

Chapter 71: Geolocation

Section 71.1: Get updates when a user's location changes

You can also receive regular updates of the user's location; for example, as they move around while using a mobile device. Location tracking over time can be very sensitive, so be sure to explain to the user ahead of time why you're requesting this permission and how you'll use the data.

```
if (navigator.geolocation) {  
    //after the user indicates that they want to turn on continuous location-tracking  
    var watchId = navigator.geolocation.watchPosition(updateLocation, geolocationFailure);  
} else {  
    console.log("Geolocation is not supported by this browser.");  
}  
  
var updateLocation = function(position) {  
    console.log("New position at: " + position.coords.latitude + ", " + position.coords.longitude);  
};
```

To turn off continuous updates:

```
navigator.geolocation.clearWatch(watchId);
```

Section 71.2: Get a user's latitude and longitude

```
if (navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(geolocationSuccess, geolocationFailure);  
} else {  
    console.log("Geolocation is not supported by this browser.");  
}  
  
// Function that will be called if the query succeeds  
var geolocationSuccess = function(pos) {  
    console.log("Your location is " + pos.coords.latitude + "°, " + pos.coords.longitude + "°.");  
};  
  
// Function that will be called if the query fails  
var geolocationFailure = function(err) {  
    console.log("ERROR (" + err.code + "): " + err.message);  
};
```

Section 71.3: More descriptive error codes

In the event that geolocation fails, your callback function will receive a `PositionError` object. The object will include an attribute named `code` that will have a value of 1, 2, or 3. Each of these numbers signifies a different kind of error; the `getErrorCode()` function below takes the `PositionError.code` as its only argument and returns a string with the name of the error that occurred.

```
var getErrorCode = function(err) {  
    switch (err.code) {  
        case err.PERMISSION_DENIED:  
            return "PERMISSION_DENIED";  
        case err.POSITION_UNAVAILABLE:  
            return "POSITION_UNAVAILABLE";  
        case err.TIMEOUT:  
            return "TIMEOUT";  
    }  
}
```

```
    default:
        return "UNKNOWN_ERROR";
    }
};
```

It can be used in `geolocationFailure()` like so:

```
var geolocationFailure = function(err) {
    console.log("ERROR (" + getErrorCode(err) + "): " + err.message);
};
```