The Geolocation API

Let's start with an example!

Click the button to get your position and display a map as a picture. This may take some time, or fail, if a geolocation is not available with the device and connection you are using (e.g. at work, beyond a proxy). Try it anyway!

This example will be explained later on in the course...

Try It

INTRODUCTION

This chapter presents the new Geolocation API and illustrates its use with several examples.

The Geolocation HTML5 JavaScript API is implemented by most modern Web browsers, and uses different means to get the current location: GPS, GSM/3G triangulation, Wifi, IP address, etc.

It is possible to prompt the user to activate the GPS (this is what most GPS navigation software does on mobile phones), or ask for a particular mean among those available. It is also possible to track the current position when it changes. This is useful for writing a navigation application or for tracking in real time the position of different participants in the case of an application that involves several persons at the same time (using WebSockets, for example).

CURRENT SUPPORT IS EXCELLENT

As at June 2015, support for this API is excellent, both on mobile and on desktop devices.

To get an updated table, check caniuse.com

TYPICAL USE

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

function showPosition(position) {
    console.log("latitude is: " +position.coords.latitude);
    console.log("longitude is: " +position.coords.longitude);
}

function onerror(err) {
    console.log("Could not get the position");
}
```

This online example at JS Bin shows how to get the current longitude and latitude and

display them in an HTML page. Try it below in your browser:

Click the button to get your coordinates:

Where am I ?

Note that the first time you execute this example, for privacy reasons, the browser will ask if you agree to share your position with the application.

Source code of this typical example:

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <title>Basic example of use of the geolocation API</title>
 </head>
<body>
<p id="msg">Click the button to get your coordinates:</p>
<button onclick="getLocation()">Where am I ?</button>
<script>
    vardisplayCoords=document.getElementById("msg");
    function getLocation() {
       if (navigator.geolocation) {

 navigator.geolocation.getCurrentPosition(showPosition);
       } else {
          displayCoords.innerHTML="Geolocation API not
supported by your browser.";
       }
   }

   function showPosition(position) {
       displayCoords.innerHTML="Latitude:
" + position.coords.latitude +
                                "<br />Longitude:
```

```
" +position.coords.longitude;
    }
</script>
</body>
</html>
```

```
function getLocation(){
    if (navigator.geolocation){
        navigator.geolocation.getCurrentPosition(showPosition);
    } else{
        x.innerHTML="Geolocation not supported by your browser.";
    }
}

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

Explanations:

- *Line 14* checks if the Web browser supports the geolocation API by testing the variable `navigator.geolocation`. If not `null`, then the geolocation API is supported.
- *Line 15* calls`navigator.geolocation.getCurrentPosition(showPosition)` passing a callback function as a parameter (in this example we did not specify a callback in case of error). **When a current position is available, the callback function will be called asynchronously, and the input parameter of this callback function will be the current position,** like in the function `showPosition(position)` of the example.
- *Line 22:* the `position` objects has a `coords` property that is the object that holds the `longitude` and the `latitude`.

EXTERNAL RESOURCES:

- The W3C specification about the geolocation API (in Recommendation status, aka, Web standard)
- Good article from the Opera dev Web site

- Excellent tutorial on the Google Maps API. Some examples in this part of the course are illustrated using Google Maps.