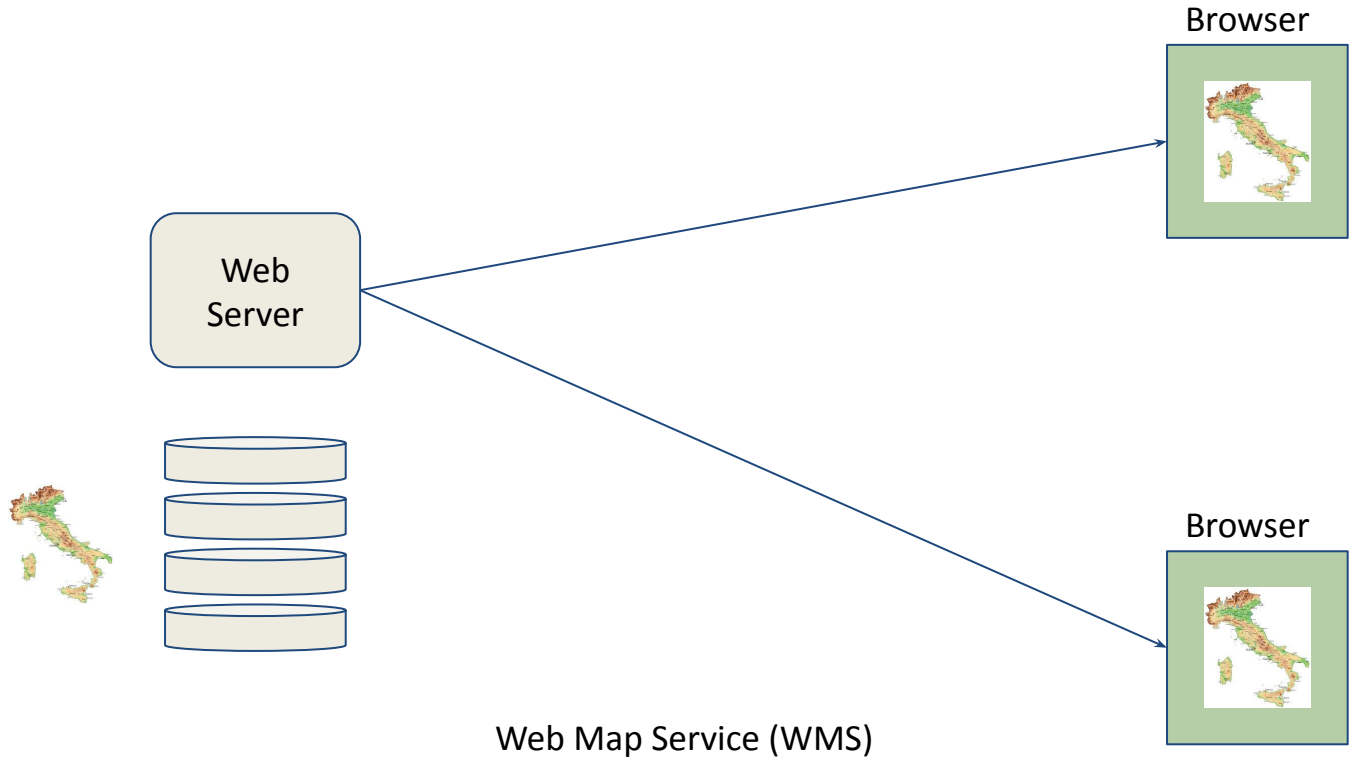
Tiled Web Map

InfoUma 2024-25 Andrea Marchetti

How to display a map on the web



What happens if the map is big?

Services like Google Maps face the following challenge:

How to display a high resolution map for every corner of the earth?

Tiled Web Map

An efficient solution to publish maps on the web

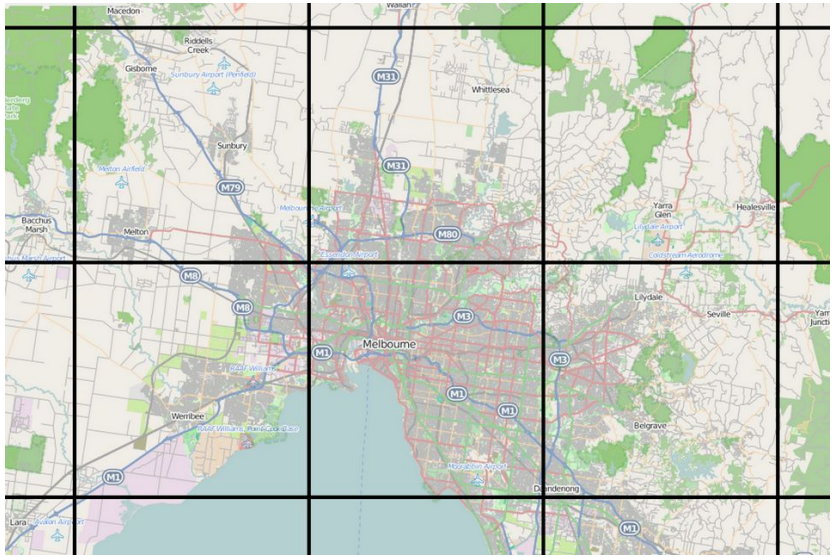
Complexity in space (rather than in time)

Used by many map providers

Google Maps, Bing, Yahoo Maps, OpenStreetMaps, ...

Tiled Web Map

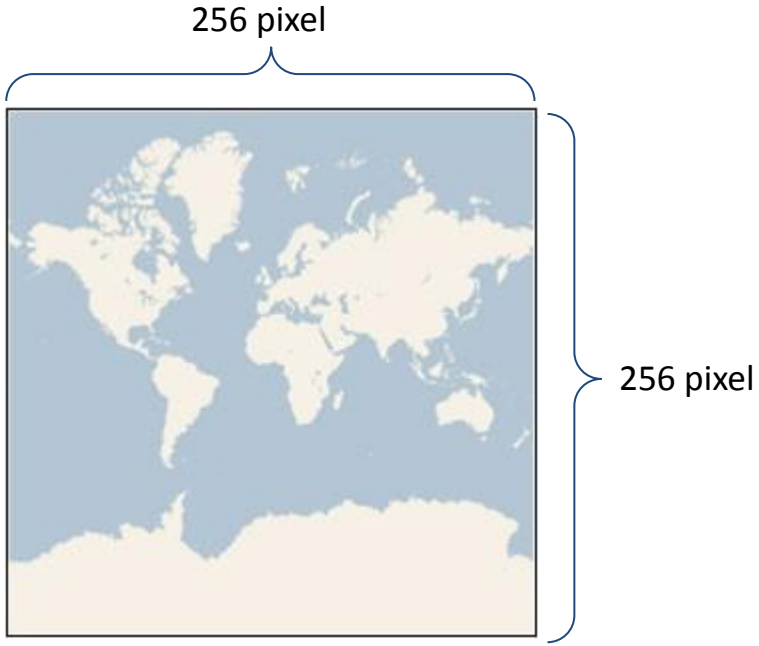
Tiled web maps, was developed by the Open Source Geospatial Foundation.



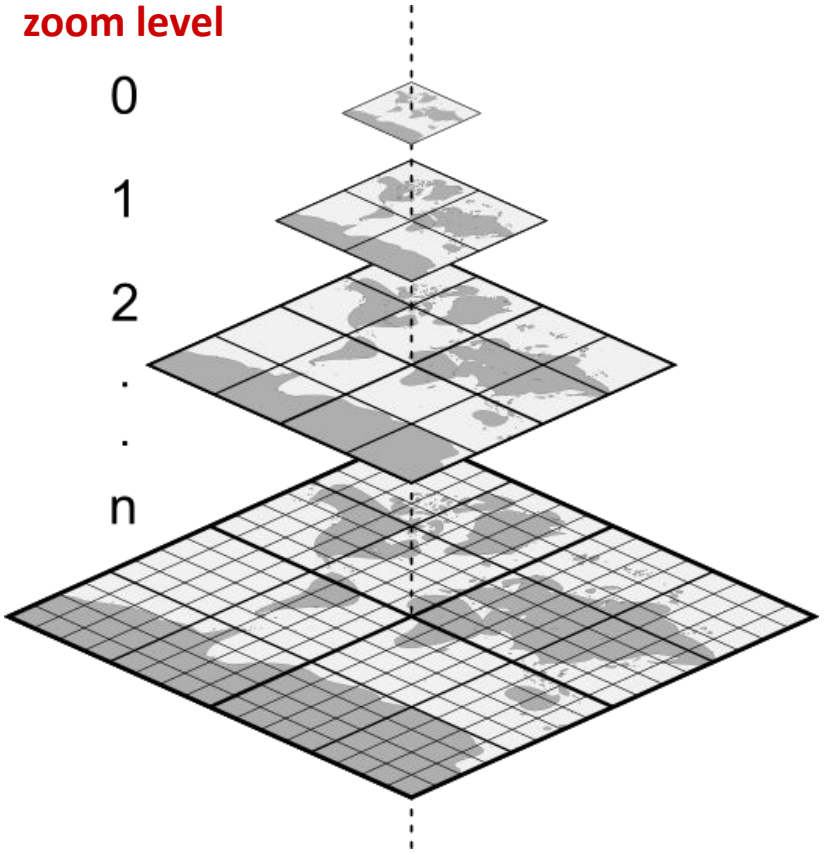
TILE

Every tile (any zoom) has a fixed dimension usually 256x256 pixel

At the outer most zoom level, 0, the entire world is rendered in a single map tile



