Lecture 10
Introduction to Google Maps

Senior-Lecturer: Sarsenova Zh.N.

Introduction

- 1. Step: Create an HTML page
- 2. Step: Add a map with marker
- 3. Step: Get an API key

https://developers.google.com/maps/documentation/javascript/examples/map-simple

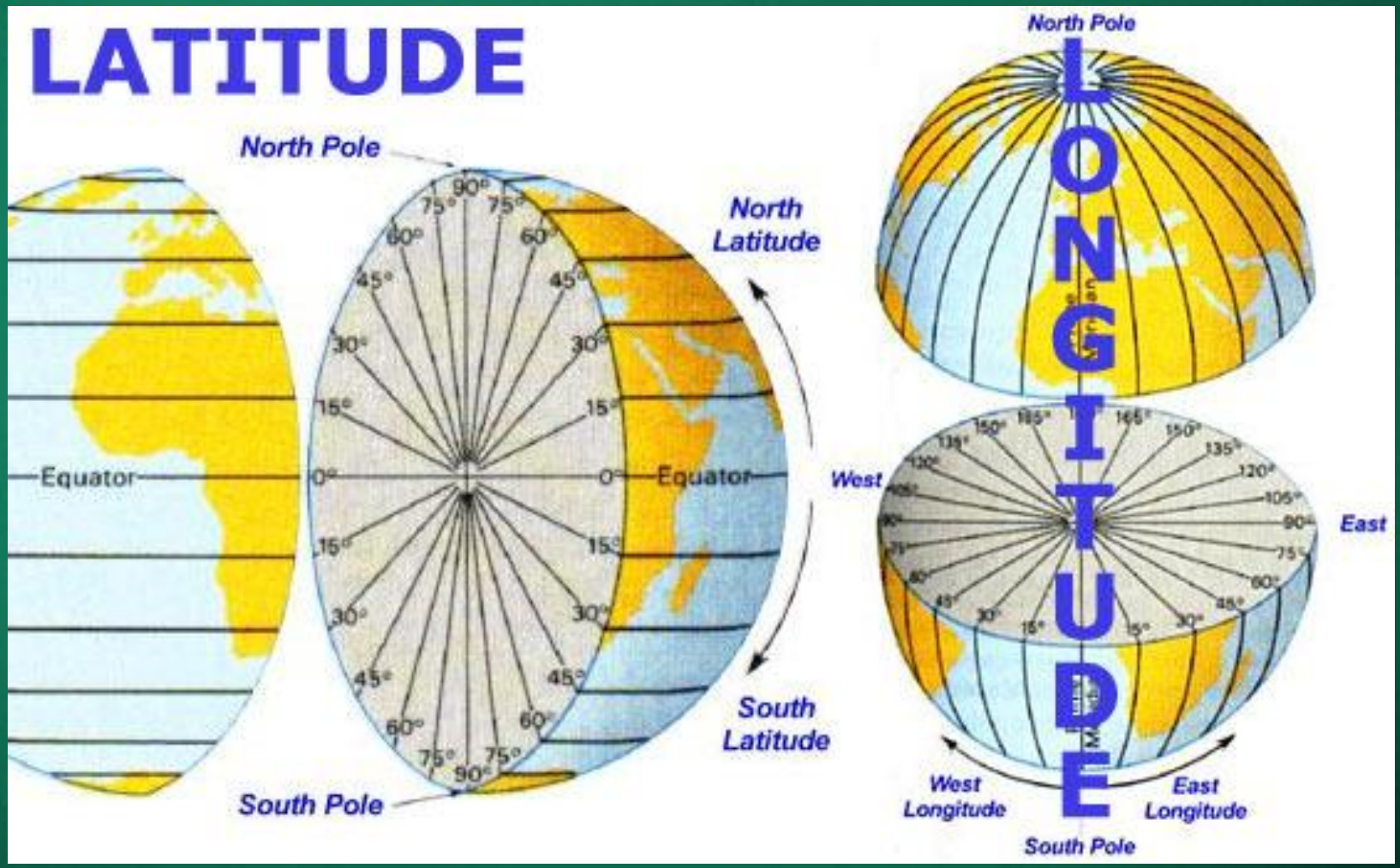
Google Maps API

- An API is a set of methods and tools that can be used for building software applications.

