



CHAPTER 13.

HTML5 위치정보와 드래그앤 드롭





SVG

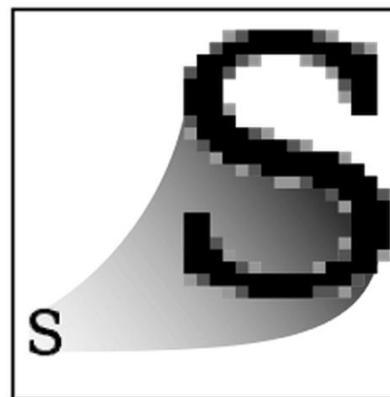
- **SVG(Scalable Vector Graphics)**는 XML-기반의 벡터 이미지 포맷
- 웹에서 벡터-기반의 그래픽을 정의하는데 사용
- 1999년부터 W3C에 의하여 표준





SVG의 장점

- SVG 그래픽은 확대되거나 크기가 변경되어도 품질이 손상되지 않는다.
- SVG 파일에서 모든 요소와 속성은 애니메이션이 가능하다.
- SVG 이미지는 어떤 텍스트 에디터로도 생성하고 편집할 수 있다.



Raster
.jpeg .gif .png



Vector
.svg



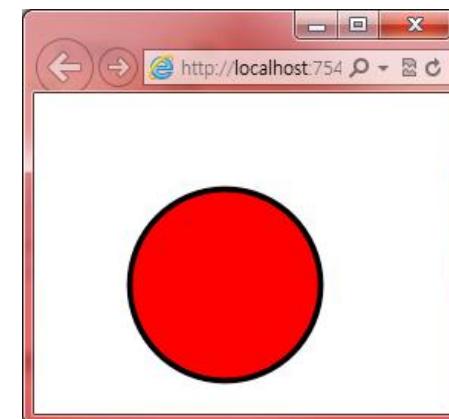
SVG

- **SVG Shapes**
 - SVG has some predefined shape elements that can be used by developers:
 - Rectangle <rect>
 - Circle <circle>
 - Ellipse <ellipse>
 - Line <line>
 - Polyline <polyline>
 - Polygon <polygon>
 - Path <path>
- **SVG Reference**
 - http://www.w3schools.com/svg/svg_reference.asp



원 예제

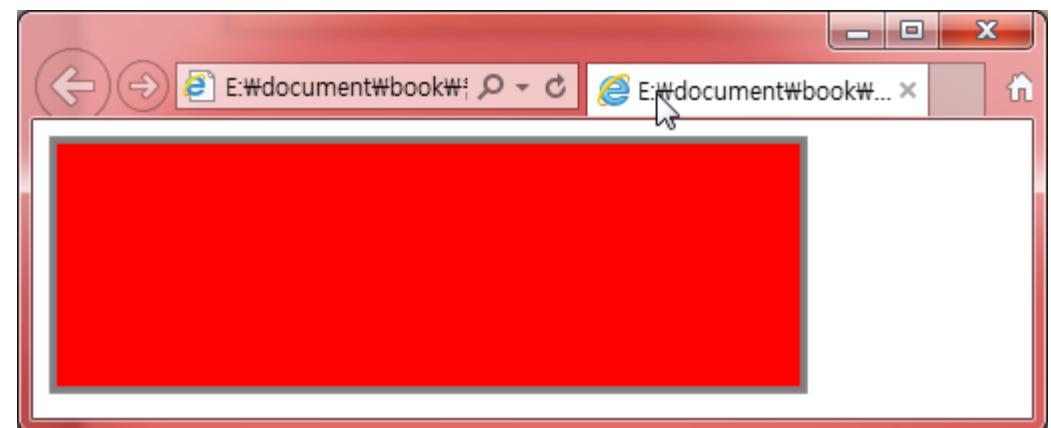
```
<!DOCTYPE html>
<html>
<body>
  <svg xmlns="http://www.w3.org/2000/svg" version="1.1">
    <circle cx="100" cy="100" r="50" stroke="black"
           stroke-width="3" fill="red" />
  </svg>
</body>
</html>
```





