

# Kakao Map

# 지도

## 1. 지도 생성하기

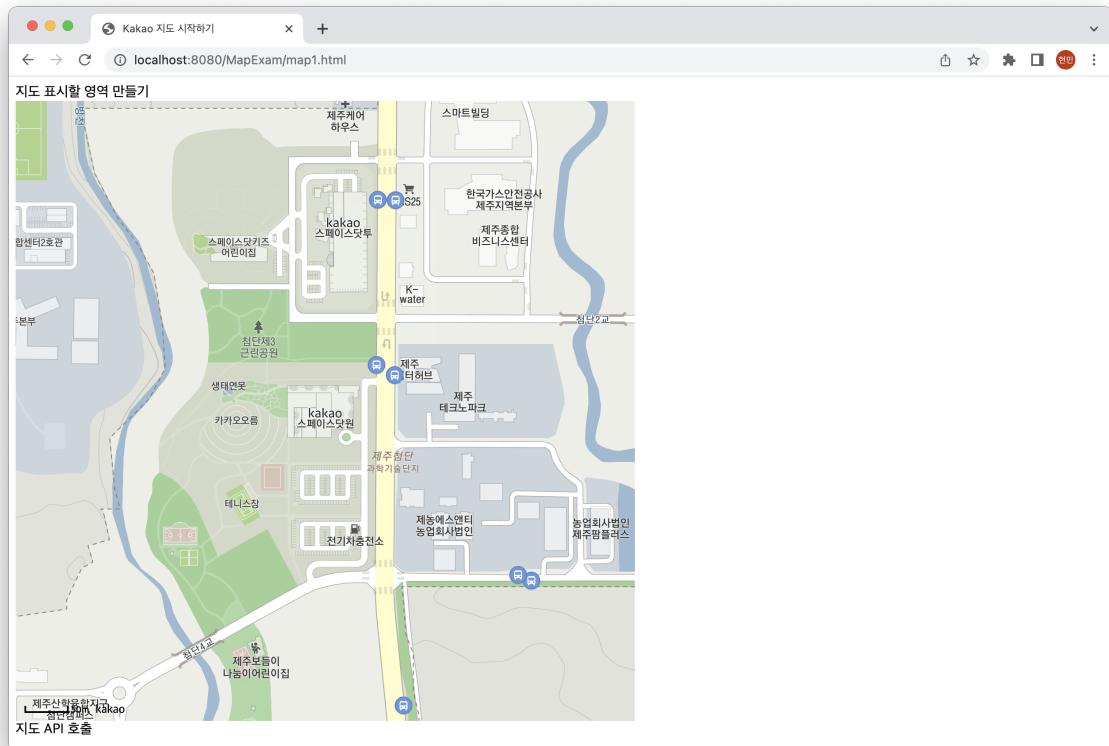
```
지도 표시할 영역 만들기
<div id="map" style="width:700px;height:700px;"></div>

지도 API 호출
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=fc7abe9b9479ffa297474094a4964dcf"></script>

<script>
  var container = document.getElementById('map');

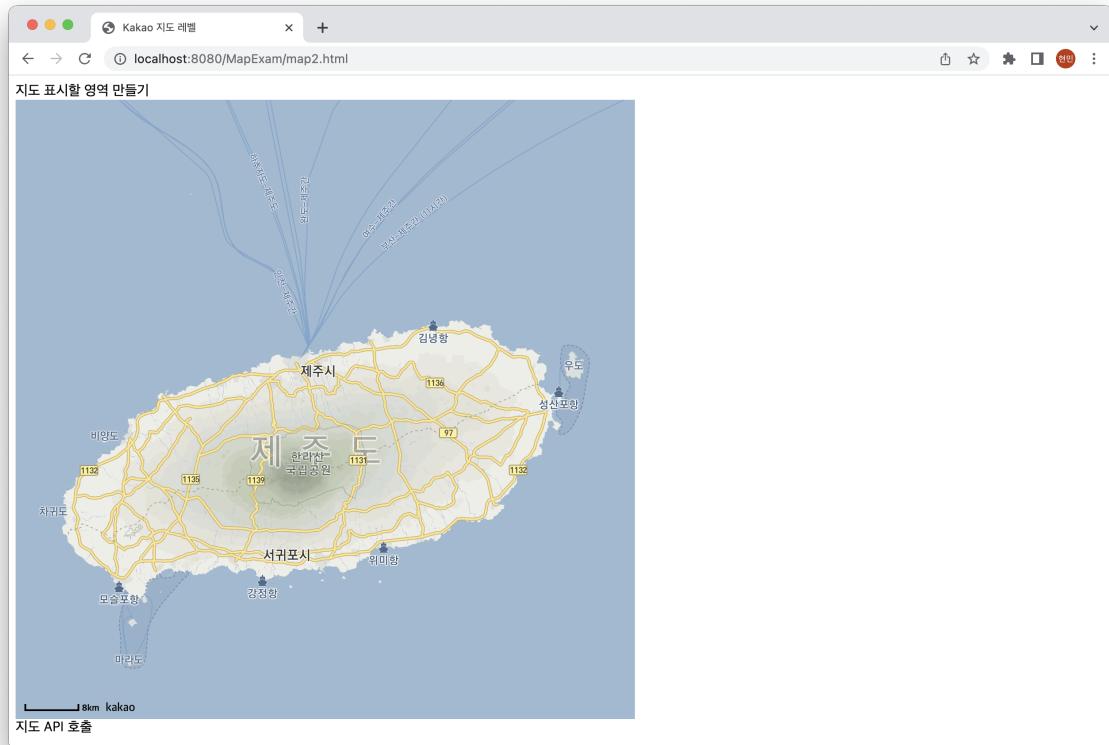
  var options = {
    center: new kakao.maps.LatLng(33.450701, 126.570667),
    level: 3
  };

  var map = new kakao.maps.Map(container, options);
</script>
```



## 2. 지도 레벨 바꾸기

`level: 10 //지도 크기 ex) 10: 서울 전체`



### 3. 지도 정보 얻어오기

```

function getInfo(){
    //지도에서 현재 중심 좌표 값 얻기
    var center = map.getCenter();

    //지도에서 현재 레벨(축적) 얻기
    var level = map.getLevel();

    //지도에서 현재 영역 얻기
    var bounds = map.getBounds();

    //지도의 영역 정보 얻기
    var boundsStr = bounds.toString();

    var message = '지도 중심 좌표는 위도 ' + center.getLatitude() + ',<br>';
    message += '경도 ' + center.getLongitude() + '이고<br>';
    message += '지도의 레벨은 ' + level + '입니다.<br><br>';
    message += '지도의 영역 정보는 ' + boundsStr + '입니다.';
}