Multi Resolution Image Pyramid

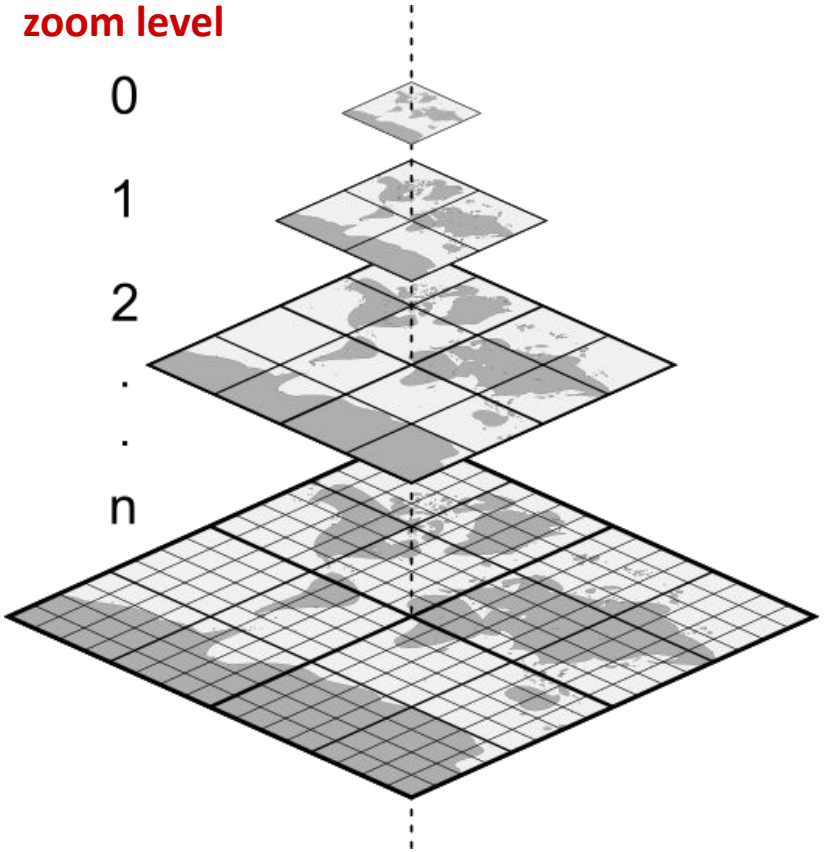


Maps are generated once for **all** level of zoom and then sliced into tiles

Each **zoom** level quadruples the number of tiles

At zoom level **n** there are $4^n = 2^{2n}$ tiles

Multi Resolution Image Pyramid

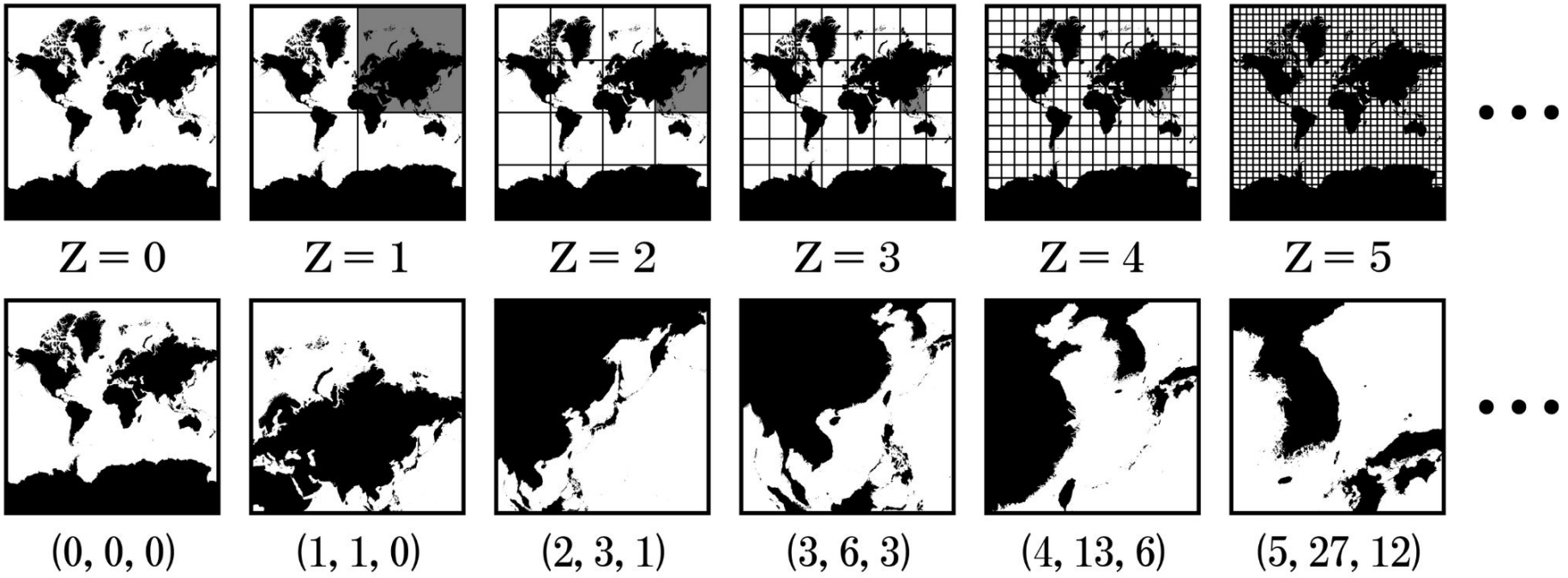


At level n we have the image at the maximum resolution divided into tiles

A pyramid is built starting from this image by reducing the resolution.

The space required to store a pyramid is one third more than the original image at the maximum resolution

Multi Resolution Image Pyramid



URL Tile format

Typical URL of a tile
`http://.../Z/X/Y.png`

Z is the zoom level,

X, Y identify the tile.

