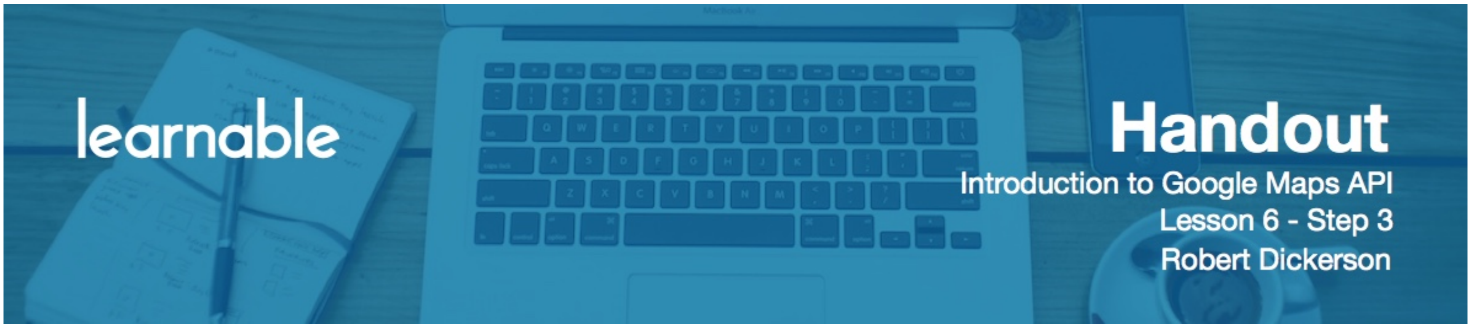

learnable

Handout

Introduction to Google Maps API
Lesson 6 - Step 3
Robert Dickerson

Introduction

Displaying a marker on a map is useful, but we want users to be able to interact with it and get more information about the point of interest. This is why **info windows** were created. An info window is just a pop-up that contains HTML or plain text and can be triggered when a user clicks on a marker.

At this step you will learn how to create an info window, add and style the HTML content for the window and then associate the info window with the marker click event.

Create Info Window

What we want to do display an info window when a user clicks on the airplane. The window should contain airport arrival and departure statistics.

Create a function called `addInfoWindow` and pass the marker into it:

```
function addInfoWindow(marker) {
  var contentString = '<div class="infowindowcontent">'+
      '<div class="row">' +
      '<p class="total greenbk">78.3%</p>'+
      '<p class="location">NEW YORK NY</p>'+
      '<p class="code">JFK</p>'+
      '</div>'+
      '<div class="data">'+
      '<p class="tagbelow">Avg On-Time</p>'+
      '<p class="label">Arrivals</p>'+
      '<p class="details">76% (8,590)</p>' +
      '<p class="label">Departures</p>'+
      '<p class="details">80.5% (8,589)</p>' +
      '<p class="coords">40.6413111 , -73.77813909</p>' +
      '</div>' +
      '</div>';
  var infoWindow = new google.maps.InfoWindow({
    content: contentString,

    disableAutoPan: false,

    maxWidth: 300,

    zIndex: 1

  });

  google.maps.event.addListener(marker, 'click', function(e) {

    infoWindow.open(map,marker);
  });
}
```

We create a new info window object and save it to the variable `infowindow`. A set of parameters is passed to configure it.

The first parameter defines the content that we want to display.

`disableAutoPan` means that when the info window is open, the map will be panned so you can get a full view of the info window.

`maxWidth` sets the maximum width of the info window itself in pixels.

`zIndex` is used in order to determine the priority of the overlays. If you have multiple info windows that lay on top of each other, you can set the one that appears on the top or set the ordering that they appear.

Then we attach a click handler to the marker, so that when user clicks on the marker, an info window will be displayed. The `open` method takes the map and the marker you want to bind to.

Style HTML content

Now let's discuss HTML content displayed inside the infowindow and style it. The content is based on data we want to display for the course project. For each US airport, we will be showing the on-time arrivals, departures and total number of flights for the December 2014. We show the average on-time percentage along with individual percentages for the arrival and departures. We include the airport code, city and state and coordinate location.

Let's apply some styling to the content:

```
.infowindowcontent {
  font-family: "FranklinITCStdLight","Helvetica","Arial",sans-serif;
  line-height: 10px;

  width: 300px;
  height: 225px;

  padding: 0px;
  margin: 0px;
}

.row {
  margin: 0px;
  padding: 0px;
}

.row .total {
  float: left;
  font-size: 28px;

  margin: 0px 10px 5px 0px;
  padding: 20px 10px 20px 10px;
  text-align: center;
}

.highlight {

  width: 60px;
  padding: 5px;
  margin: 0px;

}

.row .location {
  margin: 0px;
  padding: 10px 0px 10px 0px;
  color: #333;
  font-size: 16px;
}

.row .code {
  margin: 0px;
  padding: 3px 0px 0px 10px;
  font-size: 26px;
}

.data {
  margin-top: 14px;
  padding:0px;
}

.tagbelow {
  font-size: 10px;

}

.label {
  text-transform: uppercase;
  font-size: 16px;
}

.details {
  font-size: 18px;
}

.coords {
  font-size: 10px;
}

.greenbk {
  background: #00a651;
}

.yellowbk {
  background: #fff202;
}

.orangebk {
  background: #ff8300;
}

.redbk {
  background: #ff2d08;
}
```

Display Info Window on marker click

Let's wrap up this step by discussing how you can force a click event trigger to occur to open an info window. This is particularly useful if you want to open an info window automatically on load or after a search.

Google provides an `event.trigger` method:

```
google.maps.event.trigger(newMarker, 'click');
```

It takes two parameters: the marker and the event we want to call.