

Tecnologie Web: Leaflet

Raffaele Montella, PhD

raffaele.montella@uniparthenope.it

Sommario

- Introduzione
- Preparare la pagina
- Impostazione della mappa
- Marcatori
- Cerchi
- Poligoni
- Popup
- Eventi
- Conclusioni

Introduzione

- Leaflet è la principale libreria JavaScript open source per mappe interattive ottimizzate per dispositivi mobili.
- Con un peso di circa 38 KB di JS, ha tutte le funzionalità di mappatura di cui la maggior parte degli sviluppatori ha bisogno.
- È stata progettata pensando a semplicità, prestazioni e usabilità.

Introduzione

- Funziona in modo efficiente su tutte le principali piattaforme desktop e mobili.
- Può essere estesa con molti plug-in.
- Ha un'API facile da usare e ben documentata.
- Ha un codice sorgente semplice e leggibile.

Preparare la pagina

Prima di scrivere qualsiasi codice per la mappa, è necessario eseguire le seguenti fasi di preparazione sulla pagina:

- 1) Includere il file CSS di Leaflet nella sezione principale del documento:

```
<link rel="stylesheet"
href="https://unpkg.com/leaflet@1.5.1/dist/leaflet.css"
integrity="sha512-
xwE/Az9zrjBIphAcBb3F6JVqxf46+CDLwflMH1oNu6KEQCAWi6HcDUbeO
fBIptF7tcCzusKFjFw2yuvEpDL9wQ=="
crossorigin="" />
```

Preparare la pagina

2) Includere il file JavaScript del Leaflet dopo il CSS :

```
<!-- Make sure you put this AFTER Leaflet's CSS -->  
<script  
src="https://unpkg.com/leaflet@1.5.1/dist/leaflet.js"  
integrity="sha512-  
GffPMF3RvMeYyc1LWMHtK8EbPv0iNZ8/oTtHPx9/cc2ILxQ+u905qIwdp  
ULaqDkyBKgOaB57QTMg7ztg8Jm2Og=="  
crossorigin=""></script>
```

3) Inserire un elemento div con un determinato ID nel punto in cui si desidera posizionare la propria mappa:

```
<div id="mapid"></div>
```

Preparare la pagina

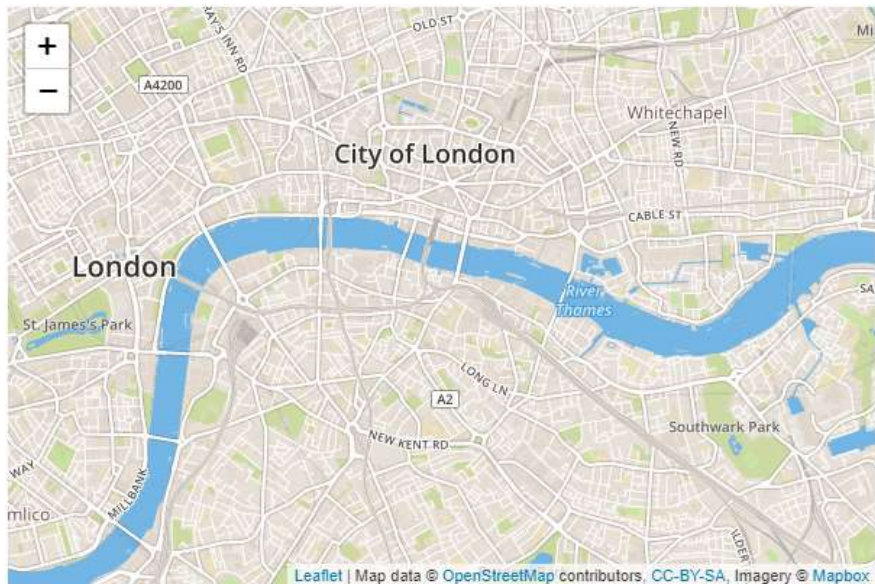
- 4) Assicurarsi che il contenitore della mappa abbia un'altezza definita, ad esempio impostandolo nel CSS:

```
#mapid { height: 180px; }
```

Ora si è pronti per inizializzare la mappa e operare con essa.

Impostazione della mappa

Creare una mappa del centro di Londra con tessere Mapbox Streets.



Innanzitutto è necessario inizializzare la mappa e impostare la vista sulle coordinate geografiche scelte e su un livello di zoom:

```
var mymap =  
L.map('mapid').setView([51  
.505, -0.09], 13);
```


Impostazione della mappa

- Tutte le interazioni del mouse e del tocco sulla mappa sono abilitate.
- La mappa ha i controlli di zoom e attribuzione.
- La chiamata ***setView*** restituisce anche l'oggetto *map*.

Impostazione della mappa

- La maggior parte dei metodi Leaflet agisce in questo modo quando non restituisce un valore esplicito.
- Questo consente un comodo concatenamento di metodi simile a jQuery.

