

Folium – a wrapper for

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

Quick Start

Installation

Use: `pip install folium`
or `pip install git+git+goo.gl/kmkGd3` - to install directly from GitHub

Basic Map

```
> import folium
> from folium import plugins
> m = folium.Map(location = [45.372, -121.6972])
> folium.Marker([45.3288, -121.6625],
    popup='<i>Mt. Hood Meadows</i>',
    tooltip='click me').add_to(m)
```

> m



Markers

Icon Markers

```
> folium.Marker([lat, lon], popup, tooltip) - add basic icon markers
> folium.Icon(color, icon) - customize marker icons
```

Circle Markers

```
> folium.Circle(location, fill_color, fill_opacity, color, color, _opacity,
radius, popup, ...) - Customize the color, radius, stroke, opacity
```

Vincent/Vega and Altair/VegaLite Markers

```
> m = folium.Map(location)
> VegaPop = folium.Popup(max_width).add_child(folium.Vega(json,
width, height)) - built-in support for vincent and altair visualizations
> folium.Marker(location, popup=VegaPop).add_to(m)
```

MarkerClusters

```
> m = folium.Map(location)
> marker_cluster = MarkerCluster().add_to(m)
> folium.Marker([lat, lon], popup, tooltip).add_to(marker_cluster)
```

ClickForMarker and PopOvers

```
> m = folium.Map(location)
> m.add_child(folium.LatLngPopup()) - conveniently add lat/lng
popovers
> m.add_child(folium.ClickForMarker(popup)) - on-the-fly placement of
markers
```

BoatMarker

```
> plugins.BoatMarker(location, heading, wind_heading, wind_speed)
- also: color, popup, icon, **kws
```

Lines and Shapes

Lines & PolyLineTextPath

```
> folium.PolyLine(locations, tooltip, popup).add_to(m) - also: color,
opacity, weight, smoothing_factor, line_cap, **kws
> plugins.PolyLineTextPath(folium.PolyLine(locations), text, repeat,
offset, attributes) - also: orientation, below, center, **kws
```

Polygons

```
> folium.Polygon(locations, tooltip, popup).add_to(m) - also: color,
opacity, weight, fill_color, fill_opacity, smooth_factor, no_clip, **kws
```

Circles

```
> folium.Circle(location, tooltip, popup).add_to(m) - also: color, opacity,
weight, fill_color, fill_opacity, radius, **kws
> folium.CircleMarker(location, tooltip, popup).add_to(m) - also: color,
opacity, weight, fill_color, fill_opacity, radius, **kws
```

Rectangles

```
> folium.Rectangle(bounds=[[lat,lon],[lat,lon]], tooltip,
popup).add_to(m) - also: color, fill_color, dash_array, weight, line_join,
line_cap, opacity, fill_opacity, **kws
```

GeoJson and TopoJson

GeoJson and GeoJsonCSS

```
> folium.GeoJson(GeoJson_path) - add name then use
folium.LayerControl().add_to(m)
> folium.GeoJsonCss(GeoJsonCss) - add styling, and popups into the data
```

TopoJson

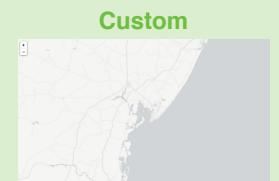
```
> folium.TopoJson(TopoJson_path) - add name then use
folium.LayerControl().add_to(m)
```

Choropleth

```
> folium.choropleth(geo_data, data, columns, key_on, fill_color) -
also: name, threshold_scale, line_color, line_weight, line_opacity, legend_name,
topojson, **kws
```

Basemaps

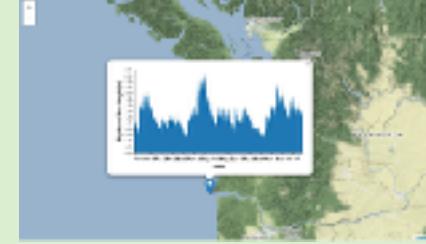
```
> folium.Map(location, tiles = 'Stamen Toner') - use a variety of tiles
```



Popups and Labels

```
> folium.Popup(html, parse_html, max_width)
```

Vega Popup



Fancy HTML Popup



Plugins

ScrollZoomToggler, Terminator and Fullscreen

```
> plugins.ScrollZoomToggler().add_to(m)
> plugins.Terminator().add_to(m)
> plugins.Fullscreen(position) - also: title, title_cancel, force_serperation_button
```

Search, Draw and MeasureControl

```
> plugins.Search(GeoJson, search_zoom, geom_type).add_to(m)
- also: search_label, position, popup_on_found
> plugins.Draw().add_to(m) - also: export
> m.add_child(plugins.MeasureControl()) - also: position, primary_length_unit,
primary_area_unit, secondary_length_unit, secondary_area_unit
```

FloatImage

```
> plugins.FloatImage(url, bottom, left).add_to(m)
```

TimestampedGeoJson

```
> plugins.TimestampedGeoJson(GeoJson, period) - also: add_last_point,
dateOptions, auto_play...
```



HeatMap and HeatMapWithTime

```
> plugins.HeatMap(data).add_to(m) - also: name, radius, min_opacity, max_val,
blur, gradient, overlay, max_zoom
> plugins.HeatMapWithTime(data).add_to(m) - also: index, name, radius,
min_opacity, max_opacity, auto_play, position, ...
```

