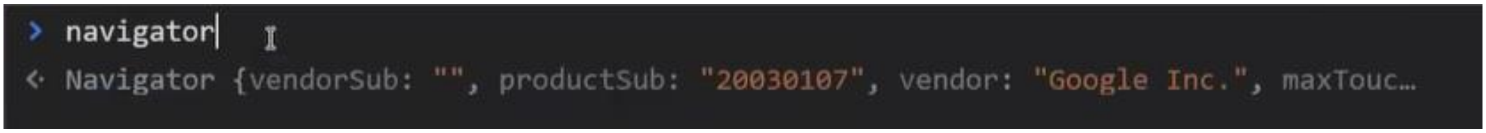
Geo – Location

Geo-Location:

HTML5 related API’s – https://developer.mozilla.org/en-US/docs/Web/API

HTML5 Geo-Location is a way to get user location. If we have the latitude and longitude, we can pin point the user anywhere in the world. In order to get the location, there is an object; like we have document related to HTML, there is an object called navigator, attached to the window object, which has everything related to Geo-Location.



```
> navigator
<· Navigator {vendorSub: "", productSub: "20030107", vendor: "Google Inc.", maxTouc…
```

If we expand onto it, we can see all the properties. One of the properties is geolocation which we will use to know the location of the user. Now, in the HTML file, we will create a button element, “Get Location” with an id="getLcnBtn"

```
<button id="getLcnBtn">Get Location</button>
```

The first step is to get the reference. In JavaScript,

```
var getLcnBtn = document.getElementById('getLcnBtn')
function getUserLocation() {

}
getLcnBtn.addEventListener('click', getUserLocation)
```

in the console, there is a navigator.geolocation, this is the Geolocation object which exists on the navigator object. Bascically, there are two objects one is navigator and other is Geolocation object. In the Geolocation, if we expand it, we get clearWatch, getCurrentPosition and watchPosition.

Going back to javascript,

```
    navigator.geolocation.getCurrentPosition()
```

3 Parameters –

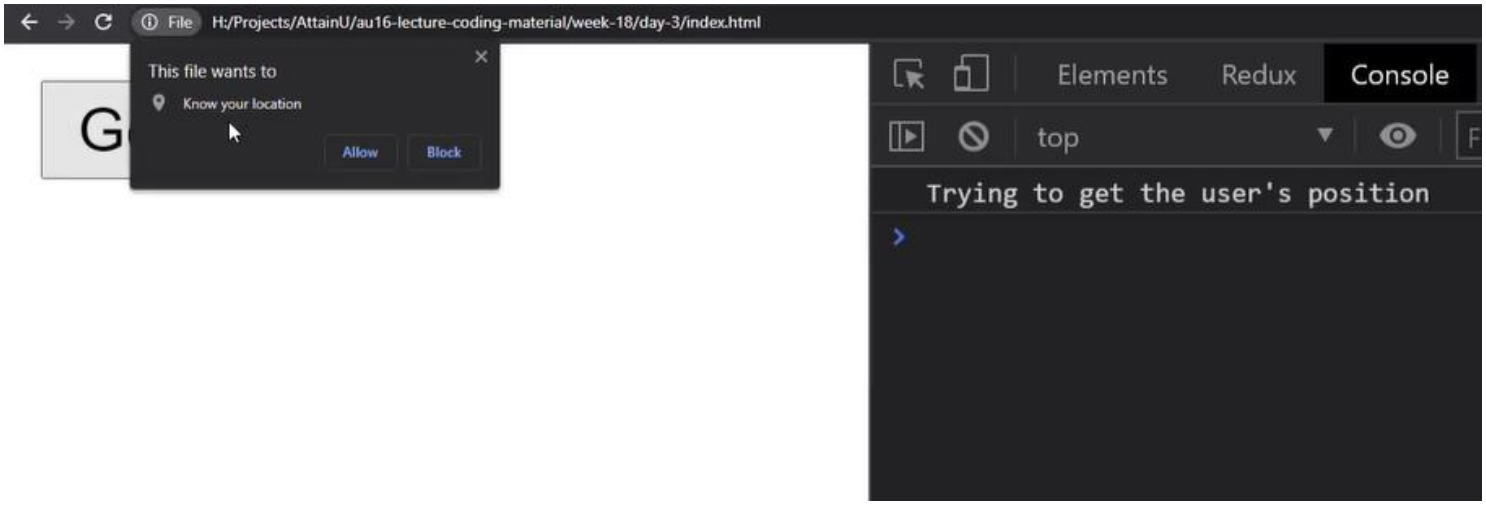
- Function (should be a function, mandatory)
- Function
- Position related object feature

Like we have passed a function in addEventListener, we need to add a function to getCurrentPosition(). So,

```
var getLcnBtn = document.getElementById('getLcnBtn')
function onPositionReceived() {
     console.log('Position Received')
}
function getUserLocation() {
     console.log('trying to get the users's position')
     navigator.geolocation.getCurrentPosition(onPositionReceived)
}

getLcnBtn.addEventListener('click', getUserLocation)
```

Expected behavior is when we click on the getLcnButton, which is from the HTML page, JavaScript will call function getUserLocation(). When the getUserLocation() function is called it would first log the line, console.log("Trying to get the user's position") after than it will call getCurrentPosition() function. this will create a popup asking to confirm or share the position. Once we allow that to happen, then JavaScript will call on PositionReceived() function.


H:/Projects/AttainU/au16-lecture-coding-material/week-18/day-3/index.html

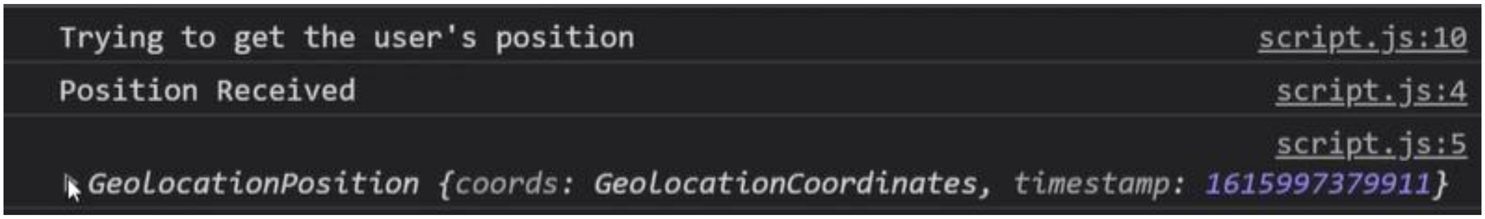
This file wants to

Know your location

```
Trying to get the user's position
```

We can clearly see that unless we have not given permission, javascript will not be able to share our location.

When a event listener handler (getUserLocation) is called by an addEventListener method, it get passed on event object and all the information related gets attached on to it. Similarly, getCurrentPosition passes one object to the function we have provided. So in onPositionRecevied(), it get one more object – position. When the user gives permission, getCurrentPosition will send an object to the onPositionRecevied.



```
Trying to get the user's position                                   script.js:10
Position Received                                                    script.js:4
                                                                     script.js:5
▸ GeoLocationPosition {coords: GeolocationCoordinates, timestamp: 1615997379911}
```

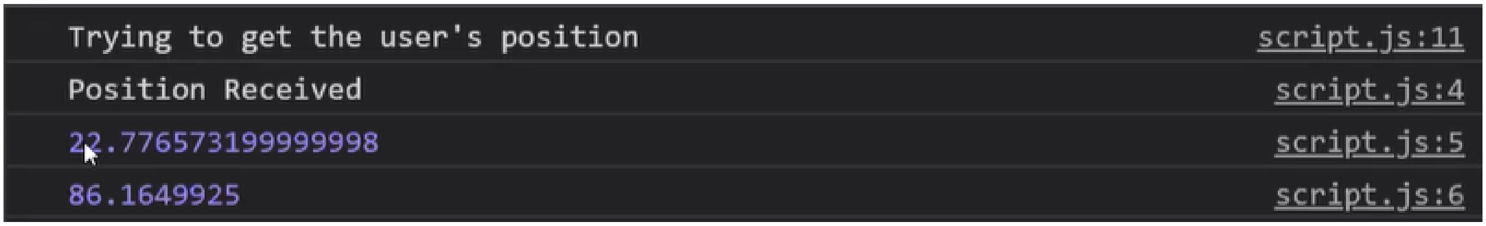
We can see the coords, and timestamp property. In “coords”, we will get the latitude and longitude values. To access these values, we can write,

```
var getLocationBtn = document.getElementById('getLocationBtn')

function onPositionReceived() {
     console.log('Position Received')
     console.log(position.coords.latitude)
     console.log(position.coords.longitude)
}

function getUserLocation() {
     console.log('trying to get the user location')
     navigator.geolocation.getCurrentPosition(onPositionReceived)
}

getLocationBtn.addEventListener('click', getUserLocation)
```



```
Trying to get the user's position                                   script.js:11
Position Received                                                    script.js:4
22.776573199999998                                                   script.js:5
86.1649925                                                           script.js:6
```

It might not give the accurate location in desktops, but if the device has GPS, then it would show the exact location of the user.

In the coords property, we have other properties like, speed, accuracy, heading are some properties that google uses to track your location, speed, heading (direction) and plot it. We can create our own Google Maps with these details.

The getCurrentPosition() accepts one parameter and the onPositionReceived() function gets called only when user gives position. We will create another function, onPositionDenied().

```
function onPositionDenied() {
     console.log('Position Denied!!!)}
```

This function gets called only when use click block on the confirmation popup.

Like, getCurrentPosition, there is another function as watchPosition(). This is a bit complicated. It continuously gets the current position the user.