

# Complex Polylines



View this example [full screen](#).

JavaScript

JavaScript + HTML

```
// This example creates an interactive map which constructs a  
// polyline based on user clicks. Note that the polyline only appears  
// once its path property contains two LatLng coordinates.
```

```
var poly;
```

```
var map;
```

```
function initialize() {
```

```
  var mapOptions = {
```

```
    zoom: 7,
```

```
    // Center the map on Chicago, USA.
```

```
    center: new google.maps.LatLng(41.879535, -87.624333)
```

```
  };
```

```
  map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
```

```
  var polyOptions = {
```

```
    strokeColor: '#000000',
```

```
    strokeOpacity: 1.0,
```

```
    strokeWeight: 3
```

```
};  
poly = new google.maps.Polyline(polyOptions);  
poly.setMap(map);  
  
// Add a listener for the click event  
google.maps.event.addListener(map, 'click', addLatLng);  
}  
  
/**  
 * Handles click events on a map, and adds a new point to the Polyline.  
 * @param {google.maps.MouseEvent} event  
 */  
function addLatLng(event) {  
  
    var path = poly.getPath();  
  
    // Because path is an MVCArray, we can simply append a new coordinate  
    // and it will automatically appear.  
    path.push(event.latLng);  
  
    // Add a new marker at the new plotted point on the polyline.  
    var marker = new google.maps.Marker({  
        position: event.latLng,  
        title: '#' + path.getLength(),  
        map: map  
    });  
}  
  
google.maps.event.addDomListener(window, 'load', initialize);
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#).

Last updated March 17, 2015.