Z=2 X=3 Y=1

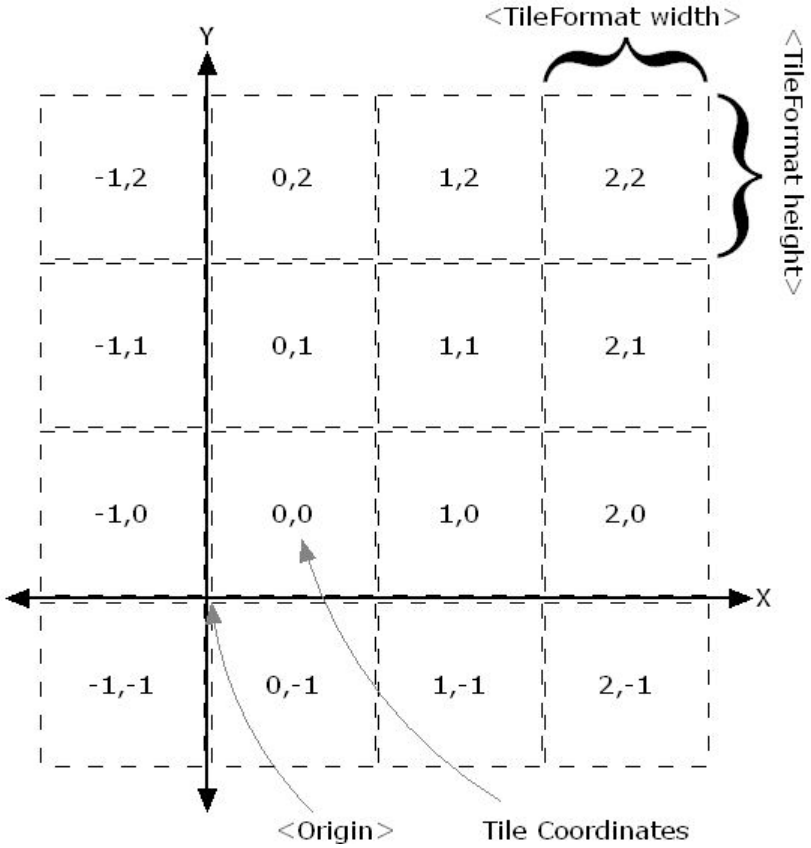
Tile Resource of CartoDB

https://a.basemaps.cartocdn.com/rastertiles/voyager/0/0/0.png

zoomLevel/X/Y.png

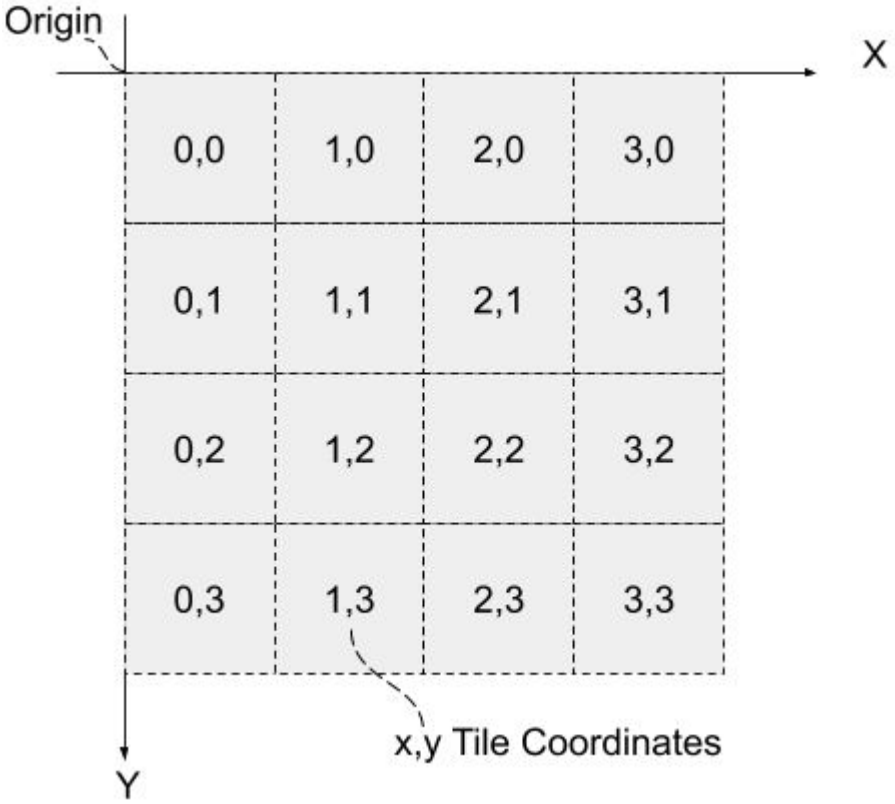
https://a.basemaps.cartocdn.com/rastertiles/voyager/1/0/0.png

Tile coordinates

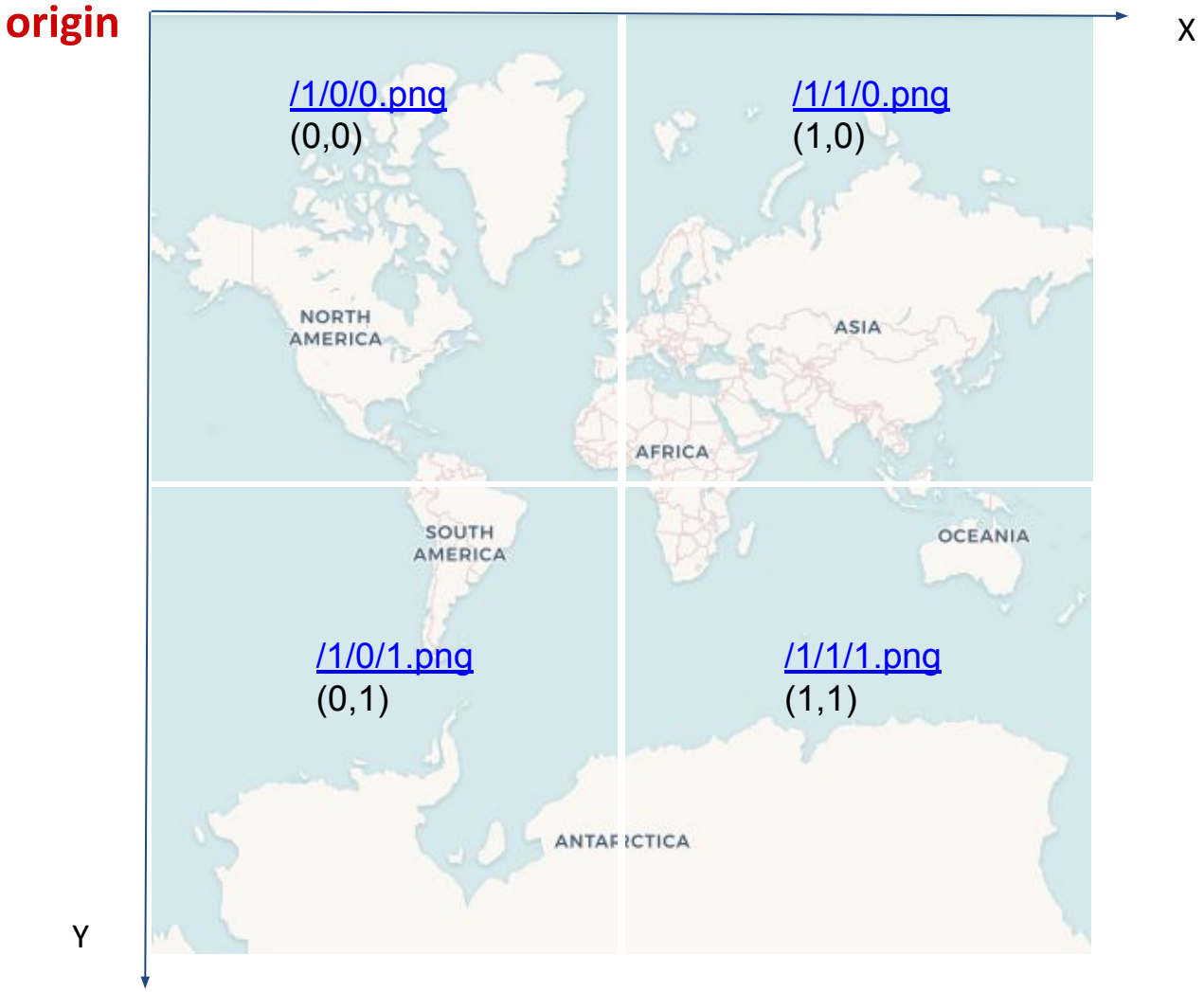


As depicted in the TMS specification—TileMap Diagram section, the Y-coordinates grow from south to north.

Tile Coordinates

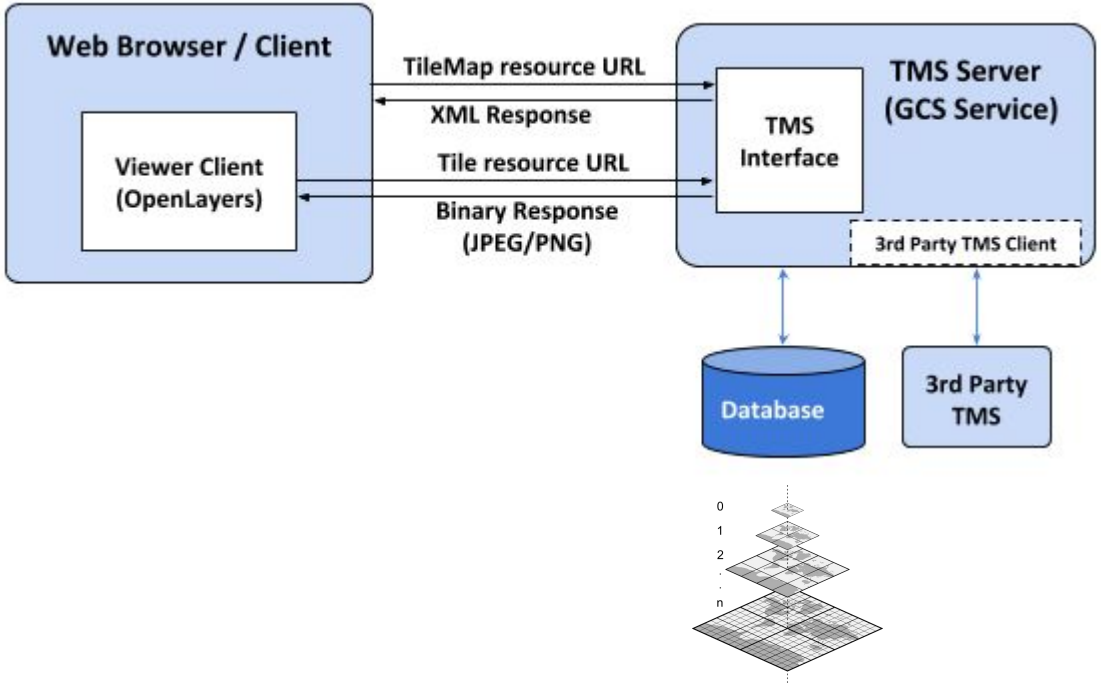


Some implementations have the opposite direction, with the grid origin at top left, and Y-coordinates numbered from north to south (e.g., OSM Tiles).
Depending on the implementation, it may be necessary to flip the y-coordinate



The 4 tiles at level 1 of CartoDB

Tiled Web Map Server



LeafLet

Leaflet is a JS library that simplifies access to a Tiled Web Map Server.

Leaflet Home Page

Leaflet

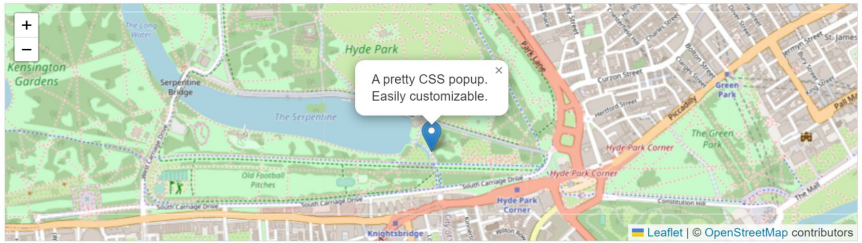
an open-source JavaScript library
for mobile-friendly interactive maps

Overview Tutorials Docs Download Plugins Blog

May 18, 2023 — Leaflet 1.9.4 has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. W[illegible]ing just about 42 KB of JS, it has all the mapping features most developers ever need.