The Basic Skeleton of creating Google Map

```
<!DOCTYPE html>
<html>
<head>
    <style type="text/css">
        #map-canvas {
            height: 500px;
        }
    </style>
    <title></title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.0.0/jquery.min.js"></script>
    <script type="text/javascript"
            src="https://maps.googleapis.com/maps/api/js?sensor=false">
    </script>
</head>
<body>

<div id="map-canvas"></div>
</body>
</html>
```

Latitude and Longitude



The sample code need to create a map with marker

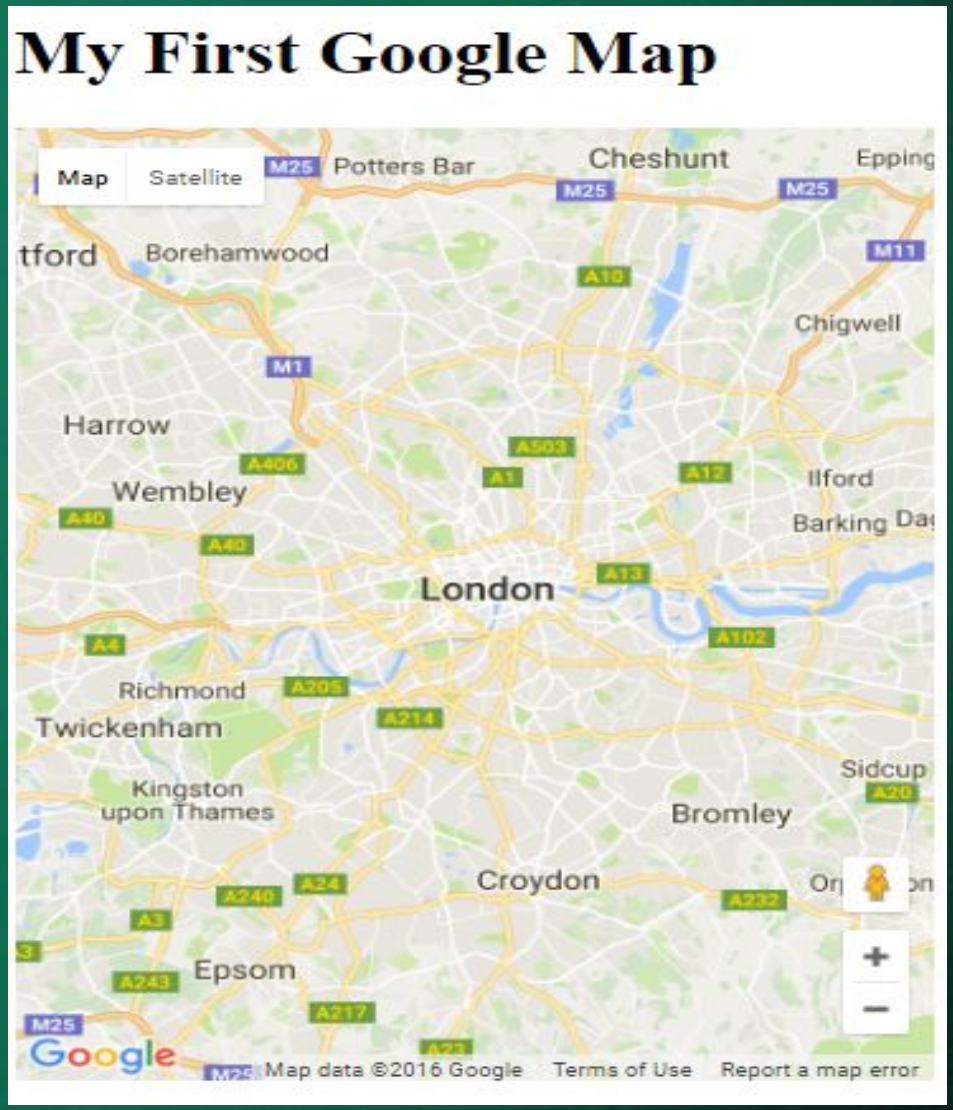
```
<!DOCTYPE html>
<html>
  <head>
    <style>
       #map {
        height: 400px;
        width: 100%;
       }
    </style>
  </head>
  <body>
    <h3>My Google Maps Demo</h3>
    <div id="map"></div>
    <script>
      function initMap() {
        var uluru = {lat: -25.363, lng: 131.044};
        var map = new google.maps.Map(document.getElementById('map'), {
          zoom: 4,
          center: uluru
        });
        var marker = new google.maps.Marker({
          position: uluru,
          map: map
        });
      }
    </script>
    <script async defer
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
    </script>
  </body>
</html>
```