```

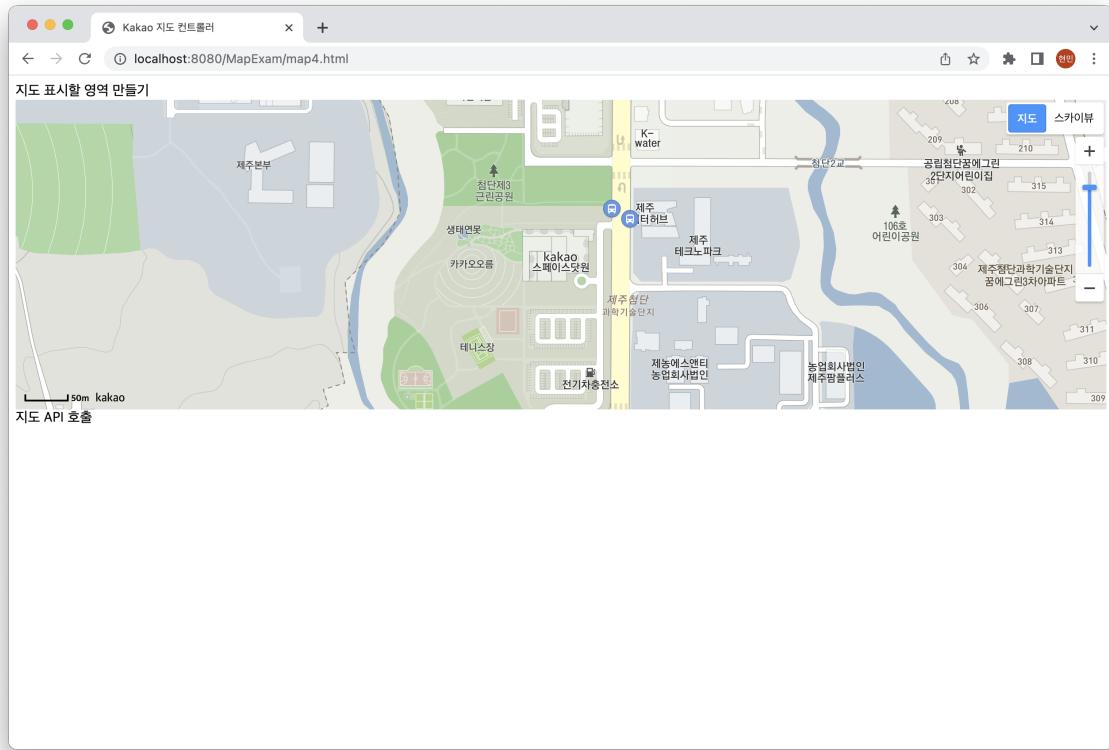
### 4. 지도에 컨트롤러 올리기

```

//지도의 형태를 바꾸는 컨트롤러 올리기
var mapTypeControl = new kakao.maps.MapTypeControl();
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);

//지도의 크기를 확대 / 축소하는 컨트롤러 올리기
var zoomControl = new kakao.maps.ZoomControl();
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);

```



## 5. 지도 이동 막기

```
//지도 이동 막기
map.setDraggable(false);
```

## 6. 지도 확대 축소 막기

```
//지도 확대 축소 막기
map.setZoomable(false);
```

## 7. 지도 범위 재설정 하기

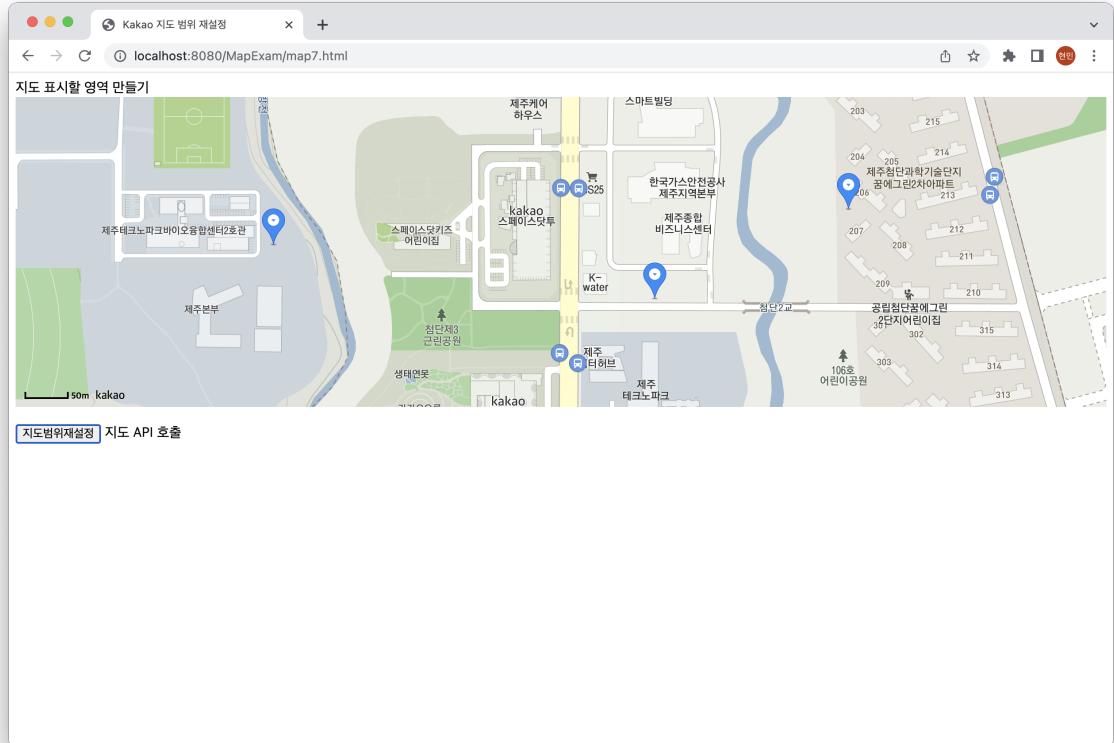
```
//표시할 좌표 값을 변수 설정
var points = [
  new kakao.maps.LatLng(33.452278, 126.567803),
  new kakao.maps.LatLng(33.452671, 126.574792),
  new kakao.maps.LatLng(33.451744, 126.572441)
];

//지도 재설정 범위 정보 객체 생성
var bounds = new kakao.maps.LatLngBounds();

//세개의 좌표 값을 지도에 올리기
var i, marker;
for(i=0; i<points.length; i++){
  //하나씩 지도에 추가
  marker = new kakao.maps.Marker({position:points[i]});
  marker.setMap(map);

  //bound 객체에 좌표 추가
  bounds.extend(points[i]);
}
```

```
//지도의 중심좌표와 레벨이 변경될 수 있다
function setBounds(){
    map.setBounds(bounds);
}
```