Leaflet is designed with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code that is a joy to contribute to.



Version 1.9.4

See plugins for enhanced features

Many Geographic Features

Well documented

Leaflet Features

Features

Leaflet doesn't try to do everything for everyone. Instead it focuses on making *the basic things work perfectly*.

Layers Out of the Box

- Tile layers, WMS
- Markers, Popups
- Vector layers: polylines, polygons, circles, rectangles
- Image overlays
- GeoJSON

Interaction Features

- Drag panning with inertia
- Scroll wheel zoom
- Pinch-zoom on mobile
- Double click zoom
- Zoom to area (shift-drag)
- Keyboard navigation
- Events: click, mouseover, etc.
- Marker dragging

Visual Features

- Zoom and pan animation
- Tile and popup fade animation
- Very nice default design for markers, popups and map controls
- Retina resolution support

Customization Features

- Pure CSS3 popups and controls for easy restyling
- Image- and HTML-based markers
- A simple interface for custom map layers and controls
- Custom map projections (with `EPSG:3857/4326/3395` out of the box)
- Powerful OOP facilities for extending existing classes

Performance Features

- Hardware acceleration on mobile makes it feel as smooth as native apps
- Utilizing CSS3 features to make panning and zooming really smooth
- Smart polyline/polygon rendering with dynamic clipping and simplification makes it very fast
- Modular build system for leaving out features you don't need
- Tap delay elimination on mobile

Map Controls

- Zoom buttons
- Attribution
- Layer switcher
- Scale

Browser Support

Desktop

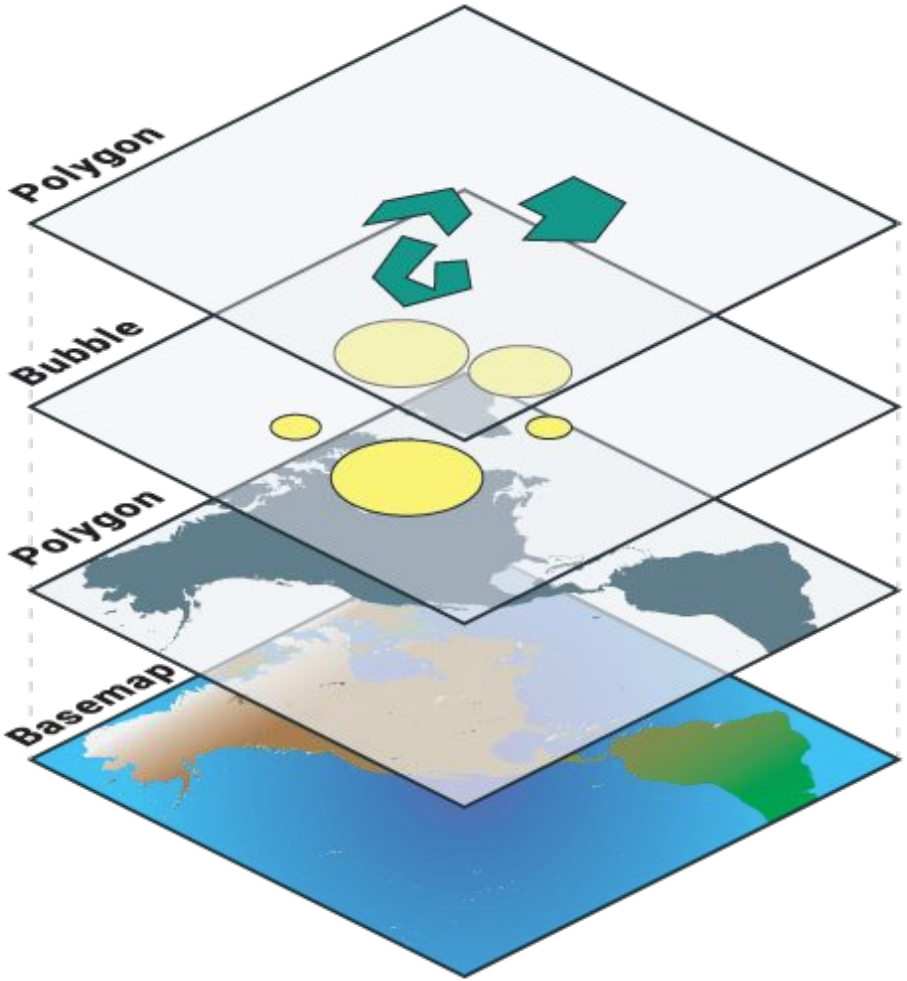
- Chrome
- Firefox
- Safari 5+
- Opera 12+
- IE 7–11
- Edge

Mobile

- Safari for iOS 7+
- Chrome for mobile
- Firefox for mobile
- IE10+ for Win8 devices