Add this new Google maps object to construct a map in the div element

Add this code to put a marker on the map. The position property sets the position of the marker.

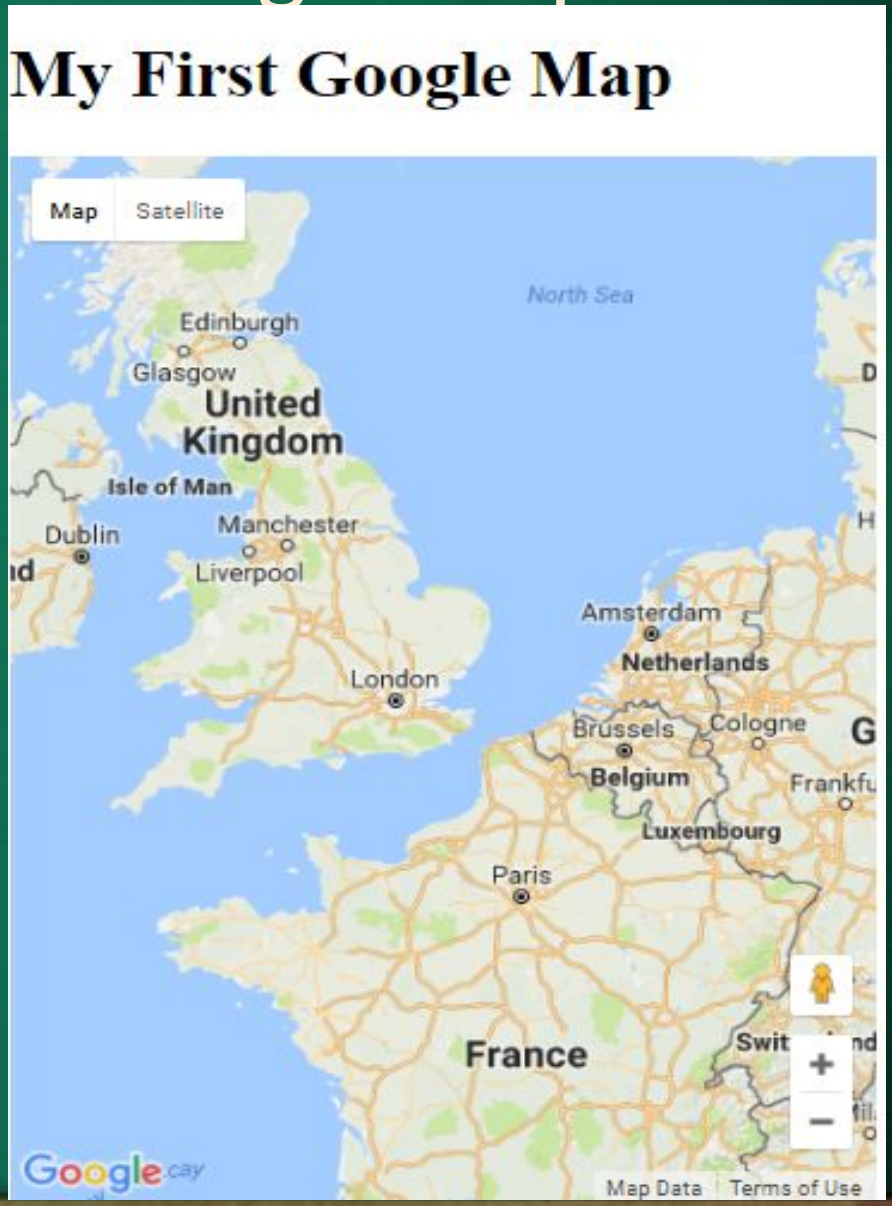
Google Maps in HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Google Map</h1>

