

# CS275 Web and Mobile App Development

## HTML5

Material created by:  
David Augenblick, Bill Mongan, Dan Ziegler, Samantha Bewley, and Matt Burlick

# HTML5

- Now that we've seen HTML, CSS, and JavaScript, let's see how we can take advantage of some of HTML5's new APIs
- Google Maps example is taken from:

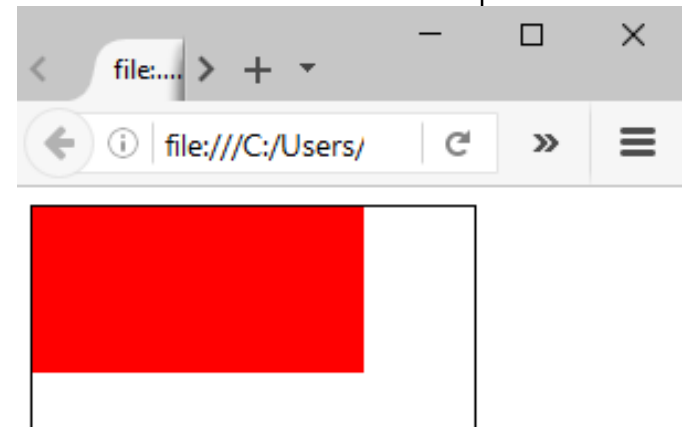
`https://developers.google.com/maps/documentation/javascript/examples/map-simple`

# HTML5 Canvas

- HTML5 provides a canvas object which we can draw on via JavaScript

```
<canvas id="myCanvas" width="200" height="100"
  style="border:1px solid #000000;">
</canvas>

<script>
  var c=document.getElementById("myCanvas");
  var ctx=c.getContext("2d");
  ctx.fillStyle="#FF0000";
  ctx.fillRect(0,0,150,75);
</script>
```



# HTML5 Geolocation

- HTML5 can also get our current geolocation from the browser/navigator.
- Note: the `getCurrentPosition` method takes two functions as parameters.
  - The first is called upon success
  - The second is called upon failure
- These are called **asynchronously** after it gets the position (or can't).

```
<html>
<body>
<div id=loc></div>
<script>
var x=document.getElementById("loc");
function showPosition(pos){
    x.innerHTML=pos.coords.latitude;
}
function error(err){
    x.innerHTML=err;
}
navigator.geolocation.getCurrentPosition(showPosition,error);
</script>
</body>
</html>
```

