

14 La cartografia sul Web e sui dispositivi mobili

Michael P. Peterson, USA

Riassunto

Da sempre, per realizzare le carte geografiche e misurare la terra, la geografia dipende da un insieme di strumenti. Dal 2005, per creare cartografia, sono stati sviluppati nuovi strumenti, disponibili *online* o *cloud-based*, chiamati *Application Programmer Interfaces* (API). Tali strumenti hanno il vantaggio di essere automaticamente disponibili a chiunque abbia una connessione internet.

In questo capitolo si esamina la nuova era dei *mash-up* nella cartografia e si spiega come realizzare e distribuire carte geografiche usando strumenti cartografici gratuiti.

14.1 Introduzione

È difficile esagerare quando si parla dell'importanza delle carte geografiche come forma di comunicazione a proposito della terra. Le carte ci aiutano a capire ciò che ci circonda, ma anche lo spazio che si trova oltre la nostra diretta percezione; influenzano ciò che pensiamo del mondo e il nostro modo di comportarci; ci collegano al nostro ambiente.

Ognuno di noi è un cartografo, nel senso che tutti noi facciamo mappe mentali. Qualche volta, abbiamo bisogno di disegnare ad altri queste carte perché ci aiutino a spiegare come trovare una certa località. La realizzazione delle carte e l'analisi delle informazioni sottostanti si sono evolute in una scienza e sono conoscenze di valore per molti e differenti tipi di lavoro.

Molti di questi strumenti per la cartografia sono oggi presenti nel *cloud* - un sofisticato sistema di *hardware* e



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. .hostei.com

Your name

Your email (account details will be sent there)

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

Type password again

14.1a

Figura 14.1a - La pagina di registrazione a <http://000webhost.com>. Le informazioni sull'account sono mostrate a destra (figura 14.1b). La scelta di un sottodominio è gratuita. Un costo è dovuto se si desidera un dominio con un nome particolare, come <http://www.peterson.com>, ma è gratuito lasciare che il sito scelga un dominio libero, come <http://geographyprof.hostei.com>. (© 2014 First Class Web Hosting) - Figura 14.1b - Informazioni sull'account.

software accessibile via internet. Tali strumenti consentono una forma molto avanzata di cartografia.

Un ulteriore vantaggio di queste carte *online* è che possono facilmente essere messe a disposizione di altri utenti.

Fare cartografia nel *cloud* richiede l'uso di un *server*. Anche se è possibile trasformare quasi ogni computer in un *server*, è più semplice e più sicuro utilizzare un servizio di *hosting online*. Servizi di *cloud-hosting*, come Amazon Web Services e Microsoft Azure, implementano la scalabilità, nel senso che possono essere potenziati

» 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

14.1b

Un'altra opzione a disposizione, quando si cerca spazio dove ospitare pagine web, sono i servizi di *web-hosting* che fanno pagare solo per la memoria del disco. Questi servizi non sono scalabili, ma sono più facili da usare e offrono la possibilità di avere gratuitamente dei sottodomini e fino a 1.500 MB di spazio su disco.

Due di questi servizi di *web-hosting* gratuiti sono *000webhost.com* e *podserver.info*. I paragrafi che seguono descrivono come creare un sito web e fornire cartografia *online* usando uno di questi servizi gratuiti di *web-hosting*.

14.2 I server nel cloud

14.2.1 Fare spazio nel cloud

La figura 14.1 illustra la pagina di registrazione del *provider* 000webhost.com e le informazioni relative all'*account*.

Si noti il sottodominio gratuito *hostei.com*.

Un indirizzo come <http://geographyprof.com> si trova ad un livello superiore e potrebbe avere dei costi in quanto rappresenta un nuovo dominio. Per ottenere un *account* gratuito è importante non registrare il proprio dominio.

Per la registrazione di un sottodominio gratuito, è necessario un indirizzo di posta elettronica.