<div id="map" style="width:100%;height:500px">
</div>

<script>
function myMap() {
  var mapCanvas =
document.getElementById("map");
  var mapOptions = {
    center: new google.maps.LatLng(51.5,
-0.2),
    zoom: 10
  }
  var map = new google.maps.Map(mapCanvas,
mapOptions);
}
</script>

<script
src="https://maps.googleapis.com/maps/api/js?
callback=myMap"></script>

</body>
</html>
```



Creating a Basic Google Map

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Google Map</h1>

<div id="map" style="width:100%;height:500px"
</div>

<script>
function myMap() {
  var mapCanvas =
document.getElementById("map");
  var mapOptions = {
    center: new
google.maps.LatLng(51.508742,-0.120850),
    zoom: 5
  };
  var map = new google.maps.Map(mapCanvas,
mapOptions);
}
</script>

<script
src="https://maps.googleapis.com/maps/api/js?
callback=myMap"></script>

</body>
</html>
```



The Map Container

- The map needs an HTML element to hold the map:
-
- The map will automatically "inherit" its size from its container element.

The Google Maps API

- The Google Maps API is a JavaScript library. It is added to the web page with a <script> tag:
- <script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script
- The **callback** parameter specifies the function to be called (**=myMap**) when the API is ready.

The myMap Function

- - myMap function initializes and display the map:

```
<script>
function myMap() {
  var mapCanvas = document.getElementById("map");
  var mapOptions = {
    center: new google.maps.LatLng(51.508742,-0.120850),
    zoom: 5
  };
var map = new google.maps.Map(mapCanvas, mapOptions);
}
</script>
```

re to

center

point.

ates in the order: latitude,

Zoom-specifies the zoom level for the map.

Zoom:0 shows a map of the Earth fully zoomed out. Higher zoom levels zoom in at a higher resolution.

New google.maps.Map() creates a new Google Maps object.

Different Map Types

```
<!DOCTYPE html>
<html>
<body>

<div id="googleMap1" style="width:400px;height:300px;"></div>
<br>
<div id="googleMap2" style="width:400px;height:300px;"></div>
<br>
<div id="googleMap3" style="width:400px;height:300px;"></div>
<br>
<div id="googleMap4" style="width:400px;height:300px;"></div>

<script>
function myMap() {
  var mapOptions1 = {
    center: new google.maps.LatLng(51.508742,-0.120850),
    zoom:9,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  var mapOptions2 = {
    center: new google.maps.LatLng(51.508742,-0.120850),
    zoom:9,
    mapTypeId: google.maps.MapTypeId.SATELLITE
  };
  var mapOptions3 = {
    center: new google.maps.LatLng(51.508742,-0.120850),
    zoom:9,
    mapTypeId: google.maps.MapTypeId.HYBRID
  };
  var mapOptions4 = {
    center: new google.maps.LatLng(51.508742,-0.120850),
    zoom:9,
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };
  var map1 = new google.maps.Map(document.getElementById("googleMap1"),mapOptions1);
  var map2 = new google.maps.Map(document.getElementById("googleMap2"),mapOptions2);
  var map3 = new google.maps.Map(document.getElementById("googleMap3"),mapOptions3);
  var map4 = new google.maps.Map(document.getElementById("googleMap4"),mapOptions4);
}
</script>

<script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>

</body>
</html>
```

Google Maps overlays

- Overlays are objects on the map that are bound to latitude/longitude coordinates.
- 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 Window – Displays content within a popup balloon on top of a map.
- Custom overlays

Adding a Marker

```
<!DOCTYPE html>
<html>

<body>

<div id="map" style="width:100%;height:500px"
</div>

<script>
function myMap() {
  var myCenter = new
google.maps.LatLng(51.508742,-0.120850);
  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: myCenter, zoom:
5};
  var map = new google.maps.Map(mapCanvas,
mapOptions);
  var marker = new
google.maps.Marker({position:myCenter});
  marker.setMap(map);
}
</script>

<script
src="https://maps.googleapis.com/maps/api/js?
callback=myMap"></script>

</body>
</html>
```

- The Marker constructor creates a marker. And The position property must be set for the marker to display.