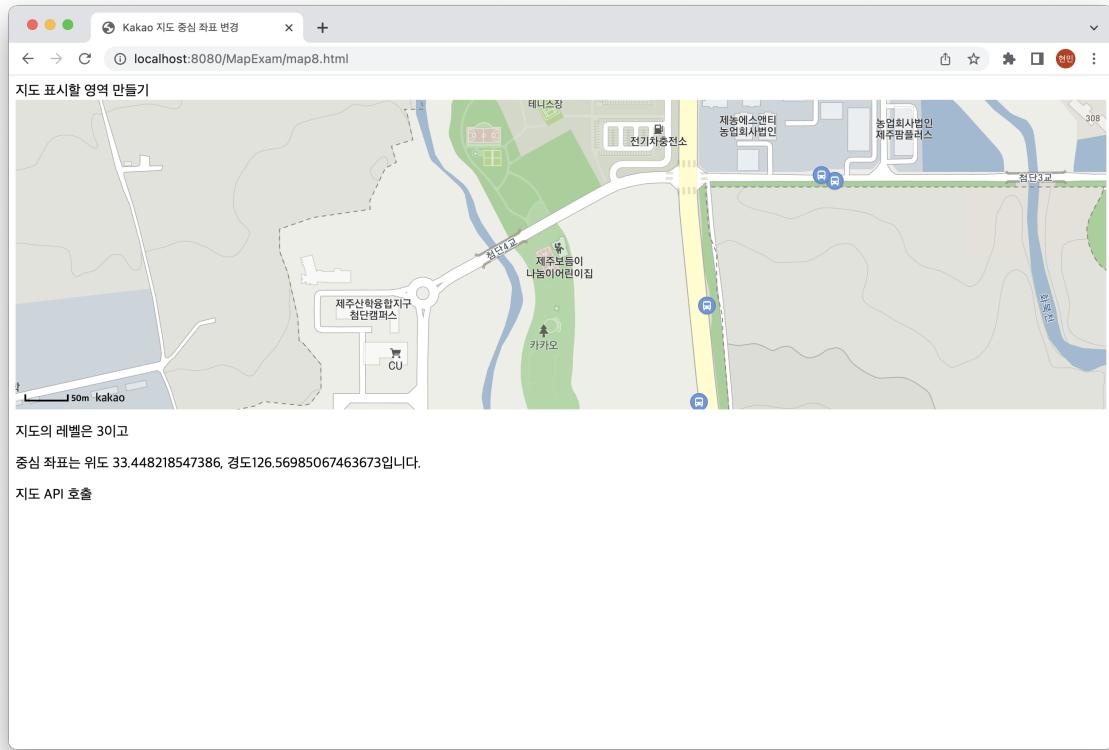
## 8. 중심좌표 변경 이벤트 등록하기

```
//지도 변경 시 중심 좌표
kakao.maps.event.addListener(map, 'center_changed', function(){
    //지도의 레벨 얻기
    var level = map.getLevel();

    //지도의 중심 좌표 얻기
    var latlng = map.getCenter();

    var message = '<p>지도의 레벨은 ' + level + '이고</p>';
    message += '<p>중심 좌표는 위도 ' + latlng.getLatitude() + ', 경도';
    message += latlng.getLongitude() + '입니다.</p>';

    var resultDiv = document.getElementById('result');
    resultDiv.innerHTML = message;
});
```

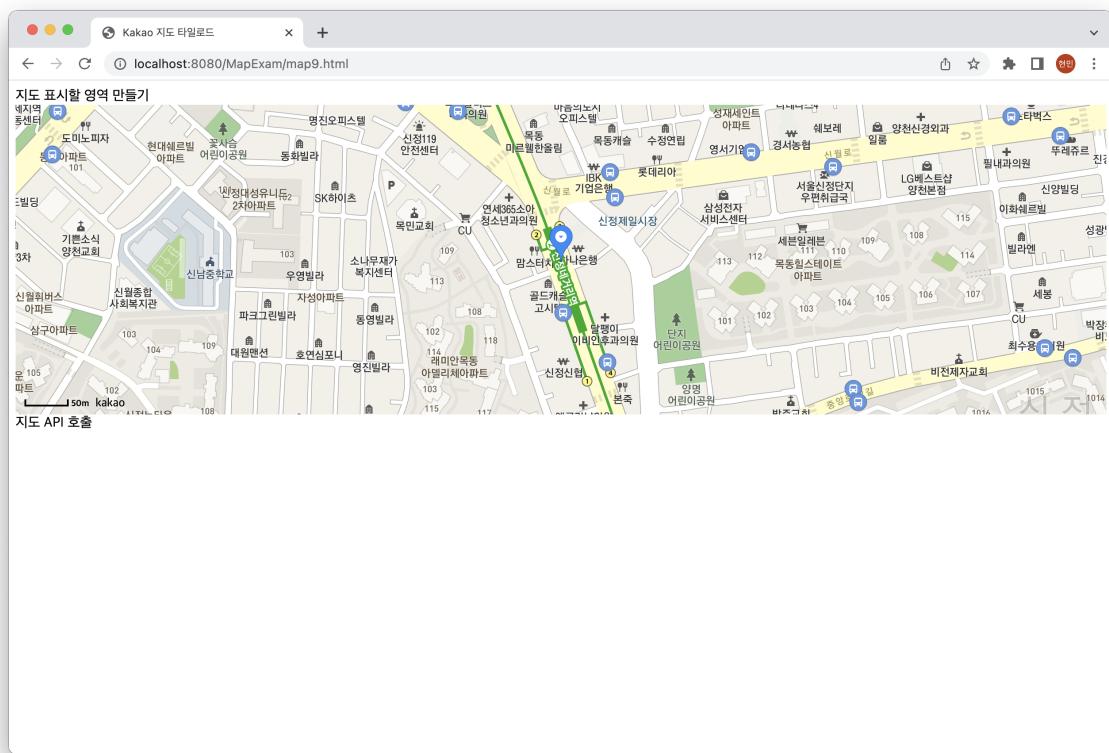


## 9. 타일로드 이벤트 등록하기

```
//마커 객체 생성
var marker = new kakao.maps.Marker();

//마커 지도에 올리기
kakao.maps.event.addListener(map, 'tilesloaded', displayMarker);

function displayMarker(){
    //마커를 지도의 중심으로 표시
    marker.setPosition(map.getCenter());
    marker.setMap(map);
}
```



## 오버레이

### 10. 마커 생성하기

```
//마커를 표시할 위치
var markerPosition = new kakao.maps.LatLng(37.522262, 126.850055);

//마커 생성
var marker = new kakao.maps.Marker({
    position:markerPosition
});

//마커 지도에 올리기
marker.setMap(map);
```

