). (search: Google Maps JavaScript API V3 Basics).

Neumann, A., Winter A. M. (2003). "Web-mapping with Scalable Vector Graphics (SVG): Delivering the promise of high quality and interactive web maps." In: Peterson, M. P. (ed.) *Maps and the Internet*. Elsevier, Amsterdam, pp.197–220.

Peterson MP (2008) *International Perspectives on Maps and the Internet.* Springer, Berlin.

Svennerberg, Gabriel (2010). *Beginning Google Maps API 3.* New York, NY: Apress.

W3Schools.com (2011). JavaScript Tutorial. [http://www.w3schools.com/js/default.asp]. (search: Learning JavaScript).

Willard, Wendy (2009). *HTML: A Beginner's Guide*. Berkeley, CA: Osborne/McGraw-Hill.

Note: The material for this chapter is based on the author's book entitled *Mapping in the Cloud* and published by Guilford Press.