Animating the Marker (Example)

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

```
<!DOCTYPE html>
<html>

<body>

<div id="map"
style="width:100%;height:500px"></div>

<script>
function myMap() {
  var mapCanvas =
document.getElementById("map");
  var myCenter = new
google.maps.LatLng(51.508742,-0.120850);
  var mapOptions = {center: myCenter, zoom:
5};
  var map = new
google.maps.Map(mapCanvas,mapOptions);
  var marker = new google.maps.Marker({
    position: myCenter,
    animation: google.maps.Animation.BOUNCE
  });
  marker.setMap(map);
}
</script>

<script
src="https://maps.googleapis.com/maps/api/j
s?callback=myMap"></script>

</body>
</html>
```

Icons instead of Marker

- We can specify an image (icon) to use of the default marker

```
  var map = new
google.maps.Map(mapCanvas,mapOptions);
  var marker = new google.maps.Marker({
    position: myCenter,
    icon: "pinkball.png"
  });
  marker.setMap(map);
}
```

Polyline

- A polyline is a line that is drawn through a series of coordinates in an ordered sequence.
- A polyline supports the following properties:

A polygline 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 user(true/false)

Example

```
<script>
function myMap() {
  var stavanger = new
google.maps.LatLng(58.983991,5.734863);
  var amsterdam = new
google.maps.LatLng(52.395715,4.888916);
  var london = new
google.maps.LatLng(51.508742,-0.120850);

  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: amsterdam,
zoom: 4};
  var map = new
google.maps.Map(mapCanvas,mapOptions);

  var flightPath = new
google.maps.Polyline({
    path: [stavanger, amsterdam, london],
    strokeColor: "#0000FF",
    strokeOpacity: 0.8,
    strokeWeight: 2
  });
  flightPath.setMap(map);
}
</script>


<script
```

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 polygone supports the following properties:

Path – specifies several LatLng coordinates for the line(first and last coordinates 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

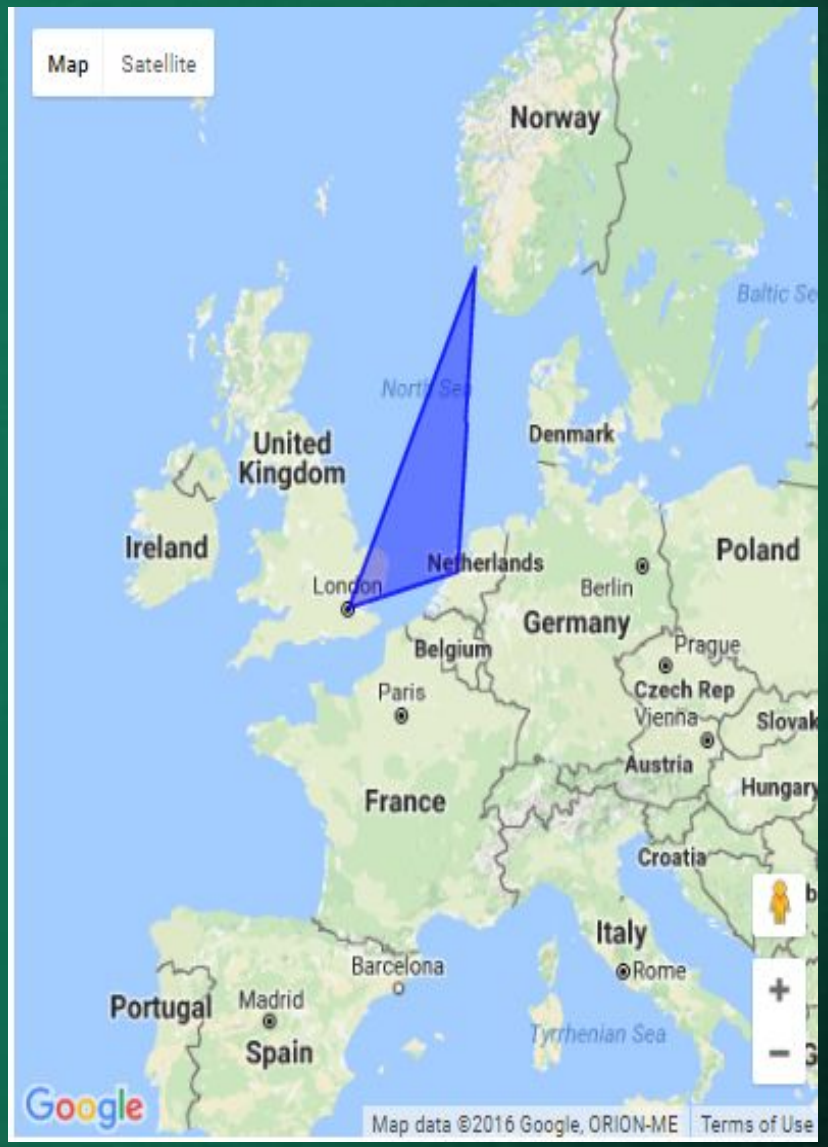
fillColor-specifies a hexadecimal color for the area within the enclosed region