La pagina con le informazioni sull'*account* mostra che l'indirizzo web assegnato dal *provider* è

<http://geographyprof.hostei.com>

(o <http://64.120.177.162>), e che vengono resi disponibili 1.500 MB di spazio gratuito. Le informazioni mostrano anche la disponibilità del web *server* Apache e di altri strumenti accessibili *online*, inclusi PHP e MySQL.

Un'interfaccia grafica dei servizi offerti è inclusa dalla maggior parte dei *provider* di *web-hosting* ed è chiamata "pannello di controllo" o cPanel (figura 14.2). Gli strumenti gestiscono la posta, la modifica dei *file*, la programmazione delle azioni e la gestione dell'*account*. Tutti questi strumenti rappresentano progetti *open-source*, che sono stati scritti e aggiornati da una piccola legione di programmatori.

Il *File Manager* è lo strumento più utile per la gestione dei *file* e per la costruzione di un sito. MySQL e phpMyAdmin vengono usati per costruire un *database*. Per accedere alle risorse del *server*, la maggior parte dei servizi di *web-hosting* usano dei cPanel abbastanza simili fra loro.

La figura 14.3 mostra la finestra di un *File Manager*, con l'accesso agli strumenti per caricare e creare nuovi *file* e *directory* (sottocartelle), oppure per spostare, cancellare e



Figura 14.2 - Un pannello di controllo standard di *web-hosting*, chiamato cPanel, che dà accesso a differenti strumenti. Il *File Manager* è il programma più importante per caricare ed editare i *file* (© 2014 First Class Web Hosting).

rinominare i *file*. L'elenco dei *file* riporta il nome, il tipo e le dimensioni, mentre i campi Proprietario (*Owner*), Gruppo (*Group*) e Permessi (*Perms*) rappresentano le specifiche di sicurezza. *Mod Time* indica quando è avvenuta l'ultima modifica al *file*. Si può lavorare sui *file* direttamente da questa finestra cliccando "Edit" alla fine di ogni nome.

La cartella **public_html** è la *directory* nella quale sono contenuti tutti i *file*. Se un *file html* deve essere visualizzato



Figura 14.3 - La finestra di un *File Manager* da un servizio di *hosting online*. Tale servizio permette di creare ed editare i *file*. Tutti i *file* a disposizione sul web devono trovarsi nella *directory/cartella public_html*. (© 2014 First Class Web Hosting)

Su internet deve risiedere in questa cartella che, generalmente, contiene un *file* chiamato *index.html* (o *index.php*) che rappresenta la pagina principale d'accesso al sito web.

Per esempio, se un indirizzo come <http://geographyprof.hostei.com/CloudMapping/> viene immesso in un *browser*, questo cercherà un *file* chiamato *index.html* in una *directory* (cartella) chiamata *CloudMapping*, anch'essa localizzata all'interno della *directory public_html*. Ciò significa che gli indirizzi seguenti visualizzeranno lo stesso *file*:
<http://geographyprof.hostei.com/OnlineMapping/>
<http://geographyprof.hostei.com/OnlineMapping/index.html>

Generalmente, il *file index.html* è un punto d'ingresso al sito web e avrà collegamenti con tutti gli altri *file* della *directory*.

Questo *file* avrà una struttura relativamente semplice: una riga con un titolo, seguita da *link* verso tutte le altre possibili operazioni. Inoltre, potrà avere, per esempio, una foto del gestore del sito, come si vede nella figura 14.4 (alla pagina seguente).

A seguire, il codice mostra come viene inserita un'immagine, usando il *tag*

``

I link alle pagine degli studenti sono separati da due linee verticali ("||"). Il codice per l'intero *file index.html* si può ottenere cercando "Peterson Mapping in the Cloud."

