158.258 Web Development

HTML5 Geolocation API

Computer Science & Information Technology

Massey University (AKLI, DISD & MTUI)

Geolocation

W3C Geolocation API The W3C Geolocation API is an effort by the World Wide Web Consortium (W3C) to standardize an interface to retrieve the geographical location information for a client-side device.

Geolocation API is ideally suited to web applications for mobile devices.

Some of what the Geolocation API can provide:

- latitude
- longitude (coordinates
- altitude (height
- and accuracy of the position gathered.

How? - via the *Geolocation Object* – **`navigator.geolocation`**

How to access to the location?

```
    <script>
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    }
    else {
        var x = document.getElementById("locationOutput");
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}

function showPosition(position) {
    var x = document.getElementById("locationOutput");
    x.innerHTML = "Latitude: " + position.coords.latitude +
              "<br>Longitude: " + position.coords.longitude;
}
</script>

<input type='button' value='Show Location' onclick="getLocation()"><br>
LOCATION: <p id='locationOutput'> Not yet known</p>
```

Show Location

LOCATION:

Not yet known

http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_geolocation

Synchronous vs. Asynchronous program design

e.g. If you're wanting to get the location, you could do this:

```
    currentLocation = geolocation.getCurrentPosition(); # NOT VALID
lat  = currentLocation.coords.latitude;
long = currentLocation.coords.longitude;
```

which seems reasonable.

However things aren't done this way!

Instead, call looks like:

```
    navigator.geolocation.getCurrentPosition(showPosition);
```

where showPosition is the **function name** (NOT call) of a function

Local vs. Remote Data

We're used to thinking of our program data being available

```
x = 7
y = 100 * x        # We expect x to be immediately available ...
```

Geolocation data is not (necessarily) local

If

- `geolocation.getCurrentPosition()` accessed local data → ALL GOOD

BUT: Potential problems if it needs to access remote data

- currentLocation = geolocation.getCurrentPosition()
    - *it's a **SYNCHRONOUS** call (in the same timeline)*
    - *doesn't return until it has the required data*
    - *everything else stops until it returns → Interface is unresponsive*

Callback functions and event-driven programming

A way to handle data that *might* be available is to:

1. Send a request, along with who should handle the response (a handler function)
2. Carry on working

IF the data arrives, the handler function will do something with the data

The same mechanism can be used for events that might happen:

- a button click
- a new request arriving for some data from a server
- a reply/response for some data we need (e.g. location)

The event handler is called a *callback function*

- It will do something appropriate with the data

A **callback function** is *supplied as a parameter*

- it's used to specify what *should happen* when/if an event occurs
- **IMPORTANT** - use the function's **name** - without ()

Callback - annotated

```
 1. <input type='button' value='Show Location' onclick="getLocation()"><br>
 2.                                                     CALLBACK #1
 3. LOCATION: <p id='locationOutput'> Not yet known</p>
 4.
 5. <script>
 6. let x = document.getElementById("locationOutput");
 7. ----------------------------------------------------------------------
 8. function showPosition(position) {   <<< THIS IS THE CALLBACK FUNCTION
 9.     x.innerHTML = "Latitude: " + position.coords.latitude +
10.                   "Longitude: " + position.coords.longitude;
11. }
12. ----------------------------------------------------------------------
13. function getLocation() {        <<<<< STARTS THE REQUEST FOR THE LOCATION
14.     if (navigator.geolocation)
15.         navigator.geolocation.getCurrentPosition(showPosition);
16.                                                   CALLBACK #2
17.     else
18.         x.innerHTML = "Geolocation is not supported by this browser.";
19. } --------------------------------------------------------------------
</script>
```

NOTE: Line **#8** - when showPosition is (eventually) called,

- **getCurrentPosition()** will be supplied with the *position* parameter

SO: the callback (**showPosition**) must have the correct number of parameters supplied.

getCurrentPosition()

The getCurrentPosition() method returns a **position** object

- as used by the **showPosition(position)** function.

position has the following attributes:

`coords.latitude`	— latitude as a decimal number (always returned)
`coords.longitude`	— longitude as a decimal number (always returned)
`coords.accuracy`	— accuracy of position (always returned)
`coords.altitude`	— height in meters above mean sea level, if available
`coords.altitudeAccuracy`	— altitude accuracy, if available
`coords.heading`	— heading as degrees clockwise from North, if available
`coords.speed`	— speed in meters per second, if available
`timestamp`	— date/time of the response, if available

Handling Geolocation failures

Requesting the location won't always work, even if the browser supports HTML5

the user may have disabled geolocation requests

the GPS won't work (inside) and no Wifi

To handle this

- `getCurrentPosition(functionOnSuccess)`

can be extended:

- `getCurrentPosition(functionOnSuccess, functionOnError)`

The `functionOnError()` call will include an *error* object that has a *code* property that indicates what went wrong e.g. look in `error.code`

Adding failure handling

```
    function locationError(error) {
    switch(error.code) {
        case error.PERMISSION DENIED:
            x.innerHTML = "Location Request: User has disabled Geolocation."
            break;
        case error.POSITION UNAVAILABLE:
            x.innerHTML = "Location Request: info not available"
            break;
        case error.TIMEOUT:
            x.innerHTML = "Location request: timed out."
            break;
        case error.UNKNOWN ERROR:
            x.innerHTML = "Location Request: unknown error"
            break;
    }
}
```

Extended location demo including error handling

```
    <input type='button' value='Show Location'  onclick="getLocation()">
LOCATION: <p id='locationOutput'> Not yet known</p>
<script>
var x = document.getElementById("locationOutput");

function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
              "<br>Longitude: " + position.coords.longitude;
}

function locationError(error) {
    switch(error.code) {
        case error.PERMISSION DENIED:
            x.innerHTML = "Location Request: User has disabled geolocation."
            break;
        case error.POSITION UNAVAILABLE:
            x.innerHTML = "Location Request: info not available"
            break;
        case error.TIMEOUT:
            x.innerHTML = "Location request: timed out."
            break;
        case error.UNKNOWN ERROR:
            x.innerHTML = "Location Request: unknown error"
            break;
    }
}

function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition, locationError);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
</script>
```

Show Location

LOCATION:

Not yet known

Other methods of Geolocation Object

The Geolocation object also has other interesting methods:

`watchPosition()`

Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car).

`clearWatch()`

Stops the `watchPosition()` method.

Display a location in a Static map

If you just want a map image, a Google API Key isn't required

```
    <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.longitude;

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

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

Click the button to get map of your position.

Show map centered on ME

PUT MAP HERE

http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_geolocation_map