JavaScript

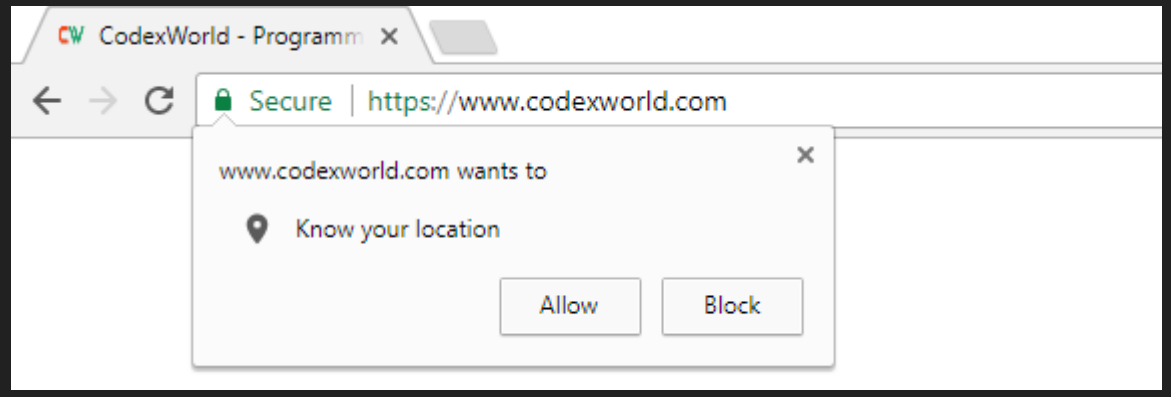
Geolocation API – 김근형 강사

Geolocation API

- Geolocation API
  - Geolocation API 는 위도와 경도 같은 현재 위치 정보를 검색하는 API 이다.

Geolocation API

- Geolocation API
  - Geolocation API 는 W3C 에서 독자적으로 제정한 명세이다.
  - 모바일 웹으로 위치정보를 이용하는 고객들이 늘면서 점점 수요가 증가하고 이미 많은 곳에서 사용하고 있는 기능이다.
  - 이 기능을 사용하기 위해서는 반드시 개인 정보의 사용 허가를 해야 한다. 그래서 Geolocation API 를 구현한 브라우저는 반드시 다음과 같은 확인창을 표시한다.
  - Geolocation API가 사용하는 측지계는 WGS 84 이다.



Geolocation API

- Geolocation API
  - 중요한 Geolocation 정보는 window.navigator.geolocation 을 통해 얻을 수 있으며 현재 있는 위치정보를 얻기 위해서는 getCurrentPosition() 을 통해 위치 정보를 얻어와야 한다.
  - 이 함수는 위치 정보를 얻었을 때 실행할 콜백 함수의 객체를 매개변수로 넘겨준다.
  - Geolocation API는 순간적으로 위치 정보를 얻는 것이 아니라 비동기적으로 위치 정보를 처리한다.
  - 따라서 getCurrentPosition() 함수는 콜백함수를 매개변수로 지정하게 된다.

Geolocation API

- getCurrentPosition() 이벤트 속성
  - 콜백 함수에서 나오는 event 에서 위치 정보를 나타내는 다양한 값이 있다.

위치 정보를 나타내는 값	설명
event.coords.latitude	위도를 수치로 반환한다. 단위는 도
event.coords.longitude	경도를 수치로 반환한다. 단위는 도
event.coords.accuracy	위도 및 경도의 정밀도를 반환한다. 단위는 미터
event.coords.altitude	GPS 고도를 반환한다. 단위는 미터이며 값을 얻을 수 없으면 null 을 반환한다.
event.coords.altitudeAccuracy	GPS 고도의 정밀도를 반환한다. 단위는 미터이며 값을 얻을 수 없으면 null 을 반환한다.
event.coords.heading	진행 방향을 0부터 360 사이의 수치로 반환한다. 이 값은 각도를 나타내며 단위는 도이다. 북쪽을 0도로 시계방향으로 증가한다. 따라서 동쪽은 90도, 서쪽은 270도 이다. 하지만 이 값을 얻을 수 있을 때는 이동 중에만 얻을 수 있으며 정지중 일 때, 즉 event.cords.speed 가 0이면 NaN 을 반환한다. 값을 얻을 수 없을 때는 null 을 반환한다.
event.coords.speed	이동 속도를 수치로 반환한다. 단위는 m/s 이다. 이 값을 얻을 수 없을 때는 null 을 반환한다.
event.timestamp	위치 정보를 얻었을 때의 시간을 나타내는 Date 객체를 반환한다.