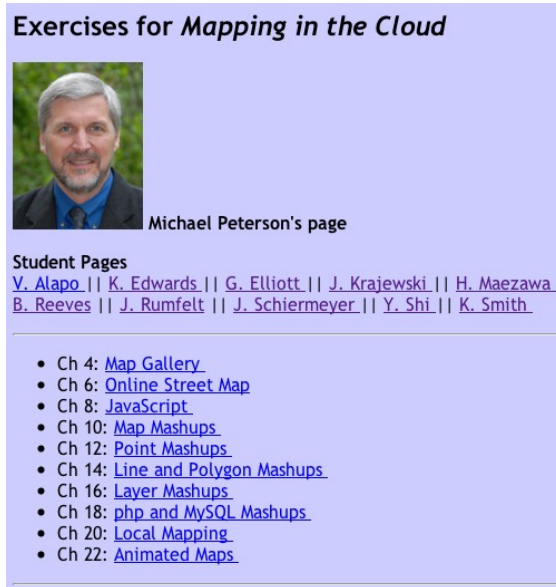


Figura 14.4 - Esempio di file *index.htm*, che include un'immagine, link a tutti gli studenti e ai compiti. Il codice mostra come visualizzare un'immagine usando il *tag img*, i link alle pagine degli altri studenti e ai compiti.

14.2.2 HTML

L' *html* è il mattone da costruzione del web, il linguaggio che rende possibile presentare le informazioni per mezzo di pagine web. È anche un contenitore per linguaggi di elaborazione, come JavaScript e PHP.

```
<html>
<head></head>
<body bgcolor="#CCCCCC">
<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> <br>
<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>
</li>
</ul>
<hr>
</body>
</html>
```

La conoscenza dell' *html* è necessaria per poter presentare la cartografia su internet. L'HTML consiste di *tag* che definiscono la struttura della pagina (Willard 2009).

Comprende un semplice codice testuale, racchiuso fra "<" e ">", che specificano come il documento apparirà nel *browser*, i *link* ai documenti o le parti grafiche. Per accedere ad un file *html* può essere usato un normale *editor* di testo (Notepad su Windows o TextEdit su Macintosh, con opportune configurazioni). Questi programmi, diversamente da quelli di videoscrittura, sono pensati per visualizzare testo non formattato. Una volta che i *file* vengono creati, possono essere aperti con un *browser*, come Explorer, Firefox o Chrome.

Tutti i file *html* iniziano con la *tag* "`<html>`" e terminano con lo stesso *tag* preceduto da una barra (es., "`</html>`") che indica che il codice *html* è finito. Dal punto di vista tecnico, tutti i *tag html* hanno un inizio e una fine e quest'ultima viene evidenziata con il *tag* preceduto da "/". Per esempio, il comando `<h1>`, che è utilizzato per il testo - per i titoli vengono usati testi più grandi - e il codice `</h1>` che interrompe tale formato di testo.

Possiamo visualizzare una carta geografica con un *tag* ``; tutti i nomi dei *file* di immagini, che fanno riferimento a *img*, devono terminare con GIF, JPG (o JPEG) o PNG perché questi sono i comuni tipi di *file* utilizzati dai diversi *browser*.

L'opzione "*src*" di *img* viene usata per specificare l'indirizzo (URL) del file. Nell'esempio in basso, si noti l'estensione del *file* dell'immagine ".gif" alla fine dell'URL. Il *tag img* include anche un certo numero di opzioni che possono essere usate per cambiare le dimensioni dell'immagine o modificare la sua posizione nella pagina. Come *hr*, anche *img* non ha un *tag* di chiusura. In una formattazione stretta XHTML, si scrive

```
<img src=map.png />
```

Il tag `<embed>` viene usato per visualizzare *file* di tipo grafico che non hanno un formato GIF, JPEG, o IMG, come, per esempio, i file PDF di Adobe, Flash, SVG, e QuickTime. Il formato è identico a quello dell'opzione di ``:

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

Uno dei tag più usati in *html* è l'ancora, generalmente utilizzata con le stringhe di testo, per creare un *link* di ipertesto. Può essere usata anche con il tag `img` per assegnare un *link* ad una immagine. Il formato del comando per il *link* è:

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

14.2.3 JavaScript

Per se stesso, l' *html* è semplicemente un linguaggio usato per formattare le pagine. In combinazione con *JavaScript*, una pagina *html* può eseguire un codice di comando (W3Schools.com 2011).