### 11. 드래그 가능한 마커 생성하기

```
//마커를 움직일 수 있도록 설정
marker.setDraggable(true);
```

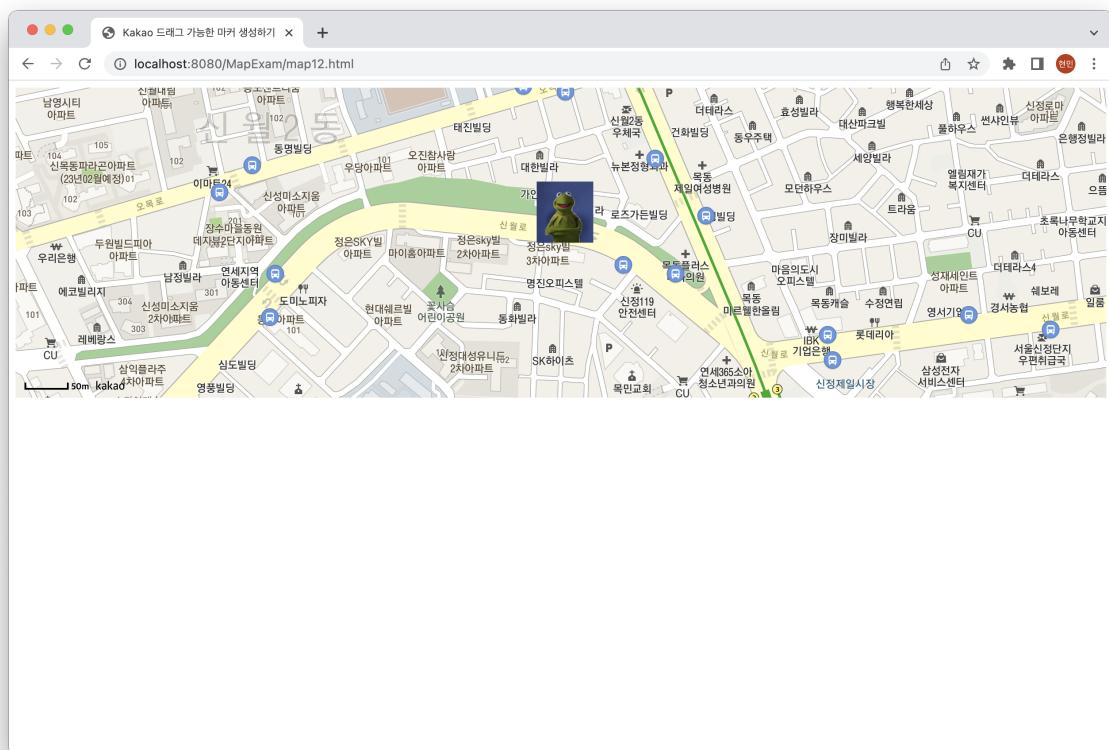
### 12. 다른 이미지로 마커 생성하기

```
//마커로 사용할 사진 설정
var imageSrc = 'kermit.jpeg',
    imageSize = new kakao.maps.Size(64,69),
    imageOption = {offset: new kakao.maps.Point(27,69)};

var markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize, imageOption),
    markerPosition = new kakao.maps.LatLng(37.522262, 126.850055);
```

```
//마커 생성
var marker = new kakao.maps.Marker({
    position: markerPosition,
    image: markerImage
});

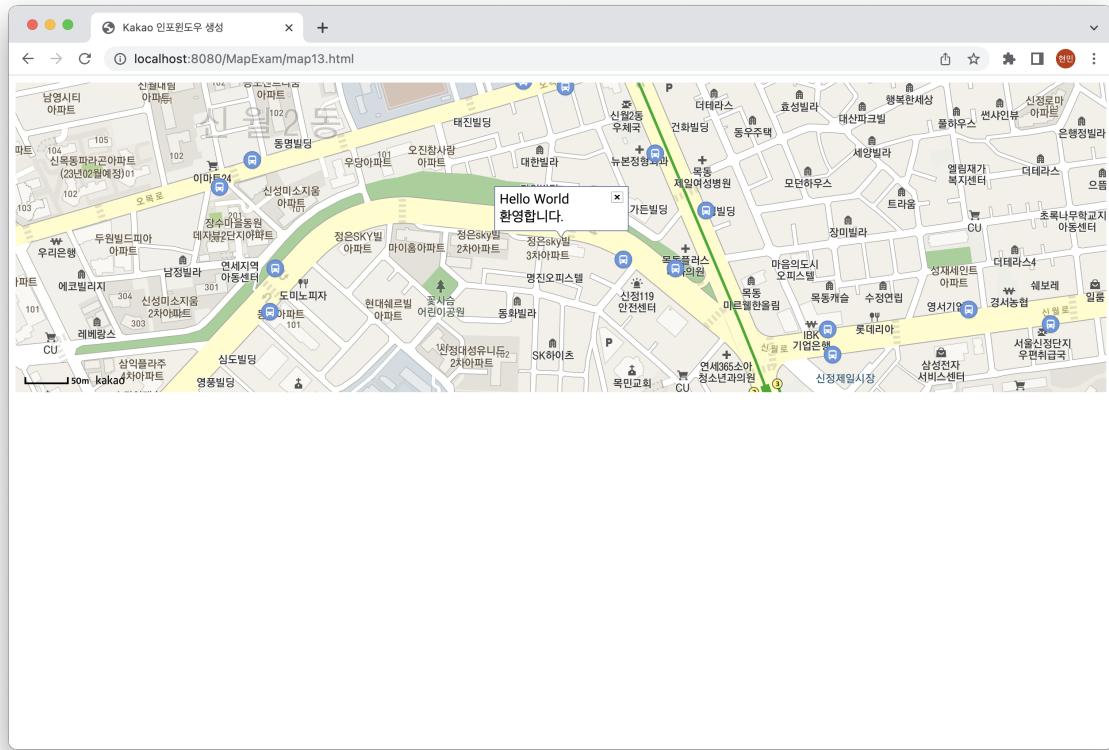
//마커 지도에 올리기
marker.setMap(map);
```



### 13. 인포윈도우 생성하기

```
//창에 나타낼 정보 작성
var iwContent = '<div style="padding:5px;">Hello World <br>환영합니다.</div>'
iwPosition = new kakao.maps.LatLng(37.522262, 126.850055),
iwRemovable = true;

//창 생성
var infowindow = new kakao.maps.InfoWindow({
    map: map,
    position: iwPosition,
    content: iwContent,
    removable: iwRemovable
});
```



#### 14. 마커에 인포윈도우 표시하기

```

//마커을리기
//마커위치설정
var markerPosition = new kakao.maps.LatLng(37.522262, 126.850055);

//마커 생성
var marker = new kakao.maps.Marker({
    position: markerPosition,
});

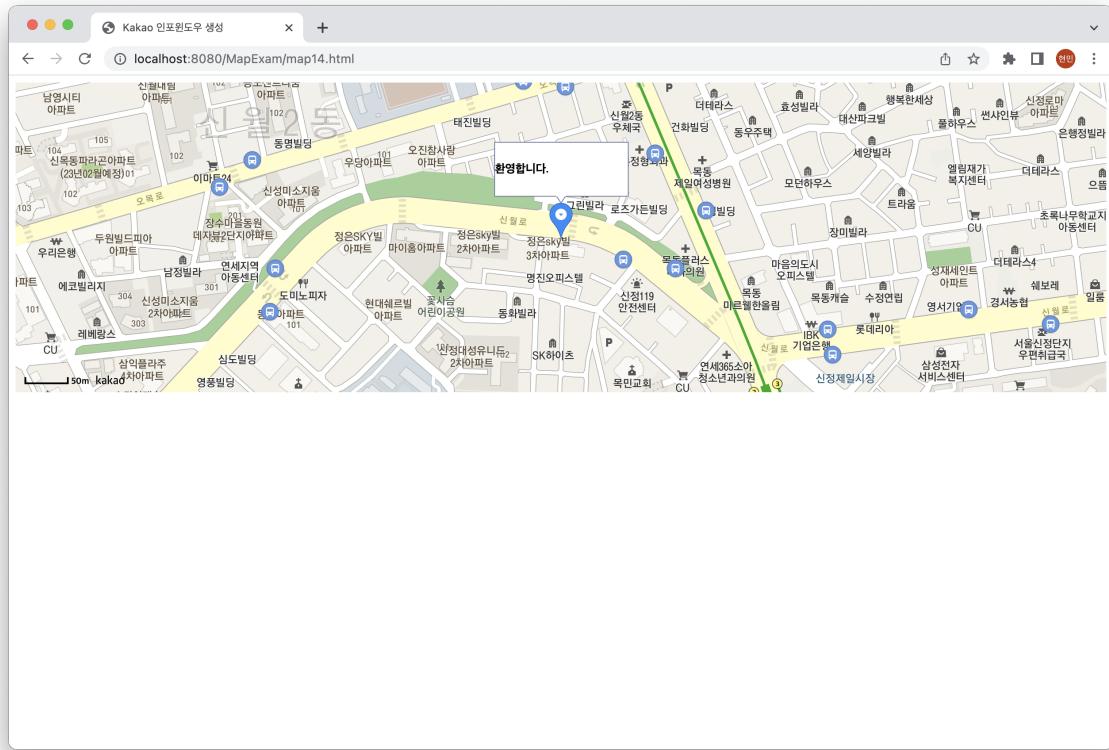
//마커 지도에 올리기
marker.setMap(map);

//정보창 올리기
//창에 나타낼 정보 작성
var iwContent = '<h5>환영합니다.</h5>';
iwPosition = new kakao.maps.LatLng(37.522262, 126.850055);

//창 생성
var infowindow = new kakao.maps.InfoWindow({
    position: iwPosition,
    content: iwContent
});

//정보창 지도에 올리기
infowindow.open(map, marker); -> marker 빼면 마커랑 정보창 겹침

