# 158.258 - Laboratory Practicals

# - Session 8 -

# Getting acquainted with Objects, maps and Location API, Canvas

# 2018

## Code to display a static Map

```
<input type='button' value="Show map centered on ME" onclick="getLocation()">
<p id='mapImg'> PUT MAP HERE</p>

<script>
function getLocation() {
                    if (navigator.geolocation)
        navigator.geolocation.watchPosition(showPosition);
                    else
        x.innerHTML = "Geolocation is not supported";
}

function showPosition(position) {
                    var latlon = position.coords.latitude + "," + position.coords.

                    var img_url = "http://maps.googleapis.com/maps/api/staticmap?c
        img_url +=      "&zoom=14&size=800x600&sensor=false";

    mapID = document.getElementById("mapImg")
    mapID.innerHTML = "<img src='"+img_url+"'>";
}
</script>
```

Show map centered on ME

PUT MAP HERE

## Page - 1: Convert the display static map example into a function

In order to do the following, convert the static map example in a way, that it would allow replacing the *innerHTML* of the item with *elementID* with the image of the map:

```
displayStaticMap(lat, long, element_ID)
```

## Page - 2: Create a page onto which you enter coordinates and it

## displays a map

Then create a page that has the following components

- Two text input boxes, **Latitude** and **Longitude**
- A 'Show Map' button

Once the button is clicked, display the map centred at those coordinates.

## Page- 3 : Create a List of coordinate objects

Extend the script of page1.html so that each time 'Save Map' is clicked (in addition to displaying the map), the coordinates be saved in an array named 'savedMapCoordinates'.

- Note that, each entry in **savedMapCoordinates** is an object that contains the two numbers.

Each time a coordinate is added to the map, display the list of coordinates at the bottom of the page.

## Page - 4: Create an object to manage the list of coordinates:

The object **mapMgr** should contain:

- **saveList** - a property that is a list of coordinates, like 'savedMapCoordinates' in page 3
- A method **add(lat, long)** that add the supplied coordinates to saveList
- A method **get(index)** - that turns an object with keys 'lat' & 'long' for saveList[index] or null
- A method **getcoordinateList()** that returns a string of coordinates separated by <*br*> so you can do something like

```
x = document.getElementId(displayID)
x.innerHTML(mapMgr.getcoordinateList())
```

>> Test it by writing explicit calls to each of the methods and display the results

## Page - 5: Redo page - 3 to use *mapMgr* to store the details

---

## Extensions

Add a new method

- **getClickablecoordinateList(callback-function)**
  Similar to *getCoordinateList()*, It returns each coordinate pair as a paragraph that has the calls to the callback function with parameters 'lat' and 'long' for the specified coordindate.

  >> The callback uses an *alert box* to display the **clicked on coordinates**.

- Center the map at the clicked coordinate List

## TEST

Create Coordinates

**REPLACE ME WITH A LIST OF COORDINATES**

## Center the map at the clicked coordinate List

# Part 2
# -Canvas-

## Page - 6 :Create a simple canvas

Add a *Canvas* to the page

- Add a *circles, a text* and a *images* to the canvas

  >> Make sure the canvas expense to the entire width and height of the screen

**Create animation by:**

- Resize the image

  >> Use a for loop to repeat the action more than once

- Add movement to the circles
- Add movement to the circles

—

. Edited 2018: Nazi Tabatabaei-Yazdi