Una volta relegata al mondo degli smanettoni, oggi la programmazione viene vista come una forma di espressione, come "un'amplificazione del pensiero" e una conoscenza necessaria. Il concetto di "alfabetizzazione informatica" porta idee nuove a proposito dell'importanza del saper programmare.

I siti di programmazione che si trovano *online* come CodeAcademy diventano sempre più popolari; Khan Academy ha aggiunto una sezione di esercizi di programmazione *online* gratuiti.

Gli esempi che seguono forniscono una breve introduzione a *JavaScript* e mostrano come questo possa essere utilizzato per chiamare altre funzioni.

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

Figura 14.5 - Una operazione ottenuta con JavaScript all'interno del corpo del file *html*.

Le funzioni sono i mattoni fondamentali di *JavaScript*. Una funzione è una procedura, un insieme di dichiarazioni che portano a termine un compito specifico, e sono generalmente definite nella sezione di testa di un documento *html* (figura 14.6). Ciò assicura che tutte le funzioni siano definite, prima che i contenuti vengano visualizzati, per poi essere richiamate nel corpo del documento.

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

Figura 14.6 - Una volta richiamata, la funzione fa il quadrato del numero che le viene passato.

La funzione **square** prende un argomento, chiamato, *number*, e la dichiarazione, **return** number * number, indica che viene restituito l'argomento della funzione moltiplicato per sé stesso. Il risultato specifica il valore che è reso dalla funzione.

Il *file* esterno, che contiene le funzioni *JavaScript*, può stare sullo stesso computer, così come il *file html* (figura 14.7), oppure risiedere su un altro *computer* o *server*.

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

Figura 14.7 - Una funzione all'interno di un documento esterno, *common.js*. La funzione viene poi richiamata dal file *html* con `<script src="common.js">`

Questo è il modo in cui il codice delle *Application Programming Interface* (API) viene distribuito.

Un riferimento a una libreria di codice API consente, ad un *webdesigner*, di accedere a migliaia di funzioni.

Piuttosto che includere il codice *JavaScript* direttamente nel file *html* (nell'intestazione e/o nel corpo del documento), è possibile scrivere la funzione *JavaScript* in un file separato. L'attributo `src` del tag `<script>` specifica il file esterno dove può essere recuperato il codice *JavaScript*.

La figura 14.8 mostra la *file* esterno *common.js* e come questo viene richiamato nella sezione di testa del documento *html*. Il *file JavaScript* esterno, può contenere funzioni multiple ma non codice *html*.

```
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>
```

Figura 14.8 - L'implementazione di una chiamata alle API di Google Maps, che permettono l'accesso a un gran numero di funzioni cartografiche. In questo caso il sensore è su "falso". Se il dispositivo mobile utilizzato può fornire la posizione corrente il valore del sensore dovrebbe indicare "vero".

La figura 14.8 mostra come viene richiamato il codice API di Google Maps. Per la decodifica, il codice API di Google Maps lavorerà sul *computer*, in locale, senza la necessità di trasferirlo su un *server*. Tuttavia, affinché la cartografia sia visibile anche ad altri, il codice dovrà risiedere su un *server*.

14.3 Le API di Google Maps

Introdotta nel 2005, subito dopo Google Maps, le Google Maps Application Programming Interface consistono di una serie di funzioni che controllano l'aspetto della cartografia, incluse la scala, la localizzazione e ogni altra informazione aggiunta, nella forma di punti, linee o aree, con le relative descrizioni associate.

L'utilizzo delle API di Google Maps è gratuito e i *provider* non fanno pagare per l'accesso. Google pone un limite al numero di carte che possono essere distribuite; un sito non può generare più di 25.000 carte caricate al giorno, per 90 giorni consecutivi. Per carta caricata si intende la visualizzazione con le API di Google Maps.

