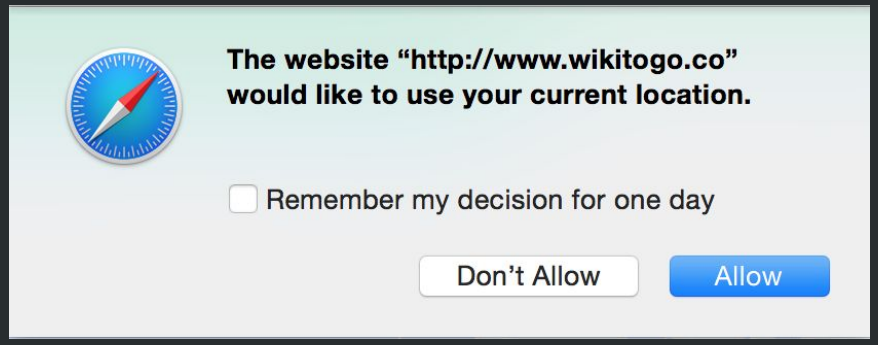
Geolocation

Where are you now?

What is it?

The Geolocation API lets you locate and track users, with their permission. The API is device agnostic - it doesn't matter how the browser gets your location, as long as the client can request and receive the data in a standard way.



Official spec: http://dev.w3.org/geo/api/

IRON HACK

How do you use it?

The API is published through the **navigator.geolocation** object. If the object exists, then geolocation is supported.

```
if ("geolocation" in navigator) {
  console.log("geolocation is available!")
} else {
  console.log("geolocation IS NOT available")
}
```

The **getCurrentPosition()** function takes three parameters:

```
navigator.geolocation.getCurrentPosition(onLocation, onError, options);
```

getCurrentPosition()

Set a success handler (what to do when the data is successfully received)

```
function onLocation (position) {
  console.log('Your latitude is ' + position.coords.latitude);
  console.log('Your longitude is ' + position.coords.longitude);
  document.getElementById('position').innerHTML = 'Lat: ' +
  position.coords.latitude + ', lon:' + position.coords.longitude
}
```

Set an error handler (what to do in case of an error)

```
function onError (error) {
  // Delivers an error object with information about the error
  console.error(error);
}
```

getCurrentPosition()

This will work with just the two callback functions, but we can pass some additional options.

getCurrentPosition()

We pass these options in an object, typically called “options”

getCurrentPosition()

We can create an options object like this:

```
var options = {
  enableHighAccuracy: true  // retrieve more accurate position, takes longer
};
```

The other options include:

- timeout: <amount in milliseconds>
- maximumAge: <amount in milliseconds>

Callbacks

Functions are objects.

You can pass a function as an argument.

A callback function is a function that you pass as an argument to another function to be executed later.

In our example, **onLocation** and **onSuccess** are callback functions that we passed to the **getCurrentPosition** function.

More about callbacks: http://javascriptissexy.com/understand-javascript-callback-functions-and-use-them/

watchPosition()

This function lets us stalk the user (with their permission, of course!). It also takes three parameters:

```
// Calls onWatch multiple times as position changes
var watchId = navigator.geolocation.watchPosition(onWatch, onError, watchOptions);
```

It calls the success handler (in our case, **onWatch**) every time the position of the device changes. It also returns a watch ID, which you can use to stop watching:

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

Geolocation exercise - Iteration 1

1. Get the browser's current location.
2. Construct a URL to show the location on a static map from Google maps, using the static map API.
3. Display the map by adding it as an image to your HTML.

Hint to get you started

Solution