사각형 예제

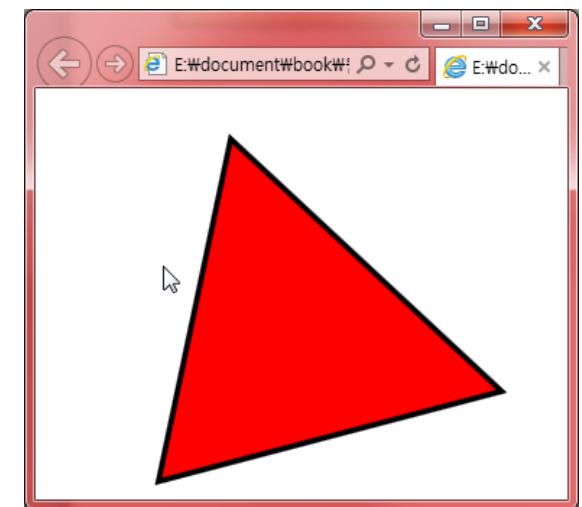
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect width="300" height="100"
    style="fill: rgb(255,0, 0); stroke-width: 3; stroke: rgb(128, 128, 128)" />
</svg>
```





다각형 예제

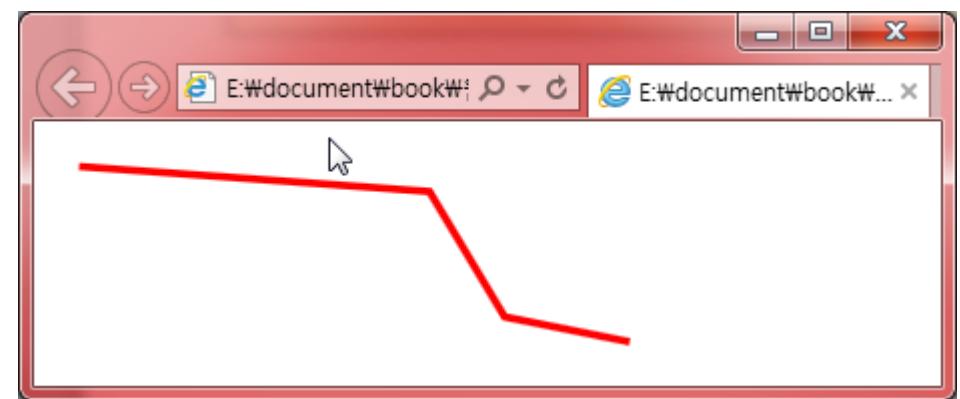
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
    <polygon points="100,20 250,160 60,210"
        style="fill: red; stroke: black; stroke-width: 3" />
</svg>
```





폴리라인 예제

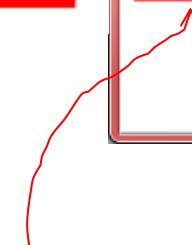
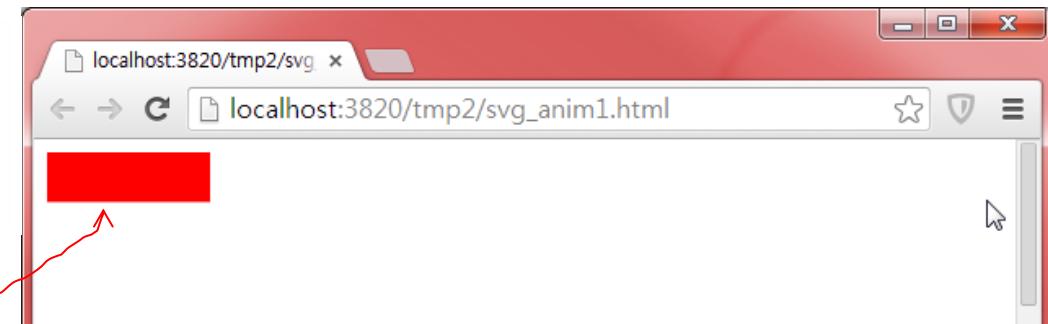
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <polyline points="10,10 150,20 180,70 230,80"
    style="fill: none; stroke: red; stroke-width: 3" />
</svg>
```





애니메이션 #1

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect width="100" height="100" fill="red">
    <animate attributeName="height" from="0" to="100" dur="10s" />
  </rect>
</svg>
```