Il livello di interazione dell'utente non ha un impatto sul numero di caricamenti. È estremamente difficile che un utente medio delle API di Google Maps possa superare il limite di 25.000 carte caricate e, anche nel caso che un sito diventi "virale", la cosa dovrebbe ripetersi per ben 90 giorni consecutivi. Limitazioni all'uso possono essere legate ad un sito specifico, in modo che questo non superi il limite; se, invece, si superano abbondantemente le 25.000 carte al giorno, Google potrebbe chiedere la registrazione del sito e il pagamento di 0.50 US\$ ogni 1.000 carte viste oltre il limite. Per esempio, se la nostra pagina di cartografia con Google Maps fornisce 100.000 carte al giorno per 90 giorni consecutivi, dovremo pagare 37.50\$ (75,000÷1,000 x 0.5) al mese, dopo il periodo iniziale di 90 giorni.

Servizi web specializzati con le API di Google Maps hanno limiti addizionali, che includono:

- le direzioni - fornire le direzioni in forma testuale - nel limite di 2.500 volte al giorno;
- la matrice delle distanze - fornisce la distanza di viaggio e il tempo di percorrenza - limitata a 100 elementi a interrogazione e 2.500 volte al giorno;
- l'altitudine - elevazione dei punti - limitata a 2.500 richieste al giorno, dove ogni richiesta restituisce fino a 512 quote;
- il *geocoding* - converte un indirizzo stradale in latitudine e longitudine - limitato a 2.500 volte al giorno;
- i luoghi - restituisce gli stabilimenti industriali e altri luoghi d'interesse intorno ad un punto - richiede una chiave API ed è limitato a 1.000 richieste al giorno.

Una chiave API di Google Maps è un codice numerico che registra un sito su Google. Non è necessaria per un uso ordinario e verrebbe richiesta solo se i limiti d'uso venissero superati o se fosse utilizzato il servizio *web Places*.

L'esempio della figura 14.9 mostra il codice *JavaScript* e la chiamata API per visualizzare una semplice carta che è centrata su una specifica posizione. Il livello di ingrandimento, che varia da 0 a 21, è posizionato su 15, al di sotto di *myOptions*. Il centro viene definito da uno specifico valore di latitudine e longitudine e l'opzione *ROADMAP* è selezionata per definire lo stile della carta. Tutte le richieste delle API avvengono nella funzione *initialize*, che è richiamata con *onload* all'interno del corpo del *file html*.

```
<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>
```

Figura 14.9 - Una richiesta di base di Google Maps. Il centro della carta può essere cambiato, così come il livello degli ingrandimenti e il tipo di cartografia. (© 2014 Google)

Un semplice cambio a questo codice può essere fatto sostituendo nuovi valori di latitudine e longitudine. Determinare la latitudine e longitudine di uno specifico punto può essere fatto in un certo numero di modi:

- clic (con il tasto destro del mouse) in Google Maps (control + clic su un Mac), selezionare "Che cosa c'è qui?" e le coordinate appariranno nella finestra *pop-up* di Google Maps;
- un clic (tasto destro) in MapQuest fa comparire i valori in una finestra *pop-up*;
- in Bing Maps la latitudine e la longitudine vengono visualizzate con un clic (tasto destro);
- per visualizzare le coordinate nel formato di gradi decimali con Google Earth, selezionare *Tools/Options* e cliccare sull'opzione relativa;
- infine, ci sono un certo numero di utilità *online*. Cercando "Finding latitude and longitude" si verrà reindirizzati verso un sito specializzato, che usa Google Maps. Es.: <http://findlatitudeandlongitude.com>

Un altro tipo di modifica che si può fare alla versione base di Google Maps è la scelta del tipo o dello stile della cartografia che viene visualizzata.

Google offre quattro viste:

- **MapTypeId.ROADMAP** visualizza la carta stradale di default;
- **MapTypeId.SATELLITE** visualizza le immagini satellitari di Google Earth;
- **MapTypeId.HYBRID** mostra un misto fra la vista normale e quella satellitare;

- **MapTypeId.TERRAIN** visualizza una carta fisica basata sulle informazioni del terreno.