Impostazione della mappa

- Creare un **layer** da aggiungere alla mappa, in questo caso si tratta di un livello di riquadro di tipo *Mapbox Streets*.
- La creazione di un layer comporta l'impostazione del template URL per i tile, il testo di attribuzione e il livello di zoom massimo del livello.
- In questo esempio sono utilizzate le tessere *mapbox.streets* dalle "Mappe classiche" di Mapbox (per utilizzare i tile da Mapbox, bisogna richiedere un token di accesso).

Impostazione della mappa

```
L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}
/{y}.png?access_token={accessToken}', {

  attribution: 'Map data &copy; <a
href="https://www.openstreetmap.org/">OpenStreetMap</a>
contributors, <a
href="https://creativecommons.org/licenses/by-
sa/2.0/">CC-BY-SA</a>, Imagery © <a
href="https://www.mapbox.com/">Mapbox</a>',

  maxZoom: 18,

  id: 'mapbox.streets',

  accessToken: 'your.mapbox.access.token'

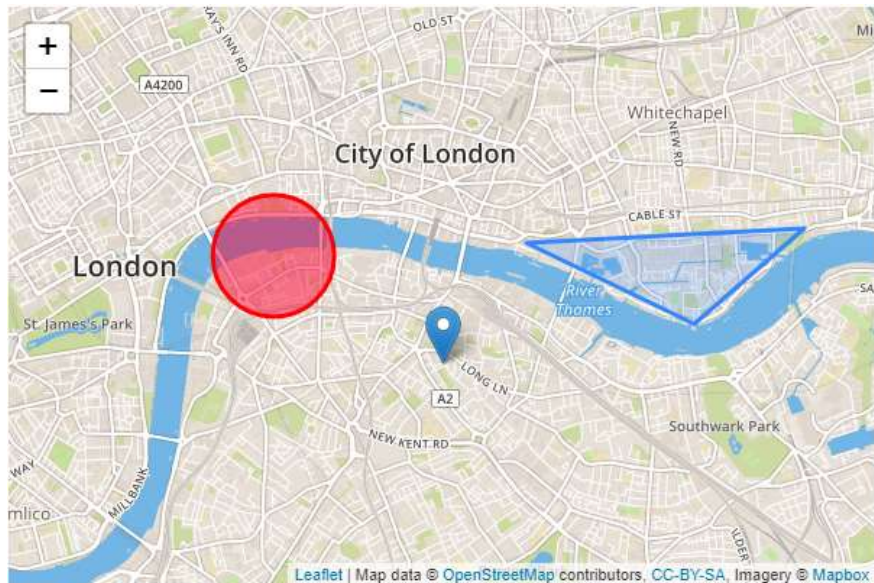
}).addTo(mymap);
```

Impostazione della mappa

- Assicurarsi che tutto il codice sia chiamato dopo l'inclusione di `div` e `leaflet.js`.
- Leaflet non contiene codice specifico di un singolo fornitore di dati cartografici.
- Si è liberi di utilizzare altri fornitori se necessario.
- Esempio:
provare a sostituire *mapbox.streets* con *mapbox.satellite*.

Marcatori

Oltre ai tiled layer, si possono aggiungere facilmente alla mappa **marcatori, polilinee, poligoni, cerchi e popup**.



Per aggiungere un **marcatore**:

```
var marker =  
L.marker([51.5, -  
0.09]).addTo(mymap);
```

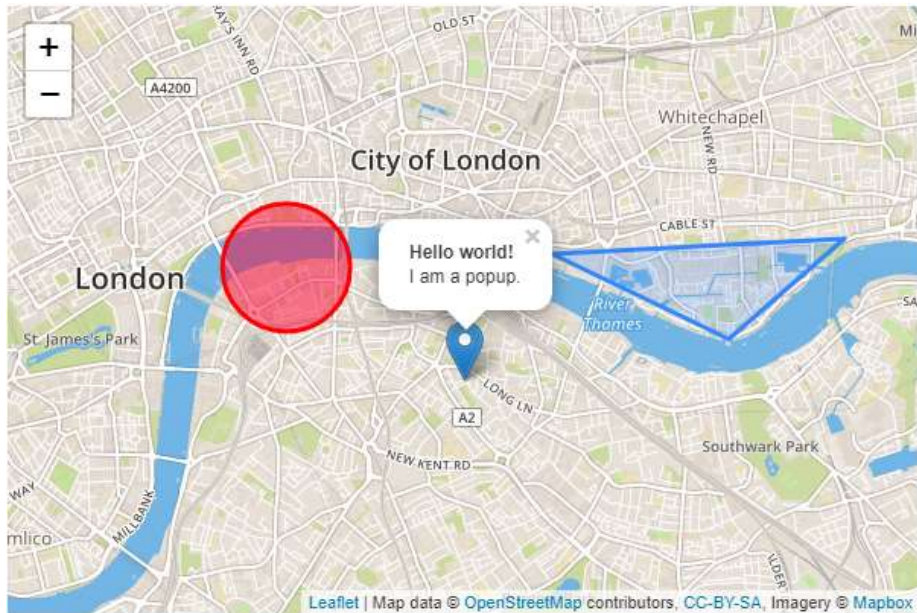
Cerchi

Per aggiungere un **cerchio**:

