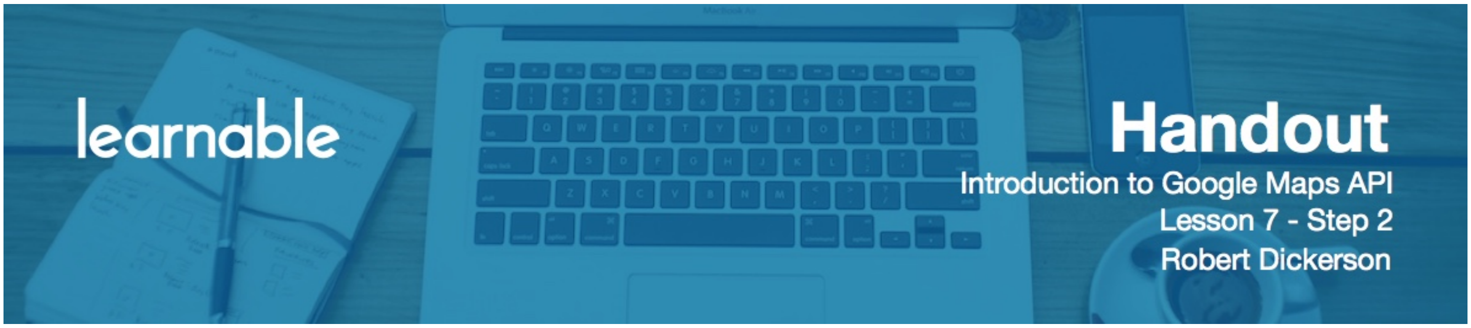

learnable

Handout

Introduction to Google Maps API
Lesson 7 - Step 2
Robert Dickerson

Cleaning up a bit of previous code

At this step we will cover two of the goals identified for our course project, displaying all the airports on the map, and creating a styled info window with each airport's unique data.

Display all airport on map

Include the airport data file on the page:

```
<script type="text/javascript" src="js/airportdata.js"></script>
```

In our `loadMap` function we want to create a `for` loop to go through each of the airports in the array and add them to the map:

```
for (var i=0;i<airportData.length;i++) {
  var airport = airportData[i];
  newMarker = this.addMarker(airport);
  addInfoWindow(newMarker);
}
```

Our airport data is saved to the `airportData` variable, so we use it here.

Update Info Window content

Inside the `addInfoWindow` we need to be able to access the parameters of the airport data, so that we can update the contents. To do that we're going to bind the airport data to the marker. Before we update the HTML content of the info window,we need to calculate the average percentage of arrival and departures, the total flights, and set the icon color and default size. We will add these calculations as properties to the airport object variable:

```
for (var i=0;i<airportData.length;i++) {
  var airport = airportData[i];

  airport.totalper = (airport.aper + airport.dper)/2;

  airport.totalflights = (airport.aop + airport.dop);

  airport.icon = 'green';

  airport.iconsize = new google.maps.Size(32,32);

  newMarker = this.addMarker(airport);

  newMarker.airport = airport;

  addInfoWindow(newMarker);
}
```

Now we can start updating the static values in the HTML to reflect each individual airport's data. The first thing we're going to do is add a local variable inside the `addInfoWindow` that points to the airport properties we bound to the marker object. This is just to make the code cleaner to read as we start integrating this content with HTML:

```
var details = marker.airport;
```

Now add this code instead of a static content:

```
var contentString = '<div class="infowindowcontent">'+
    '<div class="row">' +
    '<p class="total '+details.icon+'bk">'+Math.round(details.totalper*10)/10+'%</p>'+
    '<p class="location">'+details.airport.split("(")[0].substring(0,19)+'</p>'+
    '<p class="code">'+details.code+'</p>'+
    '</div>'+
    '<div class="data">'+
    '<p class="tagbelow">Avg On-Time</p>'+
    '<p class="label">Arrivals</p>'+
    '<p class="details">'+details.aper+'% ('+numberWithCommas(details.aop)+')</p>' +
    '<p class="label">Departures</p>'+
    '<p class="details">'+details.dper+'% ('+numberWithCommas(details.dop)+')</p>' +
    '<p class="coords">'+details.lat+' , '+details.lng+'</p>' +
    '</div>'+
    '</div>';
```

As I click on the markers, you'll notice that multiple info windows get created on every marker, so the user has to go back and close it. What I'd like to do is have one info window that moves and closes, as I click around on the individual markers. Go ahead and remove creating of the info window.

Instead of passing the parameters such as the content, I'm going to add those dynamically when the marker's clicked. I'll set the content to be specific to that particular marker. That way only a single info window will open up as you click on the different markers - only the content itself will be updated.

So let's update the content dynamically inside our event listener:

```
google.maps.event.addListener(marker, 'click', function() {
  infoWindow.close();

    infoWindow.setContent(contentString);

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

We close any info windows that are already open.

Next set the content of the info window by using the `setContent` method.

Lastly open the window.