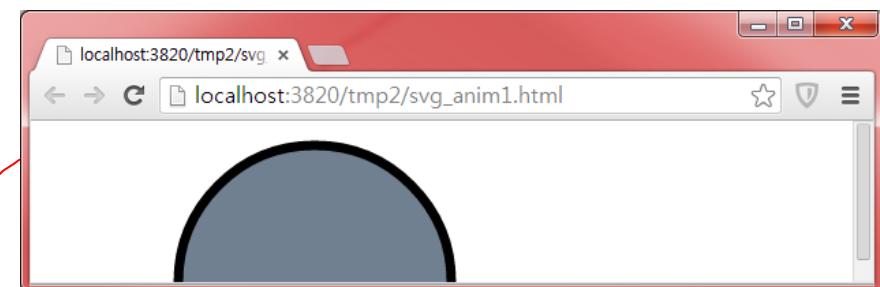
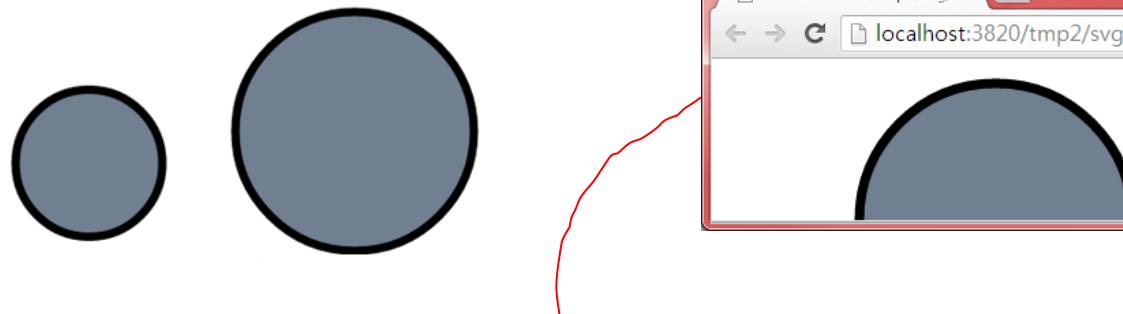


구글 크롬을 사용한다!