- Specificare il raggio in metri come secondo argomento.
- È possibile controllare l'aspetto passando le opzioni come ultimo argomento durante la creazione dell'oggetto.

```
var circle = L.circle([51.508, -0.11], {  
  color: 'red',  
  fillColor: '#f03',  
  fillOpacity: 0.5,  
  radius: 500  
}).addTo(mymap);
```

Poligoni e popup



Per aggiungere un **poligono**:

```
var polygon = L.polygon([  
    [51.509, -0.08],  
    [51.503, -0.06],  
    [51.51, -0.047]  
]) .addTo(mymap);
```

I **popup** vengono generalmente utilizzati quando si desidera allegare alcune informazioni a un particolare oggetto su una mappa.

Popup

Per aggiungere un **popup** Leaflet ha una scorciatoia molto utile:

```
marker.bindPopup("<b>Hello world!</b><br>I am a  
popup.") .openPopup();  
circle.bindPopup("I am a circle.");  
polygon.bindPopup("I am a polygon.");
```

- Il metodo ***bindPopup*** collega un popup con un contenuto HTML specificato a un marker, in modo che il popup appare quando si fa clic sull'oggetto.
- Il metodo ***openPopup*** (solo per i marker) apre immediatamente il popup allegato.

Popup

È possibile usare i popup anche come livelli (quando si ha bisogno di qualcosa di più che collegare un popup a un oggetto):

```
var popup = L.popup()  
    .setLatLng([51.5, -0.09])  
    .setContent("I am a standalone popup.")  
    .openOn(mymap);
```

- Qui si usa ***openOn*** invece di ***addTo*** perché gestisce la chiusura automatica di un popup precedentemente aperto quando se ne apre uno nuovo, migliorando l'usabilità.

Eventi

Ogni volta che succede qualcosa in Leaflet, ad es. l'utente fa clic su un indicatore o cambia lo zoom della mappa, l'oggetto corrispondente invia un evento a cui è possibile aderire con una funzione. Questo consente di reagire all'interazione dell'utente.

Per aggiungere un **evento**:

```
function onMapClick(e) {  
    alert("You clicked the map at " + e.latlng);  
}  
mymap.on('click', onMapClick);
```

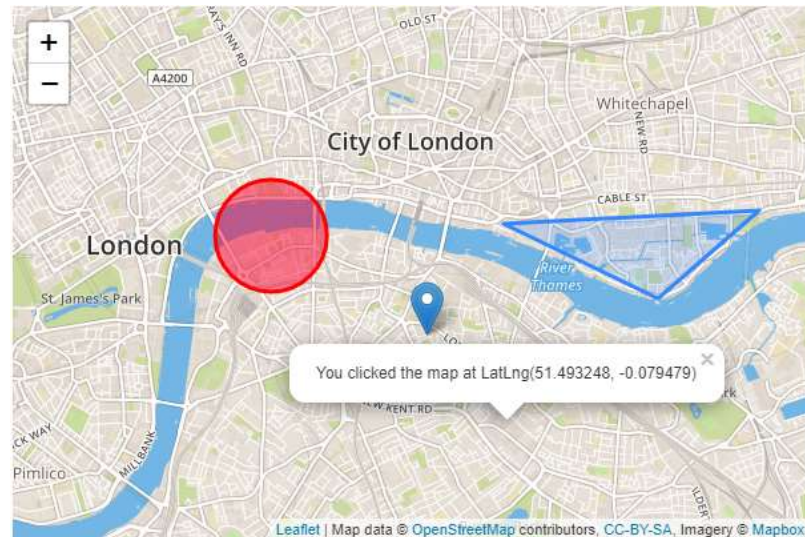
Eventi

- Ogni **oggetto** ha il proprio set di eventi.
- Il **primo argomento** della funzione di ascolto (`onMapClick`) è un oggetto evento.
- Contiene informazioni utili sull'evento che si è verificato.
- Esempio:
l'oggetto dell'evento di click sulla mappa ha la proprietà *latlng*, che è la posizione in cui si è verificato il click.

Eventi

È possibile utilizzare un popup anziché un avviso (alert). Cliccando sulla mappa, si vedranno le coordinate in un popup.

```
var popup = L.popup();  
function onMapClick(e) {  
    popup  
        .setLatLng(e.latlng)  
        .setContent("You clicked the map  
                    at" +  
e.latlng.toString())  
        .openOn(mymap);  
}
```



Conclusioni

- Leaflet rappresenta una valida alternativa a Google Maps.
- A differenza delle API di Google è completamente free e non necessita di API-Key.
- È possibile integrare Leaflet con Google Maps e con altri fornitori di servizi.
- È facilmente integrabile con i servizi di OpenLayers e OpenStreetMap.