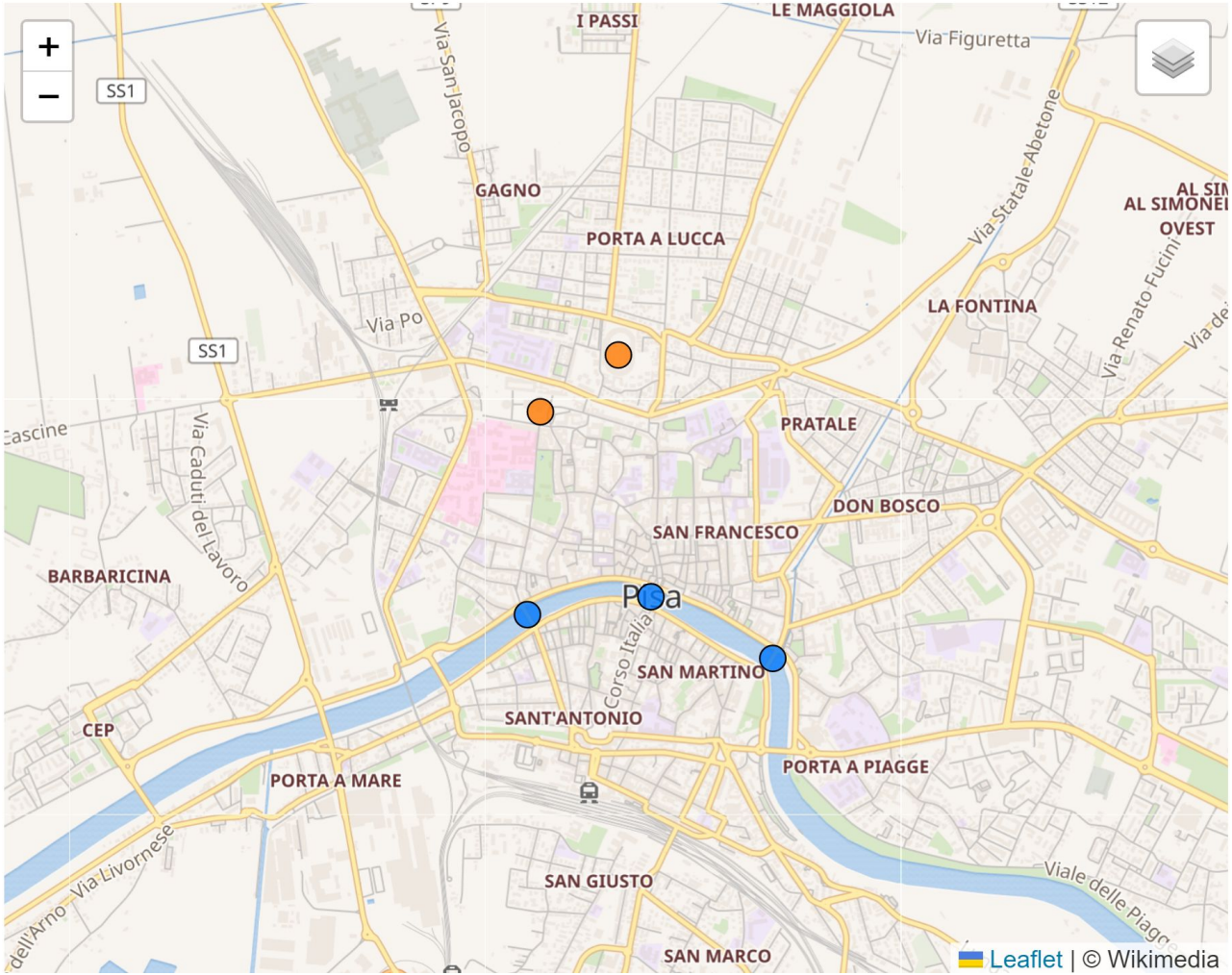
Misc

- Extremely lightweight
- No external dependencies

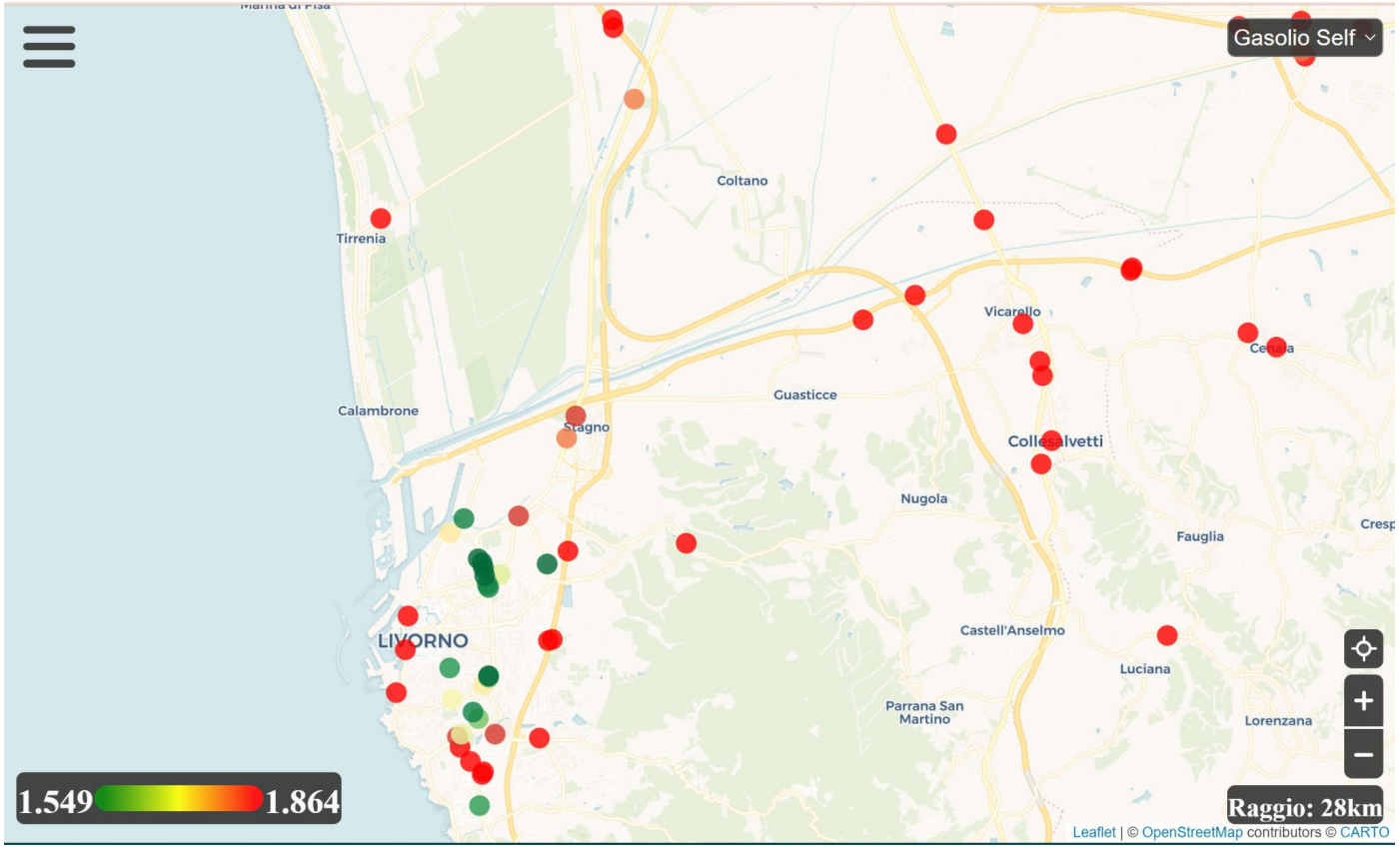


BaseMap provided by a Tile Map Providers

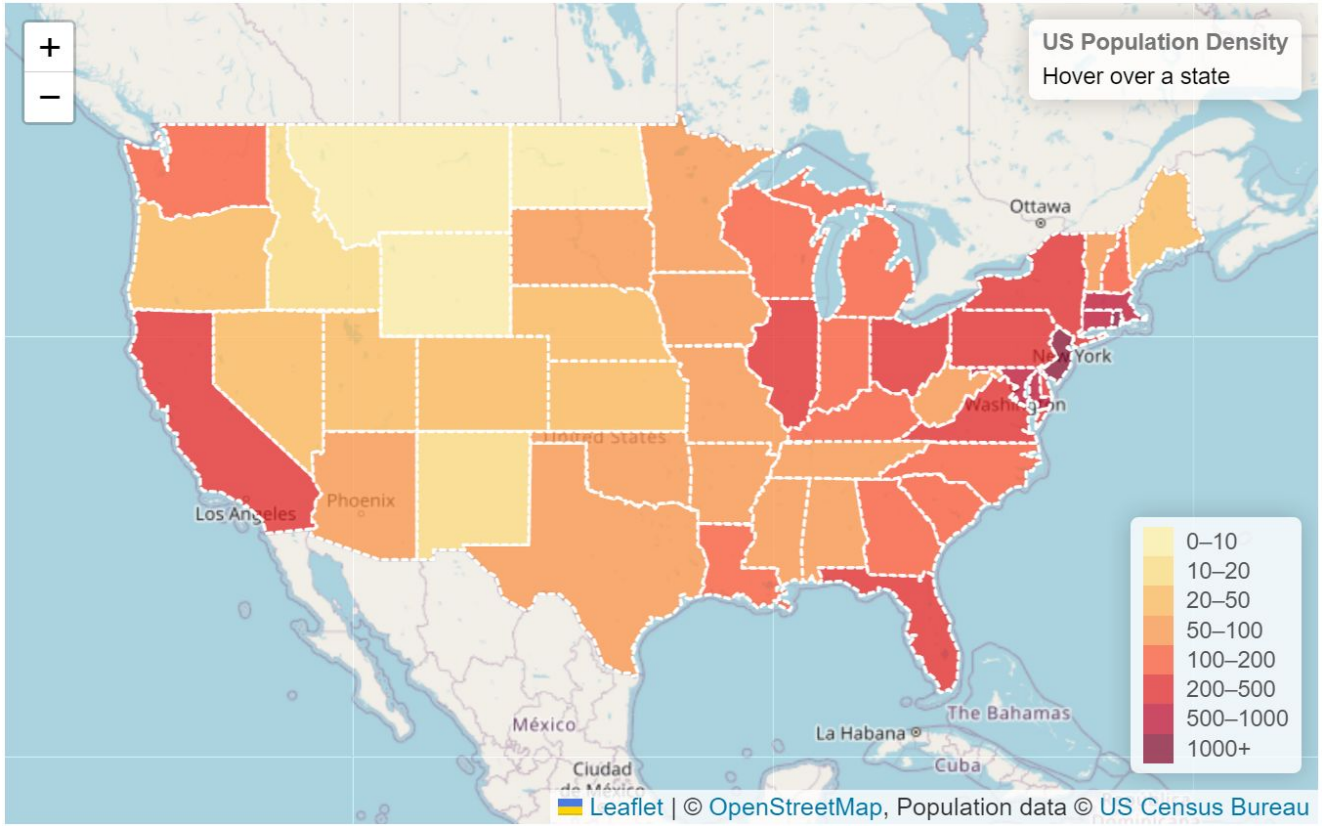
Charts with Leaflet


+
−
SS1
SS1
I PASSI
LE MAGGIOLA
Via Figuretta
Via San Jacopo
GAGNO
PORTA A LUCCA
Via Statale Abetone
AL SIMONE
OVEST
LA FONTINA
Via Renato Fucini
Via Po
PRATALE
Via Caduti del Lavoro
DON BOSCO
SAN FRANCESCO
BARBARICINA
Pisa
Corso Italia
SAN MARTINO
SANT'ANTONIO
CEP
PORTA A PIAGGE
PORTA A MARE
Via Livornese
SAN GIUSTO
Viale delle Piagg
SAN MARCO
Leaflet | © Wikimedia

Map with Markers



Choropleth Map



Leaflet.js

Easy to use API

Lightweight lib (only 42kb of js)

Support mobile applications

A valid tool to provide tile-based maps

Free Tiles Providers

- OpenStreetMap
  - Some issues for high traffic services
- MapQuest Open License
  - Free, by attribution
  - Special configuration for heavy usage
- MapBox
  - Free tier
  - Customizable design (see next slide)
  - Same family as Leaflet.js

Commercial Tile Providers

- CloudMade
  - Mirror of OSM data till few years ago
  - Leaflet was born here
  - $30 per 1M tiles
- MapBox
  - Free for low traffic
  - $30 for 900k tiles
- ESRI

Tile map providers

[Leaflet Provider Demo](#)

[Leaflet-extras/leaflet-providers](#)

[Map Compare](#)

[27- reasons-not-to-use-google-maps](#)

Warning

API Key: Some providers (e.g. Mapbox, MapTiler) require a free key or payment beyond certain limits.

CORS Policy: Some tile servers block requests if they do not come from approved domains.

CORS Policy

When you open a .html file directly (file://) in the browser, you do not go through HTTP/HTTPS, so:

There is no valid "origin" (the browser sees null as the origin).

Tile servers (OpenStreetMap, MapTiler, etc.) block CORS requests from null origin for security reasons.

✅ When you serve the file via Apache (or even a small HTTP server), everything works because the origin is known.