애니메이션 #2

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle r="100" cx="200" cy="110" fill="slategrey" stroke="#000" stroke-width="7">
    <animate attributeName="r" from="0" to="100" dur="3s" />
    <animate attributeName="cx" from="100" to="200" dur="3s" />
  </circle>
</svg>
```



구글 크롬을 사용한다!

chap13\svg_anim1.html

chap13\svg_anim1_mod.html

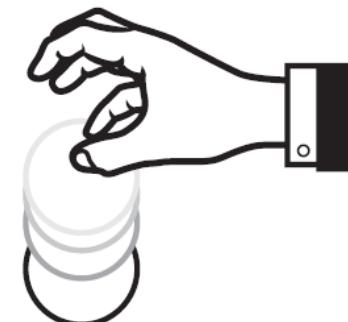


드래그와 드롭

- **드래그(drag)**와 **드롭(drop)** - 윈도우에서 아주 많이 사용하는 사용자 인터페이스 중의 하나
- 객체를 마우스로 끌어서 다른 애플리케이션에 놓는 것



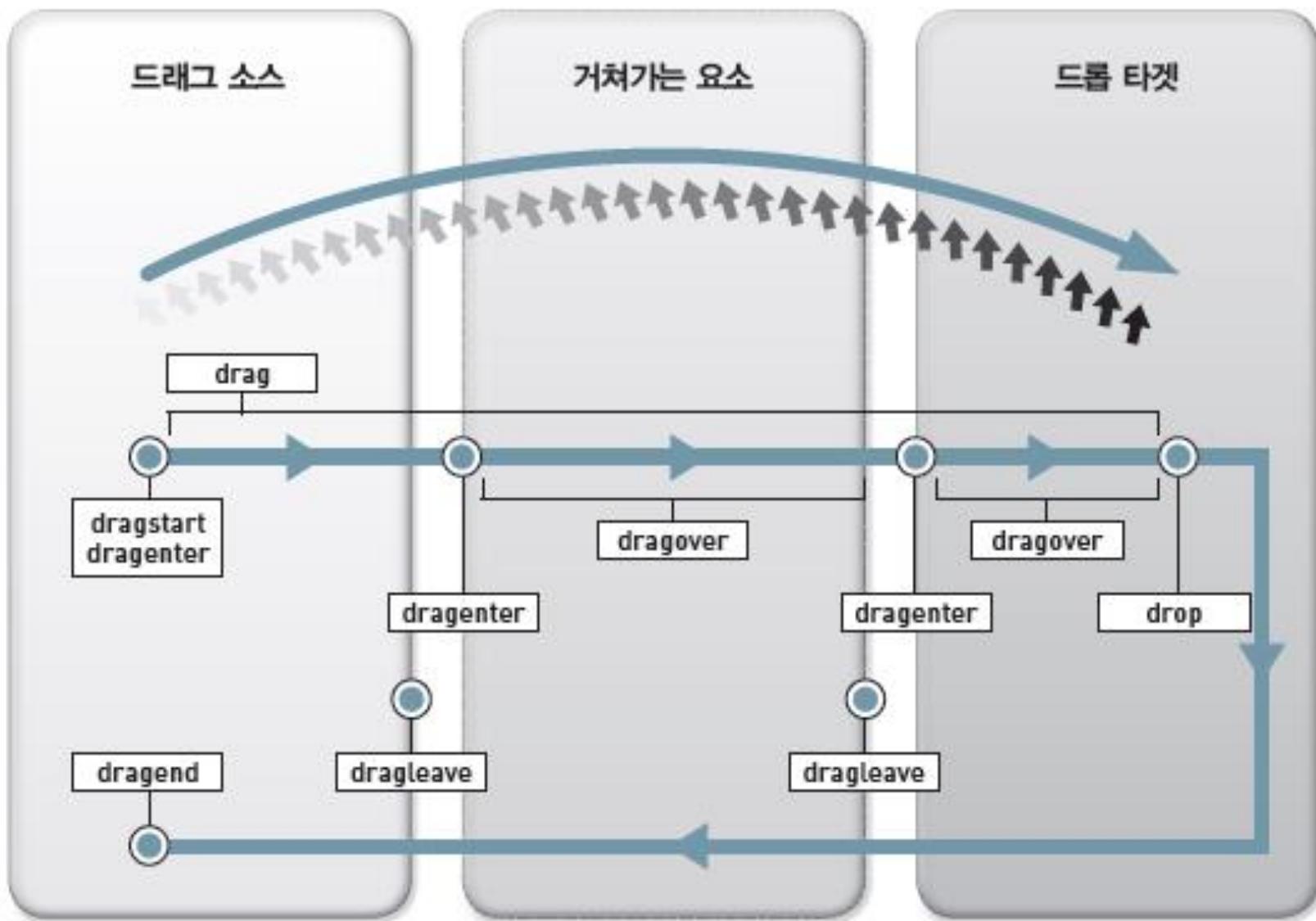
드래그



드롭



발생하는 이벤트





- dragstart 이벤트
 - 사용자가 드래그를 시작할 때 발생
 - dataTransfer 객체의 setData() 호출

```
function handleDragStart(e) {  
    e.dataTransfer.setData("Text", e.target.id);  
}
```
- drag 이벤트
 - 드래그하는 도중에 계속해서 발생하는 이벤트
- dragenter 이벤트
 - 새로운 요소 안으로 들어가면 발생하는 이벤트
 - 새로운 요소가 타겟 요소이면 drop 이벤트를 처리하는 콜백 함수 설정 가능
- dragleave 이벤트
 - 요소를 빠져 나가면 발생하는 이벤트



- dragover 이벤트
 - 드래그 동작 도중 마우스가 다른 요소 위에 있다는 것을 의미
 - default 값은 요소 위에 다른 요소가 드롭될 수 없으므로 드롭을 허용하려면 event.preventDefault() 메서드 호출 필요
- function allowDrop(e) {
 e.preventDefault();
}
- drop 이벤트
 - 마우스 버튼을 놓았을 때 발생하는 이벤트로 반드시 처리해야 함
 - dataTransfer 객체에서 getData() 메서드를 이용해 필요한 데이터를 꺼내면 됨



예제