Anche il livello di ingrandimento iniziale può essere cambiato. Un valore di "0" restituirà una carta a piccola scala. Man mano che aumenta il livello di ingrandimento aumenta anche la scala della carta. Il valore più alto è variabile nelle diverse parti del mondo. In genere, sono disponibili venti livelli di dettaglio, tranne che in alcune parti del mondo.

14.4 Mash-up di punti, linee, aree e livelli

Tutte le carte sono composte da punti, linee e aree. In aggiunta, più carte possono essere combinate insieme come singoli livelli, una funzione che è alla base del sistema informativo geografico. Gli esempi di questo paragrafo mostrano come questi elementi possono essere aggiunti a Google Maps.

14.4.1 I punti

Il segnapunti (*marker*) predefinito di Google è un simbolo a forma di goccia capovolta, ma sono disponibili un gran numero di icone alternative. È anche possibile disegnare i propri simboli perché sono semplici immagini di 32x32 pixel in formato PNG. I *markers* possono essere statici o interattivi; di quest'ultimo tipo, il più importante è quello cliccabile.

Nell'esempio della figura 14.10, la variabile **contentString** è definita con un testo formattato in *html*, che è legato ad una variabile **infoWindow**, che a sua volta è associata a **google.maps.event.addListener**. Quando l'utente clicca sul *marker*, il testo viene visualizzato in una finestra *pop-up*. L'*html* per questo *pop-up* può includere un'immagine o anche un video, utilizzando i tag **img** e **embed**.

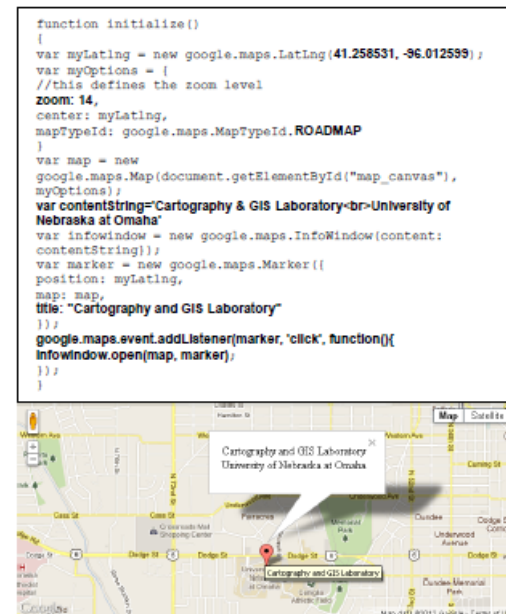


Figura 14.10 - Esempio di marker cliccabile. La variabile testuale **contentString** è definita in *html* (©2014 Google).

Nei casi in cui devono essere cartografati e frequentemente aggiornati un gran numero di punti, viene spesso utilizzato un formato web, chiamato Really Simple Syndication (RSS), che presenta molti vantaggi.

Chi pubblica sui canali RSS trae beneficio dall'associazione automatica dei contenuti mentre, per i consumatori, c'è il vantaggio dell'aggiornamento nel tempo delle informazioni cartografate. Un formato standard per il *file*, permette di pubblicare le informazioni una sola volta ma di essere visibili usando diversi programmi.

KML, Keyhole Markup Language, originariamente sviluppato per Google Earth, è un formato, utilizzato per descrivere lo spazio a due e a tre dimensioni.

Si tratta di uno standard aperto, chiamato ufficialmente OpenGIS® KML Encoding Standard (OGC KML) ed è aggiornato dall' Open Geospatial Consortium (OGC). Il formato specifica caratteristiche come *markers*, immagini, poligoni e modelli 3D. I luoghi vengono identificati sempre con latitudine e longitudine. Un gran numero di *file* KML sono disponibili su internet.

La funzione `google.maps.KmlLayer` legge un *feed* RSS, formattato come KML e identificato con un indirizzo HTTP. Le carte realizzate in questo modo, in genere, vengono visualizzate molto velocemente. Lo svantaggio consiste nel minore controllo sull'aspetto della carta, perché i simboli sono definiti nel file KML.