Geolocation API

- getCurrentPosition() 예제 - 1

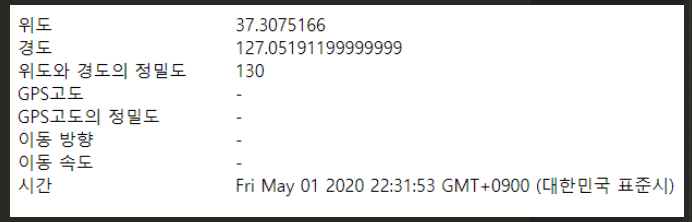
```
<dl>
    <dt>위도 </dt>
    <dd id="latitude">-</dd>
    <dt>경도 </dt>
    <dd id="longitude">-</dd>
</dl>
<script type="text/javascript">
document.addEventListener("DOMContentLoaded", function() {
    // 현 위치 정보 얻기
    window.navigator.geolocation.getCurrentPosition(show_location);
}, false);
// 위치정보를 얻은 후의 처리
function show_location(event) {
    // 위도
    var latitude = event.coords.latitude;
    document.querySelector('#latitude').textContent = latitude;
    // 경도
    var longitude = event.coords.longitude;
    document.querySelector('#longitude').textContent = longitude;
}
</script>
```

위도 37.3075166
경도 127.05191199999999

Geolocation API

- getCurrentPosition() 예제 - 2

```
<dl>
    <dt>위도 </dt>
    <dd id="latitude">-</dd>
    <dt>경도 </dt>
    <dd id="longitude">-</dd>
    <dt>위도와 경도의 정밀도 </dt>
    <dd id="accuracy">-</dd>
    <dt>GPS고도 </dt>
    <dd id="altitude">-</dd>
    <dt>GPS고도의 정밀도 </dt>
    <dd id="altitudeAccuracy">-</dd>
    <dt>이동 방향 </dt>
    <dd id="heading">-</dd>
    <dt>이동 속도 </dt>
    <dd id="speed">-</dd>
    <dt>시간 </dt>
    <dd id="timestamp">-</dd>
</dl>
<script type="text/javascript">
document.addEventListener("DOMContentLoaded", function() {
    // 현재 위치 얻기
    window.navigator.geolocation.getCurrentPosition(show_location);
}, false);
```



위도	37.3075166
경도	127.05191199999999
위도와 경도의 정밀도	130
GPS고도	-
GPS고도의 정밀도	-
이동 방향	-
이동 속도	-
시간	Fri May 01 2020 22:31:53 GMT+0900 (대한민국 표준시)

```
// 위치정보를 얻었을 때의 처리
function show_location(event) {
    // 위도
    var latitude = event.coords.latitude;
    if (latitude == null) latitude = "-";

    document.querySelector('#latitude').textContent = latitude;
    // 경도
    var longitude = event.coords.longitude;
    if (longitude == null) longitude = "-";

    document.querySelector('#longitude').textContent = longitude;

    // 위도와 경도의 정밀도
    var accuracy = event.coords.accuracy;
    if (accuracy == null) accuracy = "-";

    document.querySelector('#accuracy').textContent = accuracy;
    // GPS고도
    var altitude = event.coords.altitude;
    if (altitude == null) altitude = "-";

    document.querySelector('#altitude').textContent = altitude;
    // GPS고도의 정밀도
    var altitudeAccuracy = event.coords.altitudeAccuracy;
    if (altitudeAccuracy == null) altitudeAccuracy = "-";

    document.querySelector('#altitudeAccuracy').textContent = altitudeAccuracy;
    // 이동 방향
    var heading = event.coords.heading;
    if (heading == null) heading = "-";

    document.querySelector('#heading').textContent = heading;
    // 이동 속도
    var speed = event.coords.speed;
    if (speed == null) speed = "-";

    document.querySelector('#speed').textContent = speed;
    // 시간
    var date = event.timestamp;

    if (typeof(date) == "number") {
        date = new Date(date);
    }
    document.querySelector('#timestamp').textContent = date.toString();
}
```

Geolocation API

- 위치 정보를 얻지 못했을 때의 처리
  - 위치 정보를 얻지 못했을 때는 getCurrentPosition() 함수의 두 번째 매개변수에 지정된 함수를 실행한다.
  - 오류 내용은 이벤트 객체에서 얻을 수 있으며 다음과 같은 속성이 있다.

오류 내용을 얻기 위한 속성	설명
event.code	현재 오류를 나타내는 코드 번호를 반환한다. 각 번호의 의미는 다음과 같다. 1: 위치 정보 수집 허가를 얻지 못한 경우(PERMISSION_DENIED) 2: 위치 정보를 얻지 못한 경우(POSITION_UNAVALABLE) 3: 타임 아웃이 발생한 경우(TIMEOUT)
event.message	오류 내용을 텍스트로 반환한다.
event.PERMISSION_DENIED	항상 1을 반환
event.POSITION_UNAVALABLE	항상 2를 반환
event.TIMEOUT	항상 3을 반환

Geolocation API

- 위치 정보를 얻지 못했을 때의 처리 예제

```
document.addEventListener("DOMContentLoaded", function() {
    // 현재 위치 정보 얻기
    window.navigator.geolocation.getCurrentPosition(
        show_location, // 위치 정보를 얻었을 때의 호출되는 콜백 함수
        show_error // 위치 정보를 얻지 못했을 때 호출되는 콜백 함수
    );
}, false);
```

```
// 위치정보를 얻지 못했을 때의 처리
function show_error(event) {
    alert(event.message + "(" + event.code + ")");
}
```

Geolocation API

- 옵션 매개 변수
  - getCurrentPosition() 함수의 세 번째 매개변수에 위치 정보에 관한 변수를 정의할 수 있다.