Solution

Using a lightweight local server

```
python -m http.server 8000 (python3)
```

```
python -m SimpleHTTPServer 8000 (python2)
```

Then open: http://localhost:8000/yourfile.html

Easy to install/use - Head Part

```
<head>
// Insert link to CSS
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
   integrity="sha256-p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY="
   crossorigin=""/>

// Insert link to JS
 <!-- Make sure you put this AFTER Leaflet's CSS -->
<script  src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"

integrity="sha512-BB3hKbKWOc9Ez/TAwyWxNXeoV9c1v6FIeYiBieIWkpLjauysF18NzgR1MBNBXf8/KABdlkX68nAhlwcDFLGPCQ=
="
   crossorigin=""></script>
</head>
```

Easy to install/use - Body Part

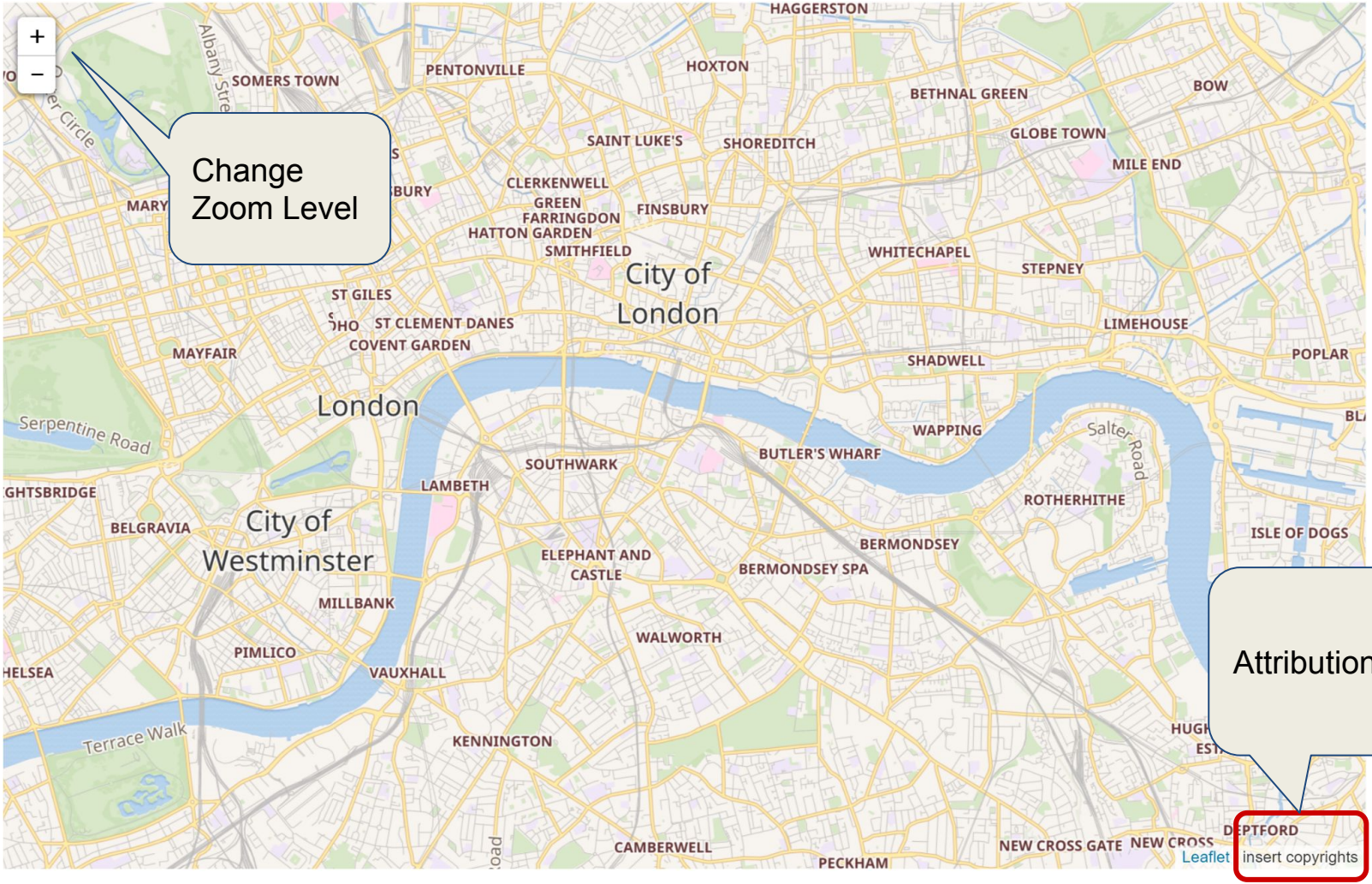
```
<body>
// Create a div element to contain the map
<div id="map" style="height: 600px"></div>

<script>

// Create an object to handle the map
var map = L.map('map').setView([51.505, -0.09], 13);

// Select the tile provider
var tms = 'https://maps.wikimedia.org/osm-intl/{z}/{x}/{y}@2x.png';

// Connect the tiles to our map
L.tileLayer(tms,{attribution: ''}).addTo(map);
</script>
</body>
```

Centre coordinates

Zoom Level

Wikimedia tile map provider

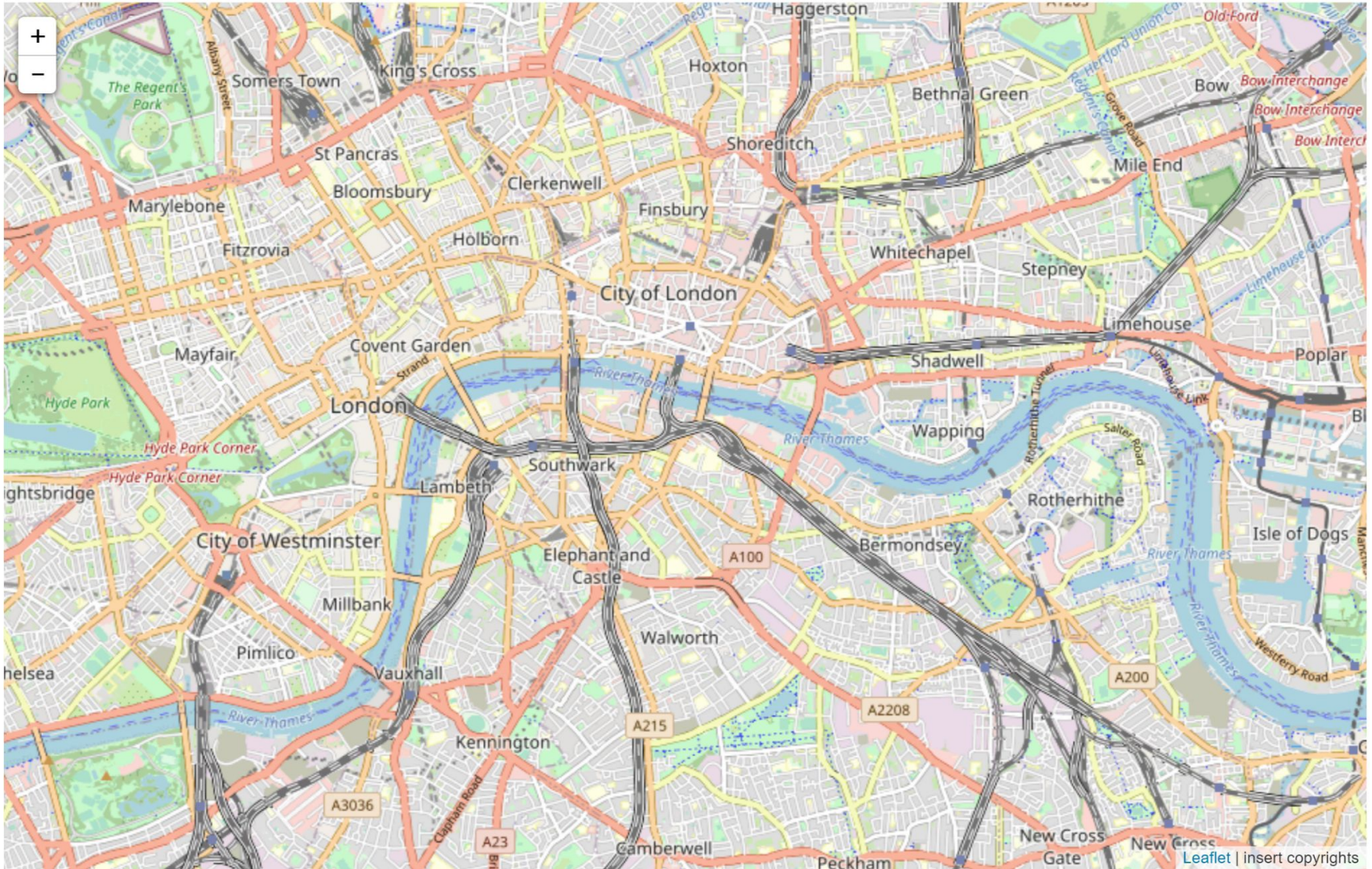
Copyright text as indicated by the tile provider