L'esempio della figura 14.12 mostra un'applicazione di un *feed* RSS per la rappresentazione dei terremoti. Questo particolare *feed* KML è aggiornato quotidianamente e mostra i terremoti nei sette giorni precedenti. Ogni *marker* è cliccabile e fornisce maggiori informazioni sul terremoto corrispondente. Il codice mostra come realizzare una carta per una certa parte del mondo.

14.4.2 Le linee

La funzione "`Polyline`" di Google Maps è usata per tracciare linee con le API di Google Maps. Nella figura 14.11, la funzione collega punti che sono stati definiti in precedenza.



Figura 14.11 - Una linea formata da tre elementi, che consistono di quattro punti (© 2014 Google).

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);
```

Figura 14.12 - Un *feed* RSS definito in formato KML, dello United States Geological Survey. Ogni icona localizza un terremoto e descrive l'evento quando cliccata. La carta in basso è visualizzata usando una normale chiamata a un livello KML, che ignora il centro predefinito e il livello di ingrandimento e pertanto duplica gran parte del mondo. La carta in alto è visualizzata con l'opzione `{preserveViewport: true}` e il `ctaLayer` che applica il centro e il livello di ingrandimento definiti dall'utente (© 2014 Google).

Le opzioni per controllare l'aspetto della linea sono `strokeColor`, `strokeOpacity`, e `strokeWeight`. Come sempre, devono essere definiti anche un centro e un livello di ingrandimento appropriati. Il centro dovrebbe essere il punto centrale della linea stessa.

La linea geodetica è distanza più breve tra due punti su una carta, perché questa è stata proiettata dalla sfera terrestre. Su gran parte delle proiezioni, compresa quella di Mercatore, la distanza più corta fra due punti sulla terra è rappresentata con una curva, anche se, in apparenza, sembra più lunga. Il cerchio massimo è definito come la distanza più corta fra due punti che dividono la terra in due emisferi uguali. È supportata dalle API di Google Maps per mezzo dell'opzione della polilinea "`geodesic: true`" (figura 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
```



Figura 14.13 - L'opzione della polilinea "`geodesic: true`", che unisce due punti sul cerchio massimo, è la distanza più breve fra due punti sulla sfera. A causa della proiezione, appare come una linea più lunga (© 2014 Google).

14.4.3 Le aree

Un poligono può essere visto come una linea che si chiude su se stessa. Consiste di una serie di punti, con l'ultimo punto sempre uguale al primo. I due attributi aggiuntivi da definire per **google.maps.Polygon** sono la sfumatura e l'opacità dell'area interna.

La figura 14.14 mostra il Triangolo delle Bermuda, nell'Oceano Atlantico. Per raffigurare il triangolo vengono fissati quattro punti che poi sono caricati in una matrice chiamata **triangleCoords** che, a sua volta, viene passata a **google.maps.Polygon**. I parametri inclusi sono **strokeColor**, **strokeOpacity**, **strokeWeight**, **fillColor** e **fillOpacity**.

14.4.4 I livelli

Finora abbiamo sovrapposto punti, linee e aree supposti quali vettori di latitudine e longitudine. Ora, sovrapponiamo un'immagine raster o una foto, che potrebbe essere una foto aerea, un'immagine satellitare o una carta scansionata. Il vantaggio della sovrapposizione di un'immagine è che può essere fatta rapidamente; non è richiesta alcuna conversione o disegno per inserire le informazioni, perché la carta sottostante è dello stesso formato.

I *file raster* possono essere sovrapposti come una singola entità o suddivisi in tessere di un mosaico che si abbinano perfettamente alla carta sottostante.