위치 정보에 대한 매개변수를 정의하기 위한 속성	설명
PositionOptions.enableHighAccuracy	true를 지정하면 브라우저에 정확한 위치 정보를 요청한다. 이 매개변수가 지정되지 않으면 false가 적용된다.
PositionOptions.timeout	위치 정보를 얻을 때까지 기다릴 시간을 밀리 초 단위로 지정한다. 만약 음수가 지정되면 0이 지정된 것으로 간주한다.
PositionOptions.maximumAge	캐시된 위치 정보의 유효 시간(밀리 초)을 지정한다. 지정하지 않으면 0으로 간주한다.

```
document.addEventListener("DOMContentLoaded", function() {
    // 옵션 인자 값을 지정
    var position_options = {
        enableHighAccuracy: true, // 정확한 위치 요구
        timeout: 60000, // 최대 대기 시간(밀리초)
        //      timeout: 0,           // 최대 대기 시간(밀리초)
        maximumAge: 0 // 캐시 유효기간(밀리초)
    };
    // 현재 위치정보 얻기
    window.navigator.geolocation.getCurrentPosition(
        show_location, // 위치 정보를 얻었을 때 실행되는 콜백 함수
        show_error, // 위치 정보를 얻지 못했을 때 실행 되는 콜백 함수
        position_options // 옵션 인자 값
    );
}, false);
```

연속된 위치정보 얻기

- 연속해서 위치정보 얻기
  - 기존의 getCurrentPosition() 함수는 위치 정보를 한번만 요청한다.
  - 반면 이런 단점을 보완하기 위해 watchPosition() 이라는 함수가 존재하며 비동기적으로 위치 정보를 얻기 위한 처리가 반복되고 항상 위치정보를 감시하는 상태를 만들 수 있다.
  - 이 함수 역시 매개변수를 최대 세 개 가질 수 있으며 위치정보를 얻었을 때 실행되는 첫번째 매개변수(successCallback), 위치정보를 얻지 못했을 때 실행되는 함수 객체(errorCallback). 위치 정보에 대한 각종 매개 변수를 포함하는 매개변수(options)로 구성된다.
  - watchPosition()이 갖고 있는 매개변수는 getCurrentPosition() 이 가지고 있는 매개변수와 정확하게 매치 된다.
  - watchPosition() 함수는 clearWatch() 함수로 종료될 때까지 위치 정보를 계속 감시한다.

연속된 위치정보 얻기

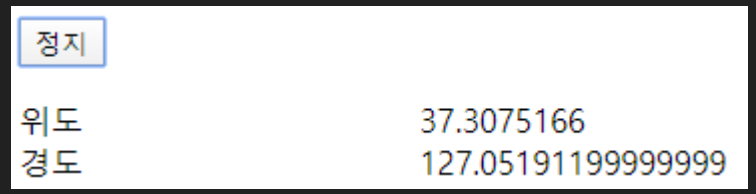
- 연속해서 위치정보 얻기 예제 - 1

```
document.addEventListener("DOMContentLoaded", function() {
    // 옵션 인자 값을 지정
    var position_options = {
        enableHighAccuracy: true, // 정확한 위치 요구
        timeout: 60000, // 최대 대기 시간(밀리초)
        maximumAge: 0 // 캐시 유효기간(밀리초)
    };
    // 현재 위치 얻기
    window.navigator.geolocation.watchPosition(monitor, null, position_options);
}, false);
```

연속된 위치정보 얻기

- 연속해서 위치정보 얻기 예제 - 2

```
// 실시간 감시
function monitor(event) {
    // 위도
    var latitude = event.coords.latitude;
    document.querySelector('#latitude').textContent = latitude;
    // 경도
    var longitude = event.coords.longitude;
    document.querySelector('#longitude').textContent = longitude;
}
```




```
<p><button type="button">시작 </button></p>
<dl>
    <dt>위도 </dt>
    <dd id="latitude">-</dd>
    <dt>경도 </dt>
    <dd id="longitude">-</dd>
</dl>
<script type="text/javascript">
document.addEventListener("DOMContentLoaded", function() {
    // 감시 식별 ID
    var watch_id = 0;
    // 버튼에 click 이벤트 리스너 지정
    var button = document.querySelector('button');
    button.addEventListener("click", function() {
        if (watch_id > 0) {
            // 실시간 감시를 정지
            window.navigator.geolocation.clearWatch(watch_id);
            // 감시 식별 ID에 0을 지정
            watch_id = 0;
            // 버튼 표기를 변경
            button.textContent = "시작 ";
        } else {
            // 실시간 감시 시작
            watch_id = window.navigator.geolocation.watchPosition(monitor);
            // 버튼 표기를 변경
            button.textContent = "정지 ";
        }
    }, false);
}, false);
```

연속된 위치정보 얻기

- 구글 맵 플랫폼 연동