```



## 15. 마커에 마우스 이벤트 등록하기

```
//마커
//마커 위치
var position = new kakao.maps.LatLng(37.522262, 126.850055);

//마커 생성
var marker = new kakao.maps.Marker({
    position: position,
});

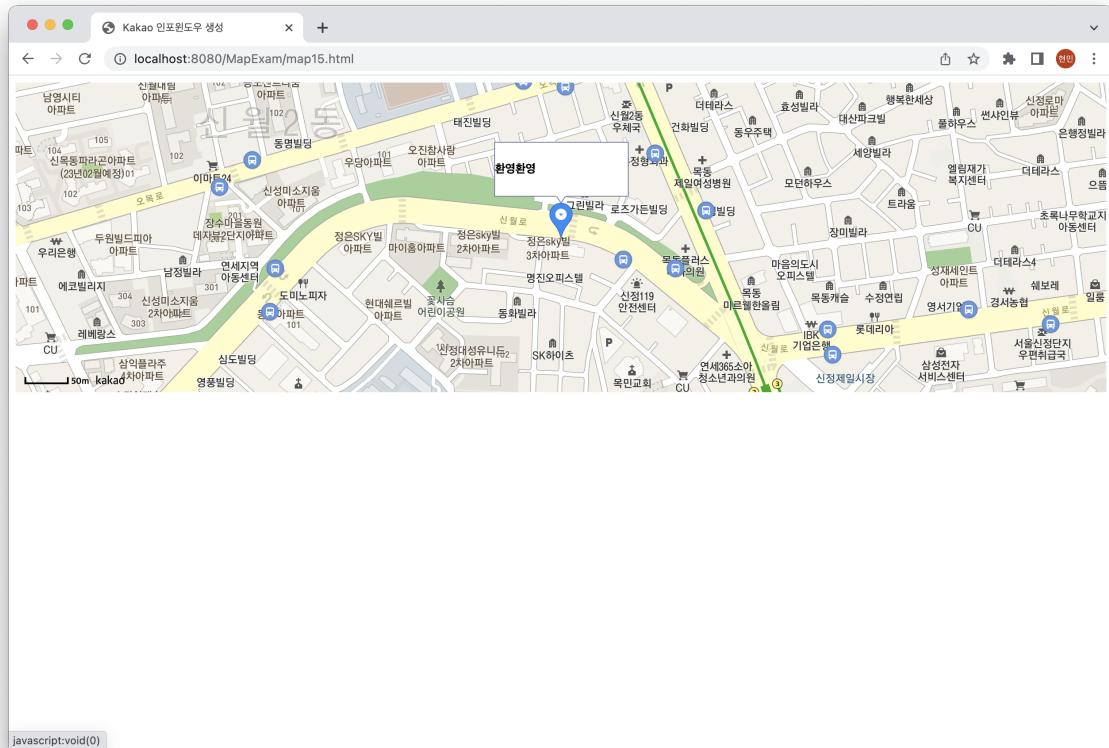
//마커 표시
marker.setMap(map);

//마커에 마우스를 올렸을때 인포 윈도우 표시
//인포윈도우 메세지
var iwContent ="<h5>환영환영</h5>";

//인포윈도우 객체 생성
var infowindow = new kakao.maps.InfoWindow({
    content: iwContent
});

//마커에 마우스를 올렸을 때 이벤트 발생
kakao.maps.event.addListener(marker, 'mouseover', function(){
    infowindow.open(map, marker);
});

//마커에서 마우스가 멀어졌을때 이벤트 생성
kakao.maps.event.addListener(marker, 'mouseout', function(){
    infowindow.close();
});
```



## 16. geolocation으로 마커 표시하기

```

if (navigator.geolocation) {

    // GeoLocation을 이용해서 접속 위치를 얻어옵니다
    navigator.geolocation.getCurrentPosition(function(position) {

        var lat = position.coords.latitude, // 위도
            lon = position.coords.longitude; // 경도

        var locPosition = new kakao.maps.LatLng(lat, lon), // 마커가 표시될 위치를 geolocation으로 얻어온 좌표로 생성합니다
            message = '<div style="padding:5px;">여기애 계신가요?!</div>'; // 인포윈도우에 표시될 내용입니다

        // 마커와 인포윈도우를 표시합니다
        displayMarker(locPosition, message);

    });

} else { // HTML5의 GeoLocation을 사용할 수 없을때 마커 표시 위치와 인포윈도우 내용을 설정합니다

    var locPosition = new kakao.maps.LatLng(37.522262, 126.850055),
        message = 'geolocation을 사용할수 없어요..'

    displayMarker(locPosition, message);
}

function displayMarker(locPosition, message) {

    // 마커를 생성합니다
    var marker = new kakao.maps.Marker({
        map: map,
        position: locPosition
    });

    var iwContent = message, // 인포윈도우에 표시할 내용
        iwRemoveable = true;

    // 인포윈도우를 생성합니다
    var infowindow = new kakao.maps.InfoWindow({
        content : iwContent,
        removable : iwRemoveable
    });
}

```

```

    // 인포윈도우를 마커위에 표시합니다
    infowindow.open(map, marker);

    // 지도 중심좌표를 접속위치로 변경합니다
    map.setCenter(locPosition);
}