L'esempio in figura 14.15 mostra una carta che è stata scansionata e salvata in formato JPEG. La latitudine e la longitudine degli angoli a sudovest e a nordovest sono state prima stimate e poi definite usando **imageBounds**. Queste coordinate vengono associate all'indirizzo dell'immagine nell'oggetto **oldmap**.

```
<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>
```



Figura 14.14 - La funzione Google Polygon disegna una forma chiusa. Le opzioni comprendono **strokeColor**, **strokeOpacity**, **strokeWeight**, **fillColor** e **fillOpacity** (© 2014 Google).

```
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);
}
```



Figura 14.15 - La sovrapposizione di una carta scansionata in formato JPEG. Carta di Newark, NJ, cortesia delle Biblioteche dell'Università del Texas, Università del Texas, Austin (© 2014 Google).

14.5 La cartografia sui dispositivi mobili

I dispositivi che rilevano la posizione geografica stanno diventando sempre più comuni. Virtualmente, ogni telefono cellulare potrebbe essere localizzato nel raggio di pochi metri. Gli smartphone hanno l'ulteriore capacità di visualizzare la loro posizione corrente su una carta geografica. I tablet basati su iOS di Apple e Android di Google generalmente possono fare la stessa cosa, con l'opportunità di visualizzare un'immagine più ampia.

C'è una grande varietà di dispositivi mobili e molti e diversi modi di determinare la posizione. Per avere un approccio comune, il World Wide Web Consortium (W3C) ha creato un'API di geolocalizzazione che è disponibile gratuitamente. Tale API, supportata da quasi tutti i *browser*, usa vari metodi per trovare la posizione del *computer* o del dispositivo mobile (Svennerberg 2010, P. 235).

Il Global Positioning System (GPS), è un sistema usato per determinare la posizione, ma funziona solo all'aperto. Nelle aree urbane, il più comune metodo per determinare la posizione è la triangolazione basata sui ripetitori Wi-Fi e telefonici.

I *software* per la localizzazione, sviluppati dalla Skyhook, azienda che si trova a Boston, utilizzano una imponente rete di riferimento, costituita dalla posizione conosciuta di oltre 250 milioni di punti di accesso Wi-Fi e di ripetitori telefonici. Per sviluppare il *database*, Skyhook ha dispiegato i suoi operatori in ogni singola via, autostrada e vicolo, in decine di migliaia di città in tutto il mondo, tracciando la precisa posizione geografica dei punti di accesso Wi-Fi e dei ripetitori telefonici.

La figura 14.16 mostra la posizione di un punto su una carta rilevata per mezzo di un *browser*, usando le API del W3C. La dichiarazione

```
“navigator.geolocation.getCurrentPosition  
(function(position)”
```

determina la posizione corrente del dispositivo. Se tale posizione non può essere determinata con un GPS, le API usano un metodo di triangolazione basato su una rete *wireless*. Questo esempio presenta un *pop-up* della posizione corrente.



```
if(navigator.geolocation) {  
  browserSupportFlag = true;  
  navigator.geolocation.getCurrentPosition(function(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);  
  });  
}
```

Figura 14.16 - Codice API del testo visualizzato nella finestra cliccabile; vengono riportate la latitudine e la longitudine correnti del dispositivo mobile. Il “+” nella riga della dichiarazione *contentString* viene utilizzato per mostrare i numeri come una stringa di testo (© 2014 Google).

L'esempio di figura 14.16 sostituisce il *pop-up* con un *marker* cliccabile.

La *contentString* per questo testo mostra la latitudine e la longitudine.

14.6 Conclusioni

Viviamo tempi incredibili per la cartografia. Nel giro di 20 anni, dagli anni '70 ai '90, le carte geografiche sono cambiate da un oggetto statico su carta ad una rappresentazione interattiva distribuita per mezzo di una rete informatica.

Nel tempo, le carte sono diventate sempre più interattive, in quanto è stato possibile fornire sempre più informazioni tematiche per mezzo dei *dash-up*, e rendere possibile la modifica della carta di base.

Gli esempi di questo capitolo forniscono un'introduzione al nuovo mondo della cartografia, che può essere utilizzata per realizzare carte anche molto sofisticate.

Riferimenti

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.

Nota: il materiale per questo capitolo è tratto dal libro dell'autore, intitolato "*Mapping in the Cloud*", pubblicato da Guilford Press.