fillOpacity-specifies the opacity of the fill color(value between 0.0 and 1.0)

editable- defines whether the line is editable by users(true/false)

Polygon Example

```
<script>
function myMap() {
  var stavanger = new
google.maps.LatLng(58.983991,5.734863);
  var amsterdam = new
google.maps.LatLng(52.395715,4.888916);
  var london = new
google.maps.LatLng(51.508742,-0.120850);

  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: amsterdam, zoom: 4};
  var map = new
google.maps.Map(mapCanvas,mapOptions);

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



A circle supports the following properties:

Center-specifies the google.maps.LatLng of the center of the circle

strokeColor - specifies a hexadecimal color for the line(format:#FFFFFF)

Radius-specifies the radius of the circle, inmeters

editable-defines whether the line is editable by users(true/false)

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

fillOpacity-specifies the opacity of the fill color(value between 0.0 and 1.0)

editable-defines whether the line is editable by users(true/false)

Google Maps – Circle Example

```
<script>
function myMap() {
  var amsterdam = new
google.maps.LatLng(52.395715,4.888916);

  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: amsterdam, zoom: 7};
  var map = new
google.maps.Map(mapCanvas,mapOptions);

  var myCity = new google.maps.Circle({
    center: amsterdam,
    radius: 50000,
    strokeColor: "#0000FF",
    strokeOpacity: 0.8,
    strokeWeight: 2,
    fillColor: "#0000FF",
    fillOpacity: 0.4
  });
  myCity.setMap(map);
}
</script>
```

Google Maps - InfoWindow

- Show in InfoWindow with some text content for a marker

```
<script>
function myMap() {
  var myCenter = new
google.maps.LatLng(51.508742,-0.120850);
  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: myCenter, zoom: 5};
  var map = new google.maps.Map(mapCanvas,
mapOptions);
  var marker = new
google.maps.Marker({position:myCenter});
  marker.setMap(map);

  var infowindow = new google.maps.InfoWindow({
    content: "Hello World!"
  });
  infowindow.open(map,marker);
}
</script>

<script
src="https://maps.googleapis.com/maps/api/js?
callback=myMap"></script>
```

Google Maps Events

Click the Marker to Zoom

```
// Zoom to 9 when clicking on marker
google.maps.event.addListener(marker,'click',function() {
  map.setZoom(9);
  map.setCenter(marker.getPosition());
  });
```

- We register for event notifications using the addListener() event handler. That method takes an object, an event to listen for, and a function to call when the specified event occurs.

Full Example