+
−
Change Zoom Level
HAGGERSTON
PENTONVILLE
HOXTON
SOMERS TOWN
BETHNAL GREEN
BOW
GLOBE TOWN
SAINT LUKE'S
SHOREDITCH
MILE END
CLERKENWELL
GREEN
FARRINGDON
FINSBURY
HATTON GARDEN
SMITHFIELD
WHITECHAPEL
STEPNEY
City of London
ST GILES
ST CLEMENT DANES
COVENT GARDEN
LIMEHOUSE
MAYFAIR
SHADWELL
POPLAR
London
Serpentine Road
WAPPING
Salter Road
BUTLER'S WHARF
SOUTHWARK
LAMBETH
ROTHERHITHE
BELGRAVIA
City of Westminster
BERMONDSEY
ISLE OF DOGS
ELEPHANT AND CASTLE
BERMONDSEY SPA
MILLBANK
WALWORTH
PIMLICO
VAUXHALL
Terrace Walk
KENNINGTON
CAMBERWELL
PECKHAM
NEW CROSS GATE
NEW CROSS
DEPTFORD
Leaflet | insert copyrights
Attribution

Change the tile provider

```
var tms =
'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
```

Openstreetmap tile map service



Exercise

Display the centre of Pisa 43.723, 10.396 with zoom 12

Markers and geometries

```
// Marker
var marker = L.marker([51.5, -0.09]);

// Circle
var circleOptions = {radius:5, color:'red'}
var circle = L.circle([51.508, -0.11], circleOptions);


// Polygon
var polygon = L.polygon([[51.509,-0.08],[51.503,-0.06],[51.51,-0.04]]);
```

Interactions - Popups

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

circle.bindPopup("I am a Circle.");

polygon.bindPopup("I am a Polygon.");
```

Layers

To display a marker ora geometry on the map we have 2 options:

```
// add each one to the map
marker.addTo(map);
circle.addTo(map);
polygon.addTo(map);

// group inside a single layer and then add the layer to the map
myLayer = L.layerGroup([marker, circle, polygono]);
myLayer.addTo(map)
```

Organize everything with layer

```
 // MARKERS
// Point of Interest layer
var poisOptions    = {radius: 8, fillColor: "#ff7800", color: "#000", weight: 1, opacity: 1, fillOpacity: 0.8};
var markerStadium = L.circleMarker([43.72518, 10.400103],poisOptions).bindPopup("<b>Pisa</b><br>The Stadium.");
var markerTower    = L.circleMarker([43.723, 10.396],poisOptions).bindPopup("<b>Pisa</b><br>The Leaning Tower.");
var poisLayer      = L.layerGroup([markerStadium, markerTower])

// Bridges layer
var bridgesOptions= {radius: 8, fillColor: "#0078ff", color: "#000", weight: 1, opacity: 1, fillOpacity: 0.8};
var markerBridge1 = L.circleMarker([43.715876, 10.401863],bridgesOptions).bindPopup("<b>Pisa</b><br>Middle Bridge.");
var markerBridge2 = L.circleMarker([43.713549, 10.408256],bridgesOptions).bindPopup("<b>Pisa</b><br>Fort Bridge.");
var markerBridge3 = L.circleMarker([43.715217, 10.395309],bridgesOptions).bindPopup("<b>Pisa</b><br>Solferino's Bridge.");
var bridgesLayer   = L.layerGroup([markerBridge1, markerBridge2, markerBridge3]);
```

Organize everything with layer

```
 // base layers or MAP TILE LAYER
var wikimedia     = 'https://maps.wikimedia.org/osm-intl/{z}/{x}/{y}@2x.png'
var OpenStreetMap = 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
var OpenTopoMap   = 'https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png'
var wikimediaLayer     = L.tileLayer(wikimedia , {attribution: '© Wikimedia'});
var OpenStreetMapLayer = L.tileLayer(OpenStreetMap , {attribution: '© OpenStreetMap'});
var OpenTopoMapLayer   = L.tileLayer(OpenTopoMap , {attribution: '© OpenTopoMap'});

// Pisa Map
var PisaMap =  L.map('map', {
    center: [43.72, 10.4],
    zoom: 14,
    layers: [wikimediaLayer,poisLayer] // Show wikimediaLayer and poisLayer by default
});
```

Organize everything with layer

```
 // DEFINE BASE LAYERS or TILE LAYER and OVERLAY LAYERS
var baseLayers = {
    "Wikimedia"    : wikimediaLayer,
    "OpenStreetMap": OpenStreetMapLayer,
    "OpenTopoMap"  : OpenTopoMapLayer
};
var overlays = {
    "Pois"         : poisLayer,
    "Bridges      ": bridgesLayer,
};
L.control.layers(baseLayers, overlays).addTo(PisaMap); // add everything to the map
```

Event handling

Every action in LeafLet, such as user click or zoom change, generates an event

We can define an handler to a particular event

```
function onMapClick(e) {
    alert("You clicked the map at " + e.latlng);
}
```

The first argument of the listener function is an event object — it contains useful information about the event that happened

And associate the handler to the event

```
map.on('click', onMapClick);
```

Event handling

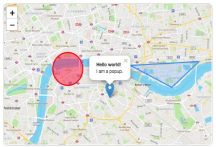
Let’s improve our example by using a popup instead of an alert

```
var popup = L.popup();

// Event handler for MapClick
function onMapClick(e) {
    popup
    .setLatLng(e.latlng)
    .setContent("You clicked the map at " + e.latlng.toString())
    .openOn(map);
}