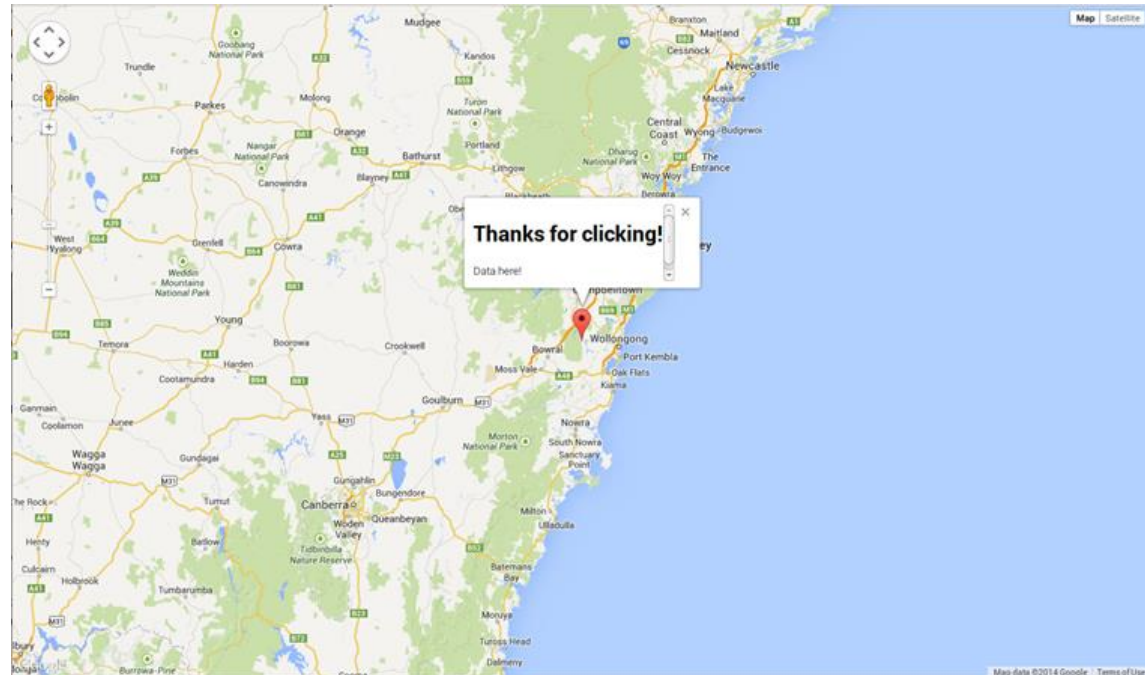
# HTML5 Geolocation

- In the previous slide we used geolocation to get our GPS fix from the browser.
- But we can also call `watchPosition()` instead of `getCurrentPosition()` to repeatedly call our `showPosition()` function with GPS updates.

# Rendering a Google Map

- Using a bunch of this stuff we can nicely, easily render a Google Map into our webpage!
- Let's go through this process...

```
<!-- Need a place for the map to go! -->  
<body>  
  <div id="map-canvas"></div>  
</body>
```



# Example: Rendering a Google Map

- Since our web page initially has nothing on it, technically it has zero size
  - And thus any rendering won't be seen.
- So let's start off the document, body and canvas as 100%

```
<style>
  html, body, #map-canvas {
    height: 100%;
  }
</style>
```

7

# Example: Rendering a Google Map

- Now we want to ask Google to generate a map and render it to the `map-canvas` object.
- We can also provide it some parameters.
- In order to do this we must reference the JavaScript code

```
<script  
  src="https://maps.googleapis.com/maps/api/js?key=YOURKEY&callback=initialize"  
  async defer>  
</script>
```

- Where `YOURKEY` is a unique key provided by Google to identify you.
  - You can request one here: <https://developers.google.com/maps/documentation/javascript/get-api-key>
- And `initialize` is the callback function we must create to create the map object (next slide).



# Example: Rendering a Google Map

```
<script>
  function initialize() {
    var mapOptions= {  zoom: 8, center: {lat: -34.397, lng: 150.644}};
    var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
  }
</script>
```

# Example: Rendering a Google Map

- We could do other fun stuff with this too!
- Let's put a marker at this location on the map...

```
var map;  
var myLatLng = {lat: -34.397, lng: 150.644};  
function initialize() {  
    var mapOptions= { zoom: 8, center: myLatLng};  
    map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);  
  
    var marker = new google.maps.Marker({  
        position: myLatLng,  
        map: map,  
        title: 'Hello World!'  
    });  
}
```



# Example: Rendering a Google Map

- And how about popping up a `div` when we click on it!

```
<script>
  var map;
  var myLatLng = {lat: -34.397, lng: 150.644};
  function initialize() {
    var mapOptions= {  zoom: 8, center: myLatLng};
    map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

    var marker = new google.maps.Marker({
      position: myLatLng,
      map: map,
      title: 'Hello World!'
    });

    var contentString = '<div id="content">'+
      '<div id="siteNotice"></div>'+
      '<h1 id="firstHeading" class="firstHeading">Thanks for clicking!</h1>'+
      '<div id="bodyContent">'+
      '<p>Data here!</p>'+
      '</div>'+
      '</div>';

    var infowindow = new google.maps.InfoWindow({
      content: contentString
    });

    google.maps.event.addListener(marker, 'click', function() {
      infowindow.open(map,marker);
    });
  }
</script>
```