```
<script>
function myMap() {
  var myCenter = new
google.maps.LatLng(51.508742,-0.120850);
  var mapCanvas =
document.getElementById("map");
  var mapOptions = {center: myCenter, zoom: 5};
  var map = new google.maps.Map(mapCanvas,
mapOptions);
  var marker = new
google.maps.Marker({position:myCenter});
  marker.setMap(map);

  // Zoom to 9 when clicking on marker
  google.maps.event.addListener(marker,'click',f
unction() {
    map.setZoom(9);
    map.setCenter(marker.getPosition());
  });
}
</script>

<script
src="https://maps.googleapis.com/maps/api/js?
callback=myMap"></script>
```

Pan Back to Marker

- Here, we save the zoom changes and pan the map back after 3 seconds:

```
google.maps.event.addListener(map,'click',function() {
  var pos = map.getZoom();
  map.setZoom(9);
  map.setCenter(marker.getPosition());
  window.setTimeout(function() {map.setZoom(pos);},3000);
});
```

Open an InfoWindow When clicking on the Marker

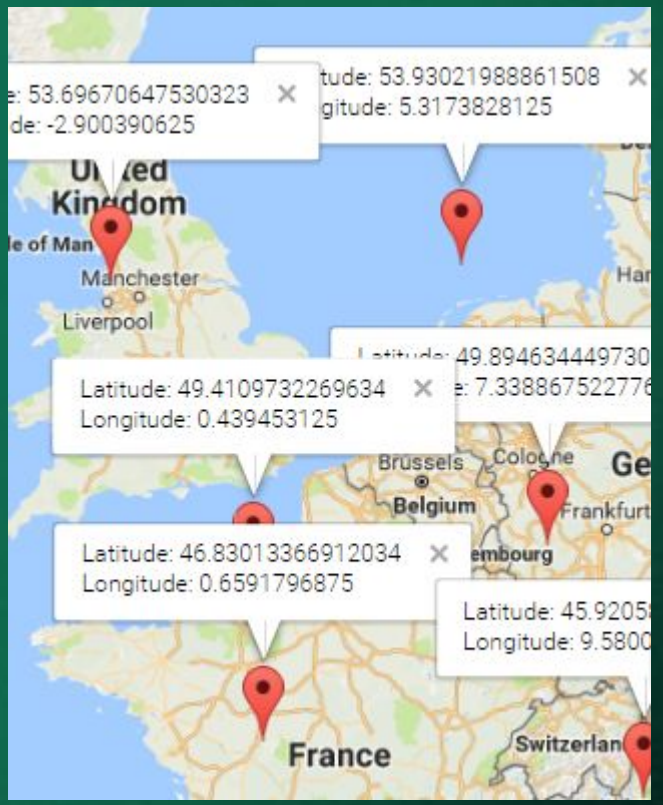
- Click on the marker to show an infowindow with some text:

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

google.maps.event.addListener(marker, 'click', function() {
  infowindow.open(map,marker);
  });
```

Set Markers and Open InfoWindow for Each Marker

- The placeMarker() function places a marker where the user has clicked, and shows an infowindow with the latitude and longitude of the marker:

```
google.maps.event.addListener(map, 'click', function(event) {
  placeMarker(map, event.latLng);
  });

function placeMarker(map, location) {
  var marker = new google.maps.Marker({
    position: location,
    map: map
  });
  var infowindow = new google.maps.InfoWindow({
    content: 'Latitude: ' + location.lat() +
    '<br>Longitude: ' + location.lng()
  });
  infowindow.open(map,marker);
}
```



Google Maps Types

Google Maps API supports:

- ROADMAP (normal, default 2D map)
- SATELLITE (photographic map)
- HYBRID (photographic map + road and city names)
- TERRAIN (map with mountains, river, etc)

Types of Google Map

- The map type is specified either within the Map properties object, with the mapTypeId property:

```
var mapOptions = {
  center:new google.maps.LatLng(51.508742,-0.120850),
  zoom:7,
  mapTypeId: google.maps.MapTypeId.HYBRID
};
```

- Or by calling the map's setMapTypeId() method:

```
map.setMapTypeId(google.maps.MapTypeId.HYBRID);
```

Good Luck!!!