```

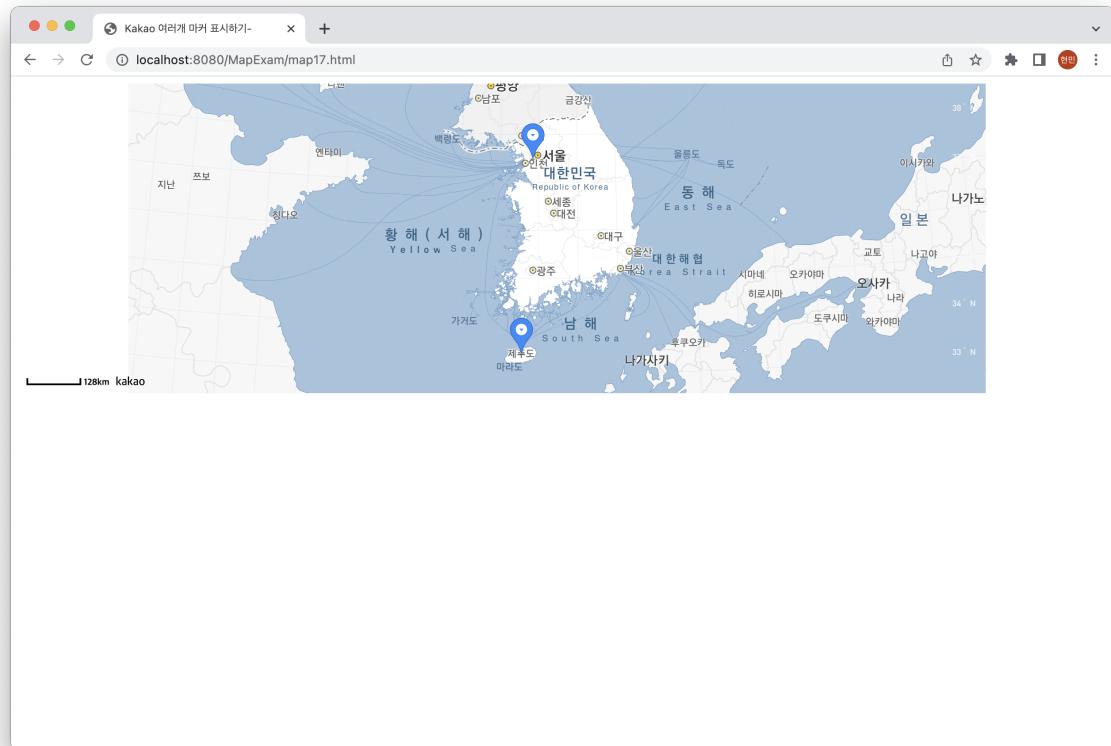
## 17. 여러개 마커 표시하기

```

//마커
//마커 위치
var positions = [
  {
    title:'카카오',
    latlng: new kakao.maps.LatLng(37.558043, 126.919002)
  },
  {
    title:'집',
    latlng: new kakao.maps.LatLng(37.522262, 126.850055)
  }
];

//마커 생성
for(var i=0; i<positions.length; i++){
  var marker = new kakao.maps.Marker({
    map: map,
    position:positions[i].latlng,
    title:positions[i].title
  });
}

```



## 18. 여러개 마커 제거하기

```

//지도를 마우스로 클릭했을때 이벤트 발생 처리
kakao.maps.event.addListener(map, 'click', function(mouseEvent){
    //클릭한 그 위치에 마커 표시
    addMarker(mouseEvent.latLng);
});

//마우스 클릭으로 생성된 마커들을 담을 배열
var markers = [];

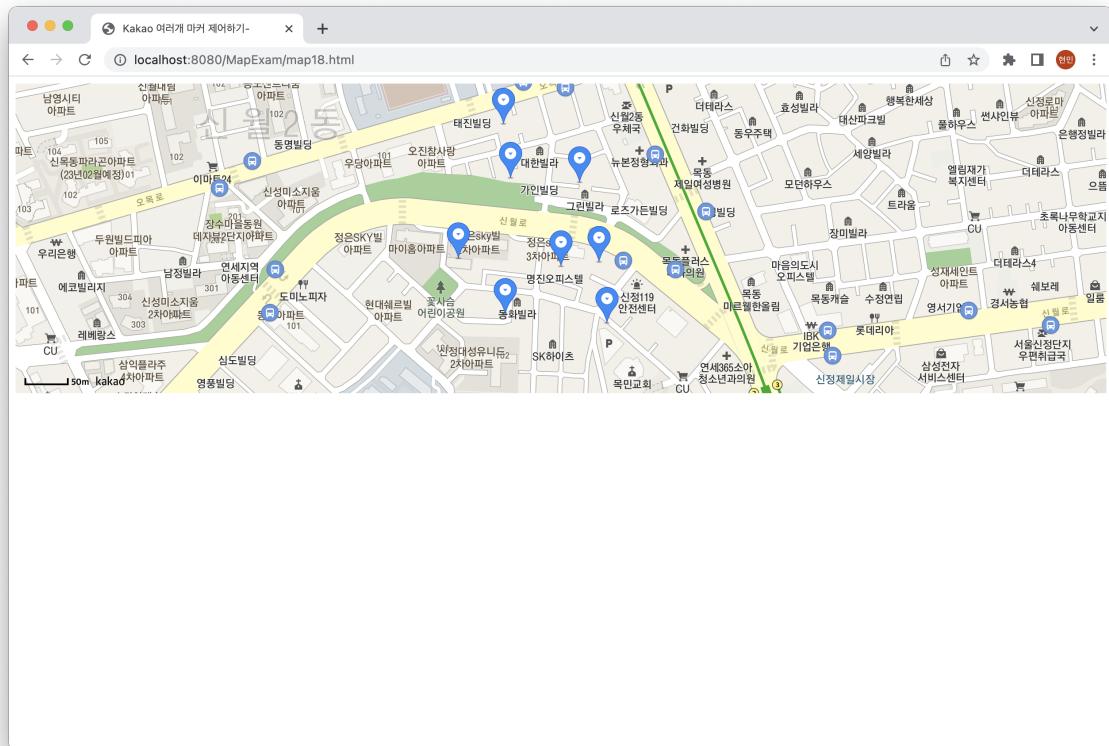
addMarker(new kakao.maps.LatLng(33.450701, 126.570667));

//마커 생성 함수
function addMarker(position){
    var marker = new kakao.maps.Marker({
        position:position
    });

    //클릭한 마커 지도에 올리기
    marker.setMap(map);

    //마커를 배열에 저장
    markers.push(marker);
}

```



## 19. 선의 거리 계산하기

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>선의 거리 계산하기</title>
    <style>
        .dot {overflow:hidden;float:left;width:12px;height:12px;background: url('https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/mini...
        .dotOverlay {position:relative;bottom:10px;border-radius:6px;border: 1px solid #ccc; border-bottom:2px solid #ddd;float:left;font-size:...
        .dotOverlay:nth-of-type(n) {border:0; box-shadow:0px 1px 2px #888; }
        .number {font-weight:bold;color:#ee6152; }
        .dotOverlay:after {content:'';position:absolute;margin-left:-6px;left:50%;bottom:-8px;width:11px;height:8px;background:url('https://t1...
        .distanceInfo {position:relative;top:5px;left:5px;list-style:none; margin:0; }
        .distanceInfo .label {display:inline-block; width:50px; }
    </style>

```

```

.distanceInfo:after {content:none;}
</style>
</head>
<body>
<div id="map" style="width:100%;height:350px;"></div>
<p>
<em>지도를 마우스로 클릭하면 선 그리기가 시작되고<br>오른쪽 마우스를 클릭하면 선 그리기가 종료됩니다</em>
</p>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=fc7abe9b9479ffa297474094a4964dcf"></script>
<script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };
var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다

var drawingFlag = false; // 선이 그려지고 있는 상태를 가지고 있을 변수입니다
var moveLine; // 선이 그려지고 있을때 마우스 움직임에 따라 그려질 선 객체입니다
var clickLine // 마우스로 클릭한 좌표로 그려질 선 객체입니다
var distanceOverlay; // 선의 거리정보를 표시할 커스텀오버레이 입니다
var dots = {} // 선이 그려지고 있을때 클릭할 때마다 클릭 지점과 거리를 표시하는 커스텀 오버레이 배열입니다.

