# IN712 Web 3 – JavaScript libraries and Web APIs Practical

1. Use galleria to show an image gallery on a simple HTML document.
2. Create an application that retrieves the stock price of a company stock symbol, as entered by a user, using the URL provided by Yahoo finance API:

[https://query.yahooapis.com/v1/public/yql?q=select%20\*%20from%20yahoo.finance.quote%20where%20symbol%20in%20(%22STOCKTICKETSYMBOLGOESHERE%22)&format=json&diagnostics=true&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys&callback](https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20yahoo.finance.quote%20where%20symbol%20in%20(%22STOCKTICKETSYMBOLGOESHERE%22)&format=json&diagnostics=true&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys&callback)=

for testing purposes, you can check out the symbols of different tech companies in the following URL: <http://www.nasdaq.com/screening/companies-by-industry.aspx?industry=Technology&sortname=marketcap&sorttype=1>

this exercise involves a minimum amount of code writing. Make sure though you understand what p2.js is doing.

1. Build a web page that displays a map using the Google Maps API. You must provide a text box for user input. The user types a string into the text box, then clicks on the map. A marker is inserted at the click location, and the tooltip and info window for that marker is set to the user-supplied contents of the text box.

Google Maps API Syntactic Features (all but the first refer to the JavaScript API, and should be placed in your .js file):

|  |  |
| --- | --- |
| Loading the library | In the <head> of your HTML file, include:  <script src = "http://maps.googleapis.com/maps/api/js?sensor=false">  </script> |
| Creating a map object | var *mapObject* = new google.maps.Map(*mapCanvas*, *mapOptions*); |
| *mapOptions* | A JavaScript literal object that must contain at a minimum, the fields   * center: a LatLng object (see below) * zoom: an integer between 0 and 21 * mapTypeId: an instance of the google.maps.MapTypeId enumeration (see below) |
| MapTypeId enumeration | google.maps.MapTypeId.ROADMAP  google.maps.MapTypeId.SATELLITE  google.maps.MapTypeId.HYBRID  google.maps.MapTypeId.TERRAIN |
| *mapCanvas* | The DOM object for a div on your web page (i.e. what you get by invoking document.getElementById). The map will expand to the size of the div, so it should have width and height set in either px or % in your .css file. |
| Creating a LatLng object | var mapCentre = new google.maps.LatLng(*latitude, longitude*);  The latitude and longitude values are reals. Example: latitude and longitude of the Eiffel Tower are 48.8584 and 2.2946. |
| Adding an event handler | google.maps.event.addListener(*mapObject*, ‘*eventName*’, function(event)  {  *callback(args); // this function is the event handler*  }); |
| Accessing the mouse location | The event object passed into the callback by the system has a field latLng which holds the latitude and longitude of the clicked location on the map. |
| Creating a marker object | var *markerObject* = new google.maps.Marker(*markerOptions*); |
| *markerOptions* | A JavaScript literal object that must contain, at a minimum, the fields   * position: a LatLng object where the marker is to be added * title: the tooltip string |
| Binding a marker object to a map object | *markerObject*.setMap(*mapObject*); |

An illustration of what your end application should look like:

