DAY 7: HTML APIs

**HTML Geolocation API:**

The HTML Geolocation API is used to get the geographical position of a user. Since this can compromise privacy, the position is not available unless the user approves it.

Geolocation is most accurate for devices with GPS, like smartphones.

**Using HTML Geolocation:**

The **getCurrentPosition**() method is used to return the user's position.

*Example:*

|  |  |
| --- | --- |
| HTML Code | Output |
| *<!DOCTYPE html>*  *<html lang="en">*  *<head>*  *<title>Geolocation</title>*  *</head>*  *<body>*  *<p>Click the button to get your coordinates</p>*  *<button onclick="getLocation()">Click Here</button>*  *<p id="demo"></p>*  *<script>*  *var x = document.getElementById("demo")*  *function getLocation() {*  *if (navigator.geolocation) {*  *navigator.geolocation.getCurrentPosition(showPosition);*  *}*  *else {*  *x.innerHTML = "Geolocation is not supported";*  *}*  *}*  *function showPosition(position) {*  *x.innerHTML = "Latitude: " + position.coords.latitude + "<br> Longitude: " + position.coords.longitude;*  *}*  *</script>*  *</body>*  *</html>* | Click the button to get your coordinates  Click Here  Click Here  Latitude: 8.5555187 Longitude: 76.8836149 |

Check if Geolocation is supported If supported, run the **getCurrentPosition**() method. If not, display a message to the user.

If the **getCurrentPosition**() method is successful, it returns a coordinates object to the function specified in the parameter (**showPosition**).

The **showPosition**() function outputs the Latitude and Longitude.

**Handling Errors and Rejections:**

*Example:*

|  |
| --- |
| *<html lang="en">*  *<head>*  *<title>Geolocation Error</title>*  *</head>*  *<body>*  *<p>Click the button to get your coordinates</p>*  *<button onclick="getLocation()">Click Here</button>*  *<p id="demo"></p>*  *<script>*  *var x = document.getElementById("demo")*  *function getLocation() {*  *if (navigator.geolocation) {*  *navigator.geolocation.getCurrentPosition(showPosition, showError);*  *}*  *else {*  *x.innerHTML = "Geolocation is not supported";*  *}*  *}*  *function showPosition(position) {*  *x.innerHTML = "Latitude: " + position.coords.latitude + "<br> Longitude: " + position.coords.longitude;*  *}*  *function showError(error) {*  *switch (error.code) {*  *case error.PERMISSION\_DENIED:*  *x.innerHTML = "User denied the request for Geolocation."*  *break;*  *case error.POSITION\_UNAVAILABLE:*  *x.innerHTML = "Location information is unavailable."*  *break;*  *case error.TIMEOUT:*  *x.innerHTML = "The request to get user location timed out."*  *break;*  *case error.UNKNOWN\_ERROR:*  *x.innerHTML = "An unknown error occurred."*  *break;*  *}*  *}*  *</script>*  *</body>*  *</html>* |

The second parameter of the **getCurrentPosition**() method is used to handle errors. It specifies a function to run if it fails to get the user's location.

**The getCurrentPosition() Method - Return Data:**

The **getCurrentPosition**() method returns an object on success. The latitude, longitude and accuracy properties are always returned. The other properties are returned if available.

|  |  |
| --- | --- |
| **Property** | **Returns** |
| coords.latitude | The latitude as a decimal number (always returned) |
| coords.longitude | The longitude as a decimal number (always returned) |
| coords.accuracy | The accuracy of position (always returned) |
| coords.altitude | The altitude in meters above the mean sea level (returned if available) |
| coords.altitudeAccuracy | The altitude accuracy of position (returned if available) |
| coords.heading | The heading as degrees clockwise from North (returned if available) |
| coords.speed | The speed in meters per second (returned if available) |
| timestamp | The date/time of the response (returned if available) |

**Geolocation Object - Other Interesting Methods:**

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.

The example below shows the **watchPosition**() method. You need an accurate GPS device to test this (like smartphone).

|  |
| --- |
| *<button onclick="getLocation()">Click Here</button>*  *<p id="demo"></p>*  *<script>*  *var x = document.getElementById("demo")*  *function getLocation() {*  *if (navigator.geolocation) {*  *navigator.geolocation.watchPosition(showPosition);*  *}*  *else {*  *x.innerHTML = "Geolocation is not supported";*  *}*  *}*  *function showPosition(position) {*  *x.innerHTML = "Latitude: " + position.coords.latitude + "<br> Longitude: " + position.coords.longitude;*  *}*  *</script>*  *</body>* |

**HTML Drag and Drop**

In HTML, any element can be dragged and dropped.

**Drag and Drop:**

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.

*Example:*

|  |
| --- |
| *<!DOCTYPE html>*  *<html lang="en">*  *<head>*  *<title>Drag and Drop</title>*  *<style>*  *#div1 {*  *width: 350px;*  *height: 70px;*  *padding: 10px;*  *border: 1px solid #aaaaaa;*  *}*  *</style>*  *<script>*  *function allowDrop(event) {*  *event.preventDefault();*  *}*  *function drag(event) {*  *event.dataTransfer.setData("text", event.target.id);*  *}*  *function drop(event) {*  *event.preventDefault();*  *var data = event.dataTransfer.getData("text");*  *event.target.appendChild(document.getElementById(data));*  *}*  *</script>*  *</head>*  *<body>*  *<p>Drag the W3Schools image into the rectangle:</p>*  *<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>*  *<img id="drag1" src="https://www.w3schools.com/html/img\_logo.gif" draggable="true" ondragstart="drag(event)"*  *width="336" height="68">*  *<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>*  *</body>*  *</html>* |