// 지도에 클릭 이벤트를 등록합니다
// 지도를 클릭하면 선 그리기가 시작됩니다 그려진 선이 있으면 지우고 다시 그립니다
kakao.maps.event.addListener(map, 'click', function(mouseEvent) {

    // 마우스로 클릭한 위치입니다
    var clickPosition = mouseEvent.latLng;

    // 지도 클릭이벤트가 발생했는데 선을 그리고있는 상태가 아니면
    if (!drawingFlag) {

        // 상태를 true로, 선이 그리고있는 상태로 변경합니다
        drawingFlag = true;

        // 지도 위에 선이 표시되고 있다면 지도에서 제거합니다
        deleteClickLine();

        // 지도 위에 커스텀오버레이가 표시되고 있다면 지도에서 제거합니다
        deleteDistance();

        // 지도 위에 선을 그리기 위해 클릭한 지점과 해당 지점의 거리정보가 표시되고 있다면 지도에서 제거합니다
        deleteCircleDot();

        // 클릭한 위치를 기준으로 선을 생성하고 지도위에 표시합니다
        clickLine = new kakao.maps.Polyline({
            map: map, // 선을 표시할 지도입니다
            path: [clickPosition], // 선을 구성하는 좌표 배열입니다 클릭한 위치를 넣어줍니다
            strokeWeight: 3, // 선의 두께입니다
            strokeColor: '#db4040', // 선의 색깔입니다
            strokeOpacity: 1, // 선의 불투명도입니다 0에서 1 사이값이며 0에 가까울수록 투명합니다
            strokeStyle: 'solid' // 선의 스타일입니다
        });

        // 선이 그려지고 있을 때 마우스 움직임에 따라 선이 그려질 위치를 표시할 선을 생성합니다
        moveLine = new kakao.maps.Polyline({
            strokeWeight: 3, // 선의 두께입니다
            strokeColor: '#db4040', // 선의 색깔입니다
            strokeOpacity: 0.5, // 선의 불투명도입니다 0에서 1 사이값이며 0에 가까울수록 투명합니다
            strokeStyle: 'solid' // 선의 스타일입니다
        });

        // 클릭한 지점에 대한 정보를 지도에 표시합니다
        displayCircleDot(clickPosition, 0);

    } else { // 선이 그려지고 있는 상태이면

        // 그려지고 있는 선의 좌표 배열을 얻어옵니다
        var path = clickLine.getPath();

        // 좌표 배열에 클릭한 위치를 추가합니다
        path.push(clickPosition);

        // 다시 선에 좌표 배열을 설정하여 클릭 위치까지 선을 그리도록 설정합니다
        clickLine.setPath(path);

        var distance = Math.round(clickLine.getLength());
        displayCircleDot(clickPosition, distance);
    }
});

// 지도에 마우스무브 이벤트를 등록합니다
// 선을 그리고있는 상태에서 마우스무브 이벤트가 발생하면 그려질 선의 위치를 동적으로 보여주도록 합니다

```

```

kakao.maps.event.addListener(map, 'mousemove', function (MouseEvent) {
    // 지도 마우스무브 이벤트가 발생했는데 선을 그리고있는 상태이면
    if (drawingFlag){
        // 마우스 커서의 현재 위치를 얻어옵니다
        var mousePosition = MouseEvent.latLng;

        // 마우스 클릭으로 그려진 선의 좌표 배열을 얻어옵니다
        var path = clickLine.getPath();

        // 마우스 클릭으로 그려진 마지막 좌표와 마우스 커서 위치의 좌표로 선을 표시합니다
        var movepath = [path[path.length-1], mousePosition];
        moveLine.setPath(movepath);
        moveLine.setMap(map);

        var distance = Math.round(clickLine.getLength() + moveLine.getLength()), // 선의 총 거리를 계산합니다
            content = '<div class="dotOverlay distanceInfo">총거리 <span class="number">' + distance + '</span>m</div>'; // 커스텀오버레이

        // 거리정보를 지도에 표시합니다
        showDistance(content, mousePosition);
    }
});

// 지도에 마우스 오른쪽 클릭 이벤트를 등록합니다
// 선을 그리고있는 상태에서 마우스 오른쪽 클릭 이벤트가 발생하면 선 그리기를 종료합니다
kakao.maps.event.addListener(map, 'rightclick', function (MouseEvent) {
    // 지도 오른쪽 클릭 이벤트가 발생했는데 선을 그리고있는 상태이면
    if (drawingFlag) {
        // 마우스무브로 그려진 선은 지도에서 제거합니다
        moveLine.setMap(null);
        moveLine = null;

        // 마우스 클릭으로 그린 선의 좌표 배열을 얻어옵니다
        var path = clickLine.getPath();

        // 선을 구성하는 좌표의 개수가 2개 이상이면
        if (path.length > 1) {
            // 마지막 클릭 지점에 대한 거리 정보 커스텀 오버레이를 지웁니다
            if (dots[dots.length-1].distance) {
                dots[dots.length-1].distance.setMap(null);
                dots[dots.length-1].distance = null;
            }

            var distance = Math.round(clickLine.getLength()), // 선의 총 거리를 계산합니다
                content = getTimeHTML(distance); // 커스텀오버레이에 추가될 내용입니다

            // 그려진 선의 거리정보를 지도에 표시합니다
            showDistance(content, path[path.length-1]);
        } else {
            // 선을 구성하는 좌표의 개수가 1개 이하이면
            // 지도에 표시되고 있는 선과 정보들을 지도에서 제거합니다.
            deleteClickLine();
            deleteCircleDot();
            deleteDistnce();
        }
    }

    // 상태를 false로, 그리지 않고 있는 상태로 변경합니다
    drawingFlag = false;
}
});

// 클릭으로 그려진 선을 지도에서 제거하는 함수입니다
function deleteClickLine() {
    if (clickLine) {
        clickLine.setMap(null);
        clickLine = null;
    }
}

// 마우스 드래그로 그려지고 있는 선의 총거리 정보를 표시하기
// 마우스 오른쪽 클릭으로 선 그리기 종료했을 때 선의 정보를 표시하는 커스텀 오버레이를 생성하고 지도에 표시하는 함수입니다
function showDistance(content, position) {
    if (distanceOverlay) { // 커스텀오버레이가 생성된 상태이면

        // 커스텀 오버레이의 위치와 표시할 내용을 설정합니다
        distanceOverlay.setPosition(position);
        distanceOverlay.setContent(content);

    } else { // 커스텀 오버레이가 생성되지 않은 상태이면

```

```

// 커스텀 오버레이를 생성하고 지도에 표시합니다
distanceOverlay = new kakao.maps.CustomOverlay({
    map: map, // 커스텀오버레이를 표시할 지도입니다
    content: content, // 커스텀오버레이에 표시할 내용입니다
    position: position, // 커스텀오버레이를 표시할 위치입니다.
    xAnchor: 0,
    yAnchor: 0,
    zIndex: 3
});
}