// Subscription to the event 'click'
map.on('click', onMapClick);
```

Documentation

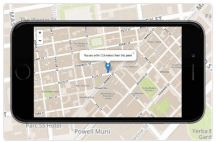
Leaflet Tutorials

Every tutorial here comes with step-by-step code explanation and is easy enough even for beginner JavaScript developers.



Leaflet Quick Start Guide

A simple step-by-step guide that will quickly get you started with Leaflet basics, including setting up a Leaflet map (with Mapbox tiles) on your page, working with markers, polylines and popups, and dealing with events.



Leaflet on Mobile

In this tutorial, you'll learn how to create a fullscreen map tuned for mobile devices like iPhone, iPad or Android phones, and how to easily detect and use the current user location.

Markers with Custom Icons

In this pretty tutorial, you'll learn how to easily define your own icons for use by the markers you put on the map.

Leaflet API reference

This reference reflects **Leaflet v1.8.0**. Check this list if you are using a different version of Leaflet.

Map

Usage example
Creation
Options
Events

Map Methods

Modifying map state
Getting map state
Layers and controls
Conversion methods
Other methods

Map Misc

Properties
Panes

UI Layers

Marker
DivOverlay
Popup
Tooltip

Raster Layers

TileLayer
TileLayer.WMS
ImageOverlay
VideoOverlay

Vector Layers

Path
Polyline
Polygon
Rectangle
Circle
CircleMarker
SVGOverlay
SVG
Canvas

Other Layers

LayerGroup
FeatureGroup
GeoJSON
GridLayer

Basic Types

LatLng
LatLngBounds
Point
Bounds
Icon
DivIcon

Controls

Zoom
Attribution
Layers
Scale

Utility

Browser
Util
Transformation
LineUtil
PolyUtil

DOM Utility

DomEvent
DomUtil
PosAnimation
Draggable

Base Classes

Class
Evented
Layer
Interactive layer
Control
Handler
Projection
CRS
Renderer

Misc

Event objects
global switches
noConflict
version

Plugins

Tile & image layers

Basemap providers
Basemap formats
Non-map base layers
Tile/image display
Tile load
Vector tiles

Overlay data

Overlay data formats
Dynamic data loading
Synthetic overlays
Data providers

Overlay Display

Markers & renderers
Overlay animations
Clustering/decluttering
Heatmaps
DataViz

Overlay interaction

Edit geometries
Time & elevation
Search & popups
Area/overlay selection

Map interaction

Layer switching controls
Interactive pan/zoom
Bookmarked pan/zoom
Fullscreen
Minimaps & synced maps
Measurement
Mouse coordinates
Events
User interface
Print/export
Geolocation

Miscellaneous

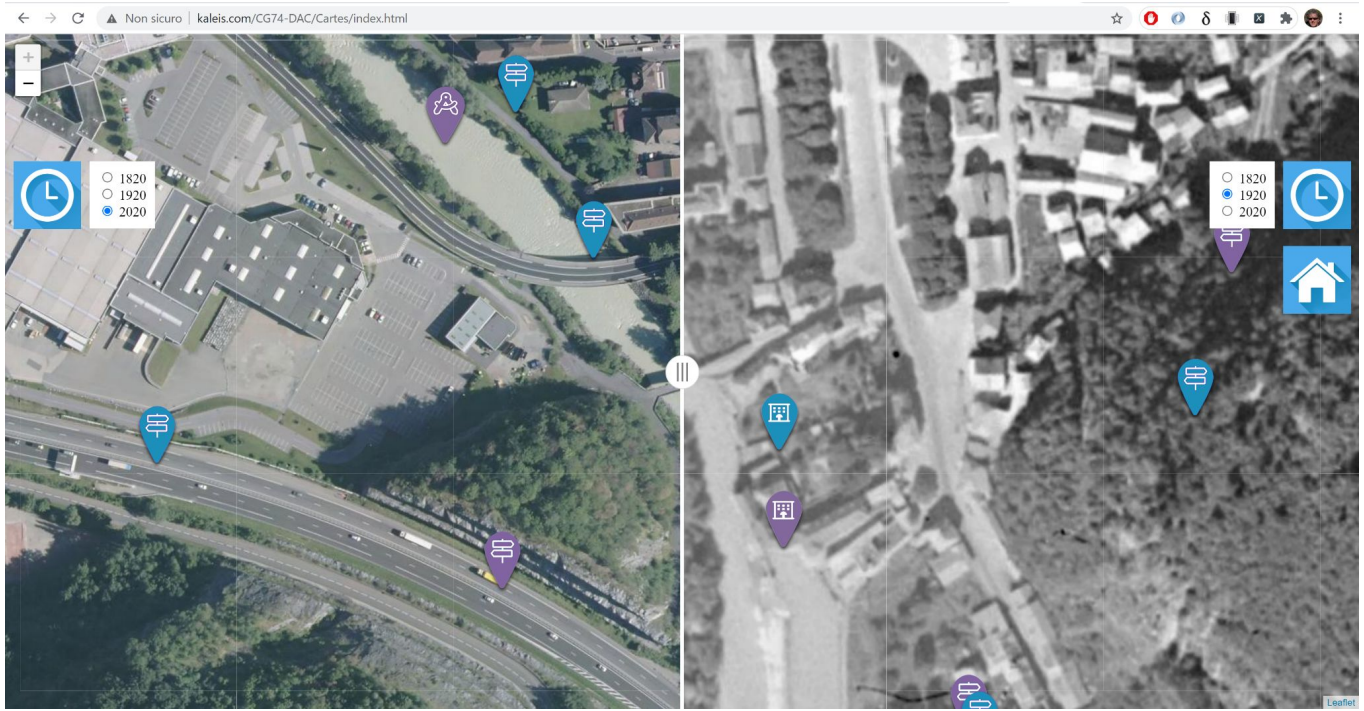
Geoprocessing
Routing
Geocoding
Plugin collections

Integration

Frameworks & build systems
3rd party

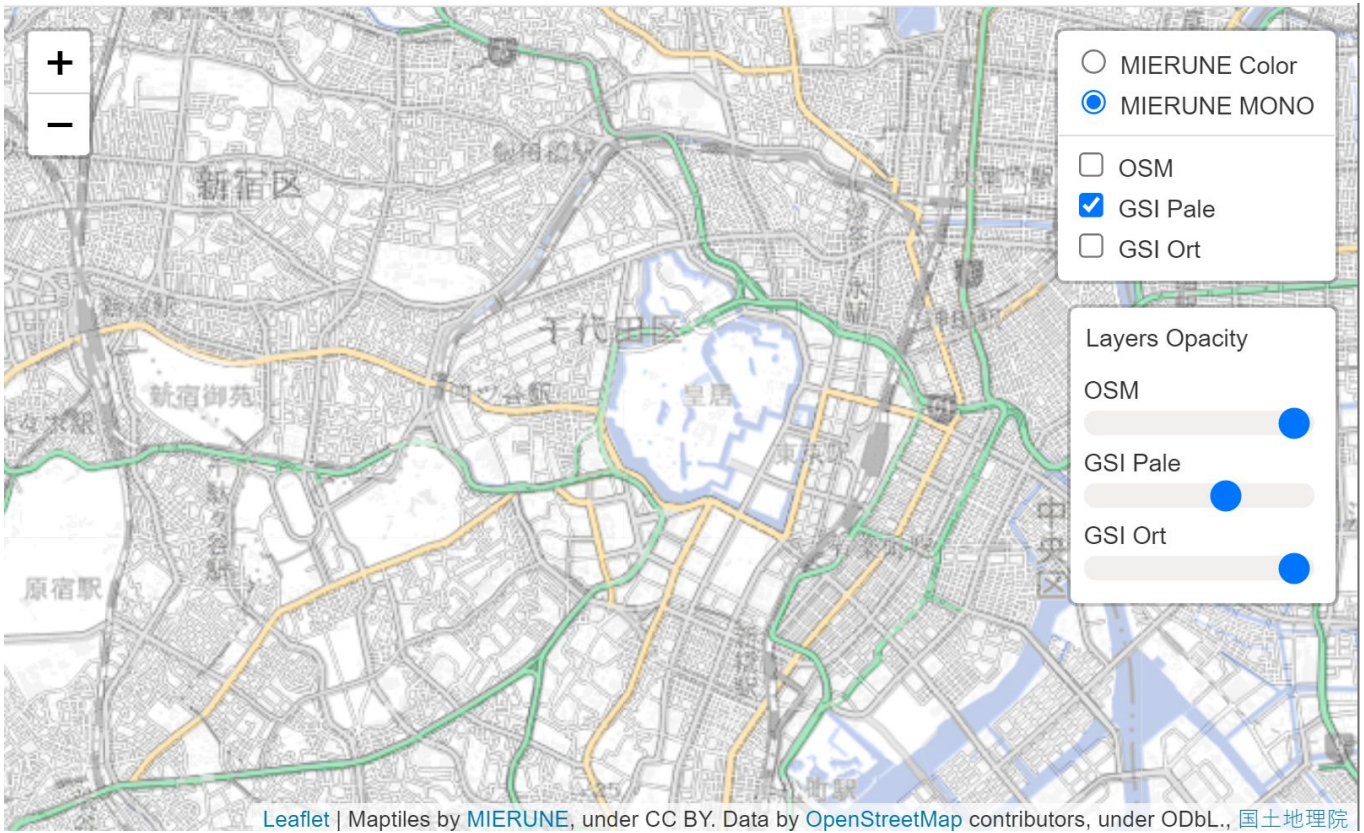
Develop your own

Plugin - Side by side

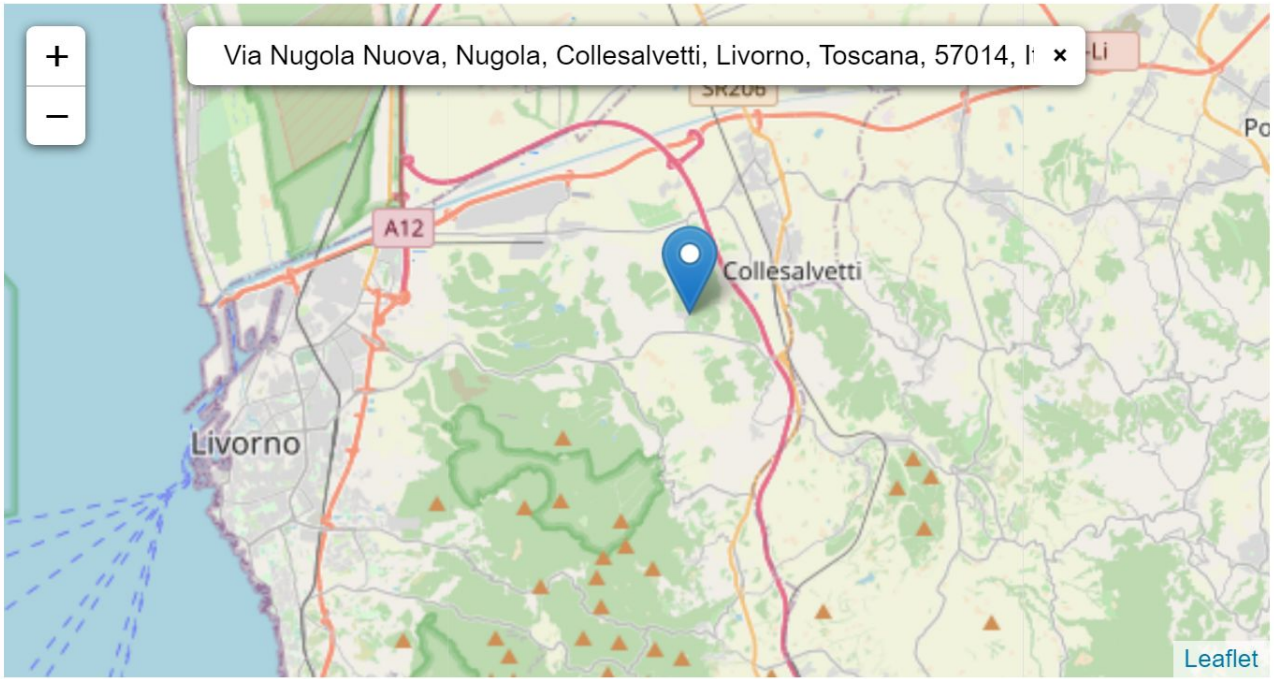


[A Leaflet control to add a split screen to compare two map overlays]

Plugin - Control layers opacity



Plugin - Geocoding



Alternatives to Leaflet in Python

Library	**Interactive**	**Based on Laeflet.js?**	**Note**
Folium	Y	Y	Easy
Plotly	Y	N based on Mapbox	Advanced Graphic
Bokeh	Y	N	
ipyleaflet	Y	Y	Very interactive

INGV

https://www.ingv.it/organizzazione/chi-siamo/personale/#739
Meletti Carlo
carlo.meletti@ingv.it

https://ingvterremoti.com/i-terremoti-in-italia/
https://emidius.mi.ingv.it/DBMI11/query_eq/
http://www.6aprile.it/conoscere-i-terremoti/2012/06/27/i-terremoti-piu-forti-della-storia-ditalia.html
https://ingvterremoti.com/la-pericolosita-sismica/
https://carta-sismicita-in-italia-ingv.hub.arcgis.com/