KEY IN-CLASS STUDENT ACTIVITY 4.6: CODE ADDITIONAL HTML5 APIs.

Lesson Objective 4.6:

Code additional HTML5 APIs.

Resources, software, and additional files needed for this lesson:

1. Microsoft® Expression® Web 4 (preferred) or Notepad (Notepad++ is preferred; it provides real-time feedback for JavaScript).
2. Internet access.

Guiding questions:

1. **List uses for an API that determines your current location.** This information can be used to provide navigation information, the distance to another location, or location and directions to local businesses. There are many uses for this information in web design.
2. **What is the purpose of a web worker?** The primary function of a web worker is to allow the web page to run multiple tasks at one time. The tasks are spawned and run in the background.
3. **What is a WebSocket?** A WebSocket is a new JavaScript API and protocol that allows for two-way communication from the web server to the client.

Student activity:

Directions to the student:

Read the following scenario. Create a one-page website to demonstrate the use of the Geolocation API.

Scenario:

Your friend, Wendy Teo, is training to run a 5-kilometer race next month to raise awareness of cystic fibrosis. Each day she runs a loop near her home, but she is not sure of the exact distance of the loop. It is vital to her training routine that she runs a specific distance each day.

To help Wendy figure out how far she is running in her practice sessions, you volunteer to create a web application that calculates the distance travelled. The web page will take advantage of the new HTML5 Geolocation API to calculate the distance travelled.

Content:

1. Create an HTML page.
2. Refer to the tutorial on HTML5 ROCKS for instructions on how to include a trip meter: *http://www.html5rocks.com/en/tutorials/geolocation/trip_meter/*
3. If possible, publish your web page and access it from a mobile device (tablet or smartphone). Move to a different location. Check to see how far you travelled.

Content:

1. Create an HTML page.
2. Refer to the tutorial on HTML5 ROCKS for instructions on how to include a trip meter: *http://www.html5rocks.com/en/tutorials/geolocation/trip_meter/*
3. If possible, publish your web page and access it from a mobile device (tablet or smartphone). Move to a different location. Check to see how far you travelled.

Answers will vary, but they should be similar to the finished example from HTML5 Rocks:

```
<!DOCTYPE html>
   <html>
   <head>
   <style>
   #tripmeter {
   border: 3px double black;
   padding: 10px;
   margin: 10px 0;}
   p {
   color: #222;
   font: 14px Arial; }
   span {
   color: #00C; }
   </style>
   </head>
   <body>
   <div id="tripmeter">
   <p>
   Starting Location (lat, lon):<br/>
   <span id="startLat">???</span>&deg;, <span
   id="startLon">???</span>&deg;
   </p>
   <p>
   Current Location (lat, lon):<br/>
   <span id="currentLat">???</span>&deg;, <span
   id="currentLon">???</span>&deg;
   </p>
   <p>
   Distance from starting location:<br/>
   <span id="distance">0</span> km
   </p>
   </div>
   <script>
   window.onload = function() {
   var startPos;
   if (navigator.geolocation) {
   navigator.geolocation.getCurrentPosition(function(position) {
   startPos = position;
   document.getElementById("startLat").innerHTML =
   startPos.coords.latitude;
```

```
document.getElementById("startLon").innerHTML =
startPos.coords.longitude;
}, function(error) {
alert("Error occurred. Error code: " + error.code);
// error.code can be:
// 0: unknown error
// 1: permission denied
// 2: position unavailable (error response from locaton provider)
// 3: timed out
});

navigator.geolocation.watchPosition(function(position) {
document.getElementById("currentLat").innerHTML =
position.coords.latitude;
document.getElementById("currentLon").innerHTML =
position.coords.longitude;
document.getElementById("distance").innerHTML =
calculateDistance(startPos.coords.latitude, startPos.coords.longitude,
position.coords.latitude, position.coords.longitude);
});
}
};

// Reused code - copyright Moveable Type Scripts - retrieved May 4,
2010.
// http://www.movable-type.co.uk/scripts/latlong.html
// Under Creative Commons License
http://creativecommons.org/licenses/by/3.0/
function calculateDistance(lat1, lon1, lat2, lon2) {
var R = 6371; // km
var dLat = (lat2-lat1).toRad();
var dLon = (lon2-lon1).toRad();
var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(lat1.toRad()) * Math.cos(lat2.toRad()) *
Math.sin(dLon/2) * Math.sin(dLon/2);
var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
var d = R * c;
return d;
}
Number.prototype.toRad = function() {
return this * Math.PI / 180;
}
</script>
</body>
</html>
```