// 그려지고 있는 선의 총거리 정보와
// 선 그리기 종료됐을 때 선의 정보를 표시하는 커스텀 오버레이를 삭제하는 함수입니다
function deleteDistance () {
    if (distanceOverlay) {
        distanceOverlay.setMap(null);
        distanceOverlay = null;
    }
}

// 선이 그려지고 있는 상태일 때 지도를 클릭하면 호출하여
// 클릭 지점에 대한 정보 (동그라미와 클릭 지점까지의 총거리)를 표출하는 함수입니다
function displayCircleDot(position, distance) {

    // 클릭 지점을 표시할 빨간 동그라미 커스텀오버레이를 생성합니다
    var circleOverlay = new kakao.maps.CustomOverlay({
        content: '<span class="dot"></span>',
        position: position,
        zIndex: 1
    });

    // 지도에 표시합니다
    circleOverlay.setMap(map);

    if (distance > 0) {
        // 클릭한 지점까지의 그려진 선의 총 거리를 표시할 커스텀 오버레이를 생성합니다
        var distanceOverlay = new kakao.maps.CustomOverlay({
            content: '<div class="dotOverlay">거리 <span class="number">' + distance + '</span>m</div>',
            position: position,
            yAnchor: 1,
            zIndex: 2
        });

        // 지도에 표시합니다
        distanceOverlay.setMap(map);
    }

    // 배열에 추가합니다
    dots.push({circle:circleOverlay, distance: distanceOverlay});
}

// 클릭 지점에 대한 정보 (동그라미와 클릭 지점까지의 총거리)를 지도에서 모두 제거하는 함수입니다
function deleteCircleDot() {
    var i;

    for ( i = 0; i < dots.length; i++ ){
        if (dots[i].circle) {
            dots[i].circle.setMap(null);
        }

        if (dots[i].distance) {
            dots[i].distance.setMap(null);
        }
    }

    dots = [];
}

// 마우스 우클릭 하여 선 그리기가 종료됐을 때 호출하여
// 그려진 선의 총거리 정보와 거리에 대한 도보, 자전거 시간을 계산하여
// HTML Content를 만들어 리턴하는 함수입니다
function getDistanceHTML(distance) {

    // 도보의 시속은 평균 4km/h 이고 도보의 분속은 67m/min입니다
    var walkTime = distance / 67 | 0;
    var walkHour = '', walkMin = '';

    // 계산한 도보 시간이 60분 보다 크면 시간으로 표시합니다
    if (walkTime > 60) {
        walkHour = '<span class="number">' + Math.floor(walkTime / 60) + '</span>시간 ';
    }
    walkMin = '<span class="number">' + walkTime % 60 + '</span>분'

    // 자전거의 평균 시속은 16km/h 이고 이것을 기준으로 자전거의 분속은 267m/min입니다
    var bicycleTime = distance / 227 | 0;
}

```

```

var bycicleHour = '';
var bycicleMin = '';

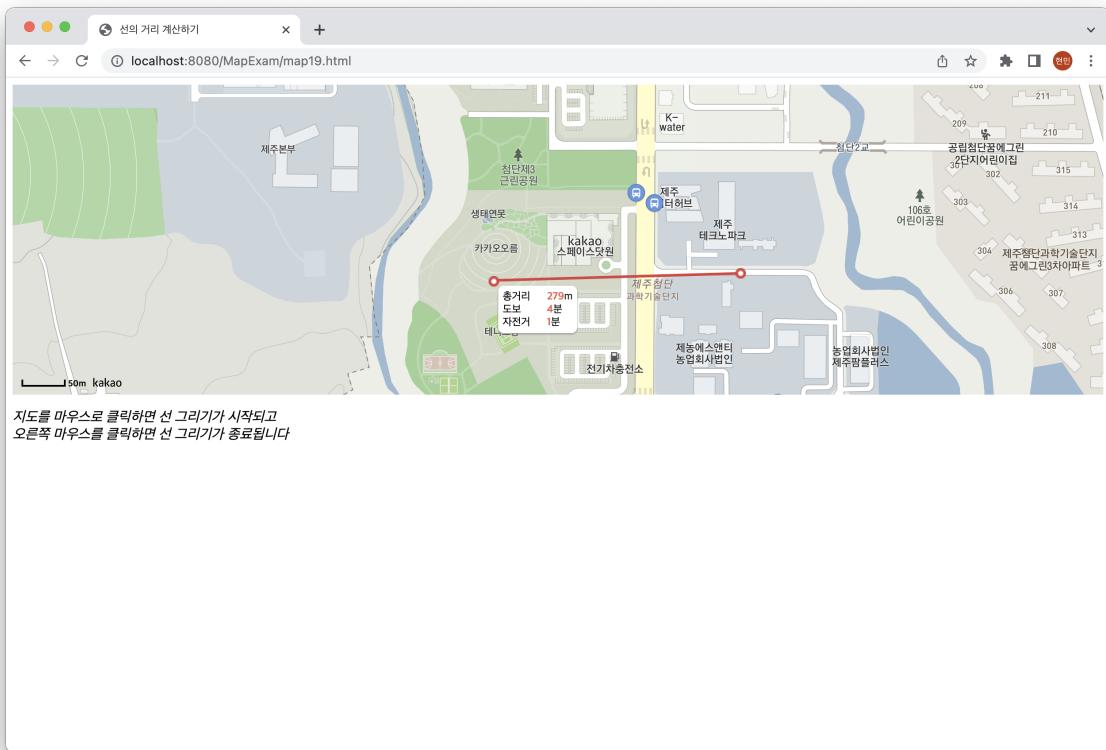
// 계산한 자전거 시간이 60분 보다 크면 시간으로 표출합니다
if (bycicleTime > 60) {
    bycicleHour = '<span class="number">' + Math.floor(bycicleTime / 60) + '</span>시간 '
}
bycicleMin = '<span class="number">' + bycicleTime % 60 + '</span>분'

// 거리와 도보 시간, 자전거 시간을 가지고 HTML Content를 만들어 리턴합니다
var content = '<ul class="dotOverlay distanceInfo">';
content += '    <li>';
content += '        <span class="label">총거리</span><span class="number">' + distance + '</span>m';
content += '    </li>';
content += '    <li>';
content += '        <span class="label">도보</span>' + walkHour + walkMin;
content += '    </li>';
content += '    <li>';
content += '        <span class="label">자전거</span>' + bycicleHour + bycicleMin;
content += '    </li>';
content += '</ul>';

return content;
}

</script>
</body>
</html>

```



## 로드뷰

### 20. 로드뷰 생성하기

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>로드뷰</title>
</head>

```

```

<body>
<div id="map" style="width:100%;height:350px;"></div>
<div id="road" style="width:100%;height:350px;"></div>
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=fc7abe9b9479ffa297474094a4964dcf"></script>
<script>
var container = document.getElementById('map');

var options = {
  center: new kakao.maps.LatLng(37.522262, 126.850055),
  level: 3
};

var map = new kakao.maps.Map(container, options);

//로드뷰는 지도와는 별도로 그릴수 있다. 지도 없이 로드뷰만 생성도 가능하다.
var roadviewContainer = document.getElementById('road');

