03. GOOGLE MAPS Overlays.

3.1. Google Maps Markers

This example shows an added marker to the Google Map:

Example:

Mapa
Satélite
Edimburgo
Glasgow
Reino Unido
Isla de Man
Irlanda
Dublín
Mánchester
Liverpool
Londres
Copenhague
Dinamarca
Hamburgo
Berlín
Polonia
Varsovia
Ámsterdam
Países Bajos
Bruselas
Bélgica
Colonia
Alemania
Fráncfort
Luxemburgo
Praga
Chequia
París
Francia
Múnich
Viena
Eslovaquia
Budapest
Austria
Hungría
Suiza
Eslovenia
Google
Datos de mapas ©2018 GeoBasis-DE/BKG (©2009), Google, Inst. Geogr. Nacional, Mapa GISrael, ORION-ME
Términos de uso

3.2. Google Maps Overlays

Overlays are objects on the map that are bound to latitude/longitude coordinates.

Google Maps has several types of overlays:

- **Marker** - Single locations on a map. Markers can also display custom icon images
- **Polyline** - Series of straight lines on a map
- **Polygon** - Series of straight lines on a map, and the shape is "closed"
- **Circle and Rectangle**
- **Info Windows** - Displays content within a popup balloon on top of a map
- **Custom overlays**

3.3. Google Maps – Add a Marker

The Marker constructor creates a marker. (Note that the position property must be set for the marker to display).

Add the marker to the map by using the **setMap**() method:

Example:

```
var marker = new google.maps.Marker({position: myCenter});
marker.setMap(map);
```

3.4. Google Maps - Animate the Marker

The example below shows how to animate the marker with the **animation** property:

Example:

```
var marker = new google.maps.Marker({
  position:myCenter,
  animation:google.maps.Animation.BOUNCE
  });

marker.setMap(map);
```

3.5. Google Maps – Icon Instead of Marker

The example below specifies an image (icon) to use instead of the default marker:

Example:

```
var marker=new google.maps.Marker({
  position:myCenter,
  icon:'pinkball.png'
  });

marker.setMap(map);
```

3.6. Google Maps – Polyline

A **Polyline** is a line that is drawn through a series of coordinates in an ordered sequence.

A polyline supports the following properties:

- **path** - specifies several latitude/longitude coordinates for the line
- **strokeColor** - specifies a hexadecimal color for the line (format: "#FFFFFF")
- **strokeOpacity** - specifies the opacity of the line (a value between 0.0 and 1.0)
- **strokeWeight** - specifies the weight of the line's stroke in pixels
- **editable** - defines whether the line is editable by users (true/false)

Example:

```
var myTrip = [stavanger,amsterdam,london];
var flightPath = new google.maps.Polyline({
  path:myTrip,
  strokeColor:"#0000FF",
  strokeOpacity:0.8,
  strokeWeight:2
});
```

3.7. Google Maps – Polygon

A **Polygon** is similar to a Polyline in that it consists of a series of coordinates in an ordered sequence. However, polygons are designed to define regions within a closed loop.

Polygons are made of straight lines, and the shape is "*closed*" (all the lines connect up). A polygon supports the following properties:

- **path** - specifies several LatLng coordinates for the line (first and last coordinate are equal)
- **strokeColor** - specifies a hexadecimal color for the line (format: "#FFFFFF")
- **strokeOpacity** - specifies the opacity of the line (a value between 0.0 and 1.0)
- **strokeWeight** - specifies the weight of the line's stroke in pixels
- **fillColor** - specifies a hexadecimal color for the area within the enclosed region (format: "#FFFFFF")
- **fillOpacity** - specifies the opacity of the fill color (a value between 0.0 and 1.0)
- **editable** - defines whether the line is editable by users (true/false)

Example:

```
var myTrip = [stavanger,amsterdam,london,stavanger];
var flightPath = new google.maps.Polygon({
  path:myTrip,
  strokeColor:"#0000FF",
  strokeOpacity:0.8,
  strokeWeight:2,
  fillColor:"#0000FF",
  fillOpacity:0.4
});
```

3.8. Google Maps – Circle

A **circle** supports the following properties:

- **center** - specifies the google.maps.LatLng of the center of the circle
- **radius** - specifies the radius of the circle, in meters
- **strokeColor** - specifies a hexadecimal color for the line around the circle (format: "#FFFFFF")
- **strokeOpacity** - specifies the opacity of the stroke color (a value between 0.0 and 1.0)
- **strokeWeight** - specifies the weight of the line's stroke in pixels
- **fillColor** - specifies a hexadecimal color for the area within the circle (format: "#FFFFFF")
- **fillOpacity** - specifies the opacity of the fill color (a value between 0.0 and 1.0)
- **editable** - defines whether the circle is editable by users (true/false)

Example:

```
var myCity = new google.maps.Circle({
  center:amsterdam,
  radius:20000,
  strokeColor:"#0000FF",
  strokeOpacity:0.8,
  strokeWeight:2,
  fillColor:"#0000FF",
  fillOpacity:0.4
});
```

3.9. Google Maps – InfoWindow

Show an InfoWindow with some text content for a marker:

Example:

```
var infowindow = new google.maps.InfoWindow({
  content:"Hello World!"
  });

infowindow.open(map,marker);
```