```
<!DOCTYPE HTML>
<html>
<head>
<style>
#shopping_cart {
    width: 450px;
    height: 100px;
    padding: 10px;
    border: 1px dotted red;
}
</style>
<script>
function allowDrop(e) {
    e.preventDefault();
}

function handleDragStart(e) {
    e.dataTransfer.effectAllowed = 'move';
    e.dataTransfer.setData("Text", e.target.id);
}
```



예제

```
function handleDrop(e) {
    e.preventDefault();
    var src = e.dataTransfer.getData("Text");
    e.target.appendChild(document.getElementById(src));
}
</script>
</head>
<body>
    <p>원하는 물건을 끌어서 옮기세요.</p>
    <div id="shopping_cart"
        ondrop="handleDrop(event)" ondragover="allowDrop(event)"></div>
        <br>
        
            
                
</body>
</html>
```



실행 결과



웹브라우저로 실행하기



HTML5 위치정보

- 위치정보(Geolocation)은 자신의 위치를 웹 사이트와 공유
- 현재 지역의 날씨, 유명한 맛집 등의 정보를 제공받을 수 있다.





geolocation 객체

- var geolocation = navigator.geolocation;

메소드	설명
<u>getCurrentPosition()</u>	사용자의 현재 위치 정보를 반환한다.
<u>watchPosition()</u>	장치의 현재 위치에 대한 정보를 주기적으로 반환한다.
<u>clearWatch()</u>	현재 진행 중인 <u>watchPosition()</u> 실행을 중지한다.



Geolocation Example

- ```
<script>
var x = document.getElementById("demo");
function getLocation() {
 if (navigator.geolocation) {
 navigator.geolocation.getCurrentPosition(showPosition);
 } else {
 x.innerHTML = "Geolocation is not supported by this browser.";
 }
}
function showPosition(position) {
 x.innerHTML = "Latitude: " + position.coords.latitude +
 "
Longitude: " + position.coords.longitude;
}
</script>
```



- Example explained:
  - 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 gets the displays the Latitude and Longitude
  - The example above is a very basic Geolocation script, with no error handling.



# Handling Errors and Rejections

- 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: geolocation.getCurrentPosition(showLocation, **showError**)
- ```
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;
    }
}
```



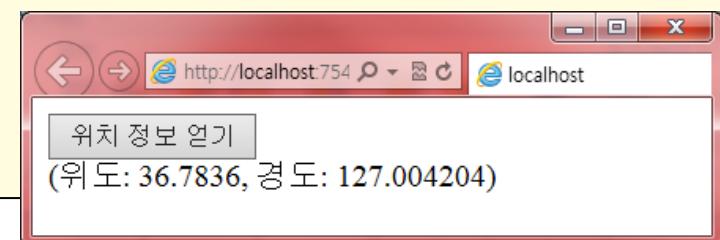
Position 객체의 중요 속성

속성	타입	설명
coords	objects	장치의 위치 정보를 갖고 있는 객체
coords.latitude	Number	위도 정보 (단위는 도(degree), -90~+90)
coords.longitude	Number	경도 정보 (단위는 도(degree), -180~+180)
coords.altitude	Number	고도 정보 (단위는 미터)
coords.heading	Number	북쪽으로부터 시계 방향으로 방위각
coords.speed	Number	장치의 현재 속도
timestamp	date	위치 정보가 반환된 시각



예제

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="getGeolocation()">위치 정보 얻기</button>
    <div id="target"></div>
    <script>
        var myDiv = document.getElementById("target");
        function getGeolocation() {
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(showLocation);
            }
        }
        function showLocation(location) {
            myDiv.innerHTML = "(위도: " + location.coords.latitude +
                ", 경도: " + location.coords.longitude + ")"
        }
    </script>
</body>
</html>
```

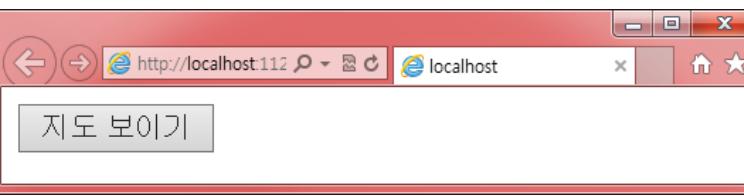




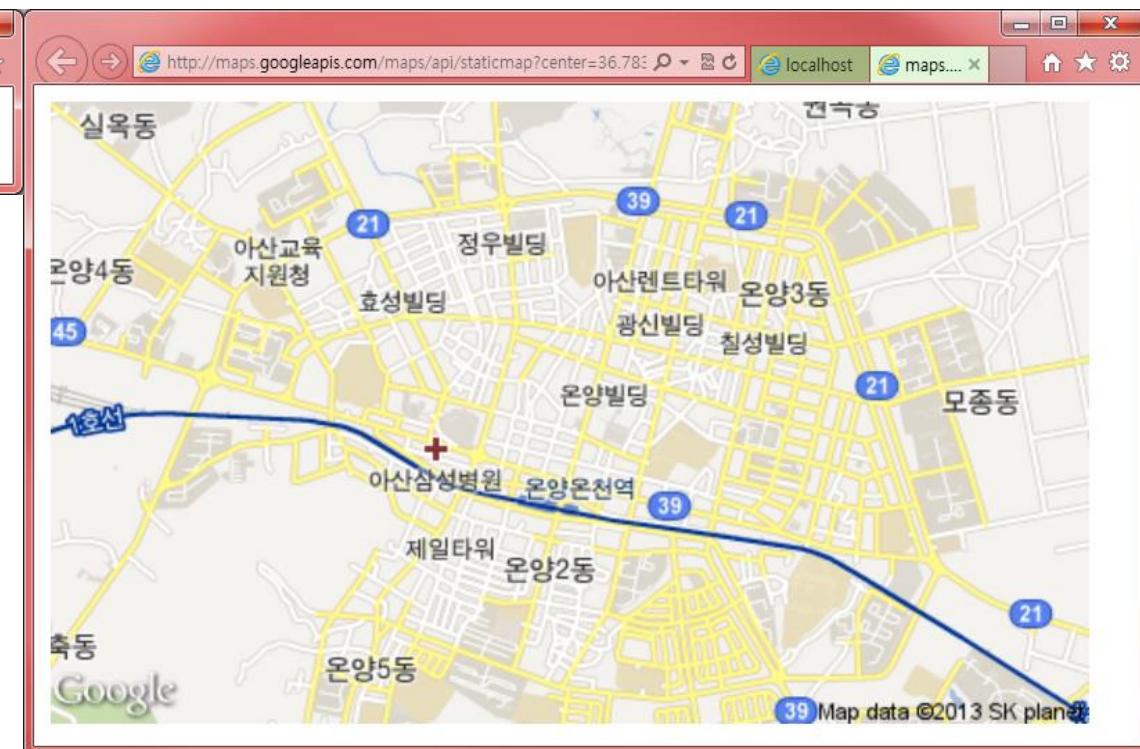
지도에 위치 표시하기

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="getGeolocation()">지도 보이기</button>
    <script>
        var myDiv = document.getElementById("target");
        function getGeolocation() {
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(showGeolocation);
            }
        }
        function showGeolocation(position) {
            var pos = position.coords.latitude + "," + position.coords.longitude;
            var url = "http://maps.googleapis.com/maps/api/staticmap?center=" +
                      pos + "&zoom=14&size=500x300&sensor=false";
            window.open(url);
        }
    </script>
</body>
</html>
```

실행결과



웹브라우저로 실행하기





이동하면서 위치 정보를 얻기

- geolocation 객체의 watchPosition()을 호출
 - watchPosition() - 사용자의 현재 위치를 연속하여 출력한다.
 - clearWatch() - watchPosition() 메서드를 중지한다.

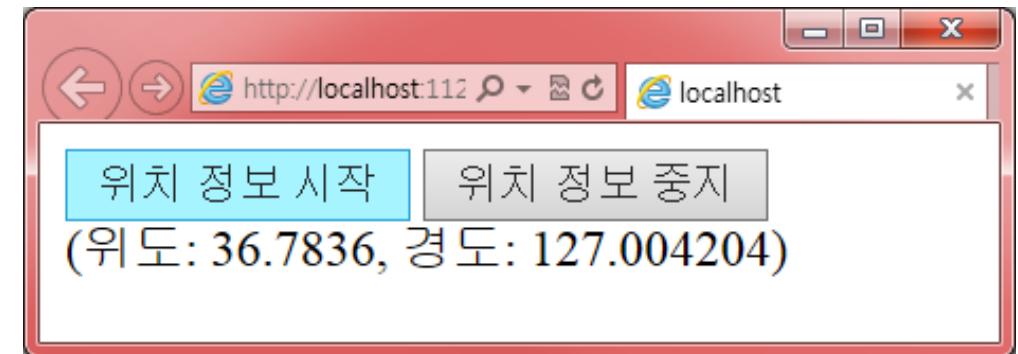


이동하면서 위치 정보를 얻기

```
<!DOCTYPE html>
<html>
<body>
    <button onclick="startGeolocation()">위치 정보 시작</button>
    <button onclick="stopGeolocation()">위치 정보 중지</button>
    <div id="target"></div>
    <script>
        var id;
        var myDiv = document.getElementById("target");
        function startGeolocation() {
            if (navigator.geolocation) {
                id = navigator.geolocation.watchPosition(showGeolocation);
            }
        }
        function showGeolocation(location) {
            myDiv.innerHTML = "(위도: " + location.coords.latitude +
                ", 경도: " + location.coords.longitude + ")";
        }
        function stopGeolocation() {
            if (navigator.geolocation) {
                navigator.geolocation.clearWatch(id);
            }
        }
    </script>
</body>
</html>
```



실행결과





HTML5 웹 워커

- 웹 워커(**web worker**): 자바스크립트에 백그라운드에서 실행되는 스레드(thread)를 도입한 것





소수 구하기

worker.js

```
// 소수를 찾는 자바스크립트 소스
var n = 1;
search: while (true) {
    n += 1;
    for (var i = 2; i <= Math.sqrt(n) ; i += 1)
        if (n % i == 0)
            continue search;
    // 소수를 발견할 때마다 바로 웹페이지로 전달한다.
    postMessage(n);
}
```



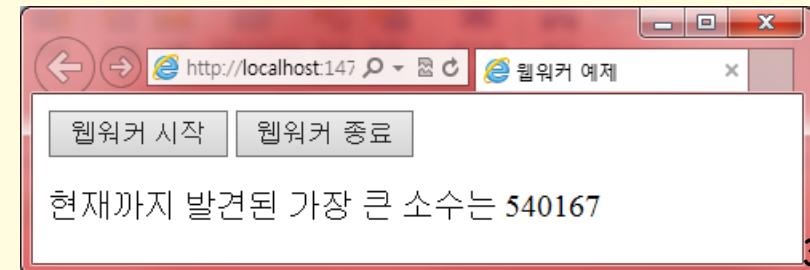
소수 구하기

```
<!DOCTYPE HTML>
<html>
<head>
    <title>웹 워커 예제</title>
</head>
<body>
    <button onclick="startWorker()">웹 워커 시작</button>
    <button onclick="stopWorker()">웹 워커 종료</button>
    <p>현재까지 발견된 가장 큰 소수는
        <output id="result"></output>
    </p>
    <script>
        var w;
```



소수 구하기

```
function startWorker() {
    if (typeof (Worker) !== "undefined") {
        if (typeof (w) == "undefined") {
            w = new Worker("worker.js");
        }
        w.onmessage = function (event) {
            document.getElementById("result").innerHTML = event.data;
        };
    } else {
        document.getElementById("result").innerHTML = "웹브라우저가
웹워커를 지원하지 않음";
    }
}
function stopWorker() {
    w.terminate();
}
</script>
</body>
</html>
```





웹 워커 용도

- 영상 처리
- 대용량 데이터 처리
- 텍스트 분석
- 데이터베이스 요청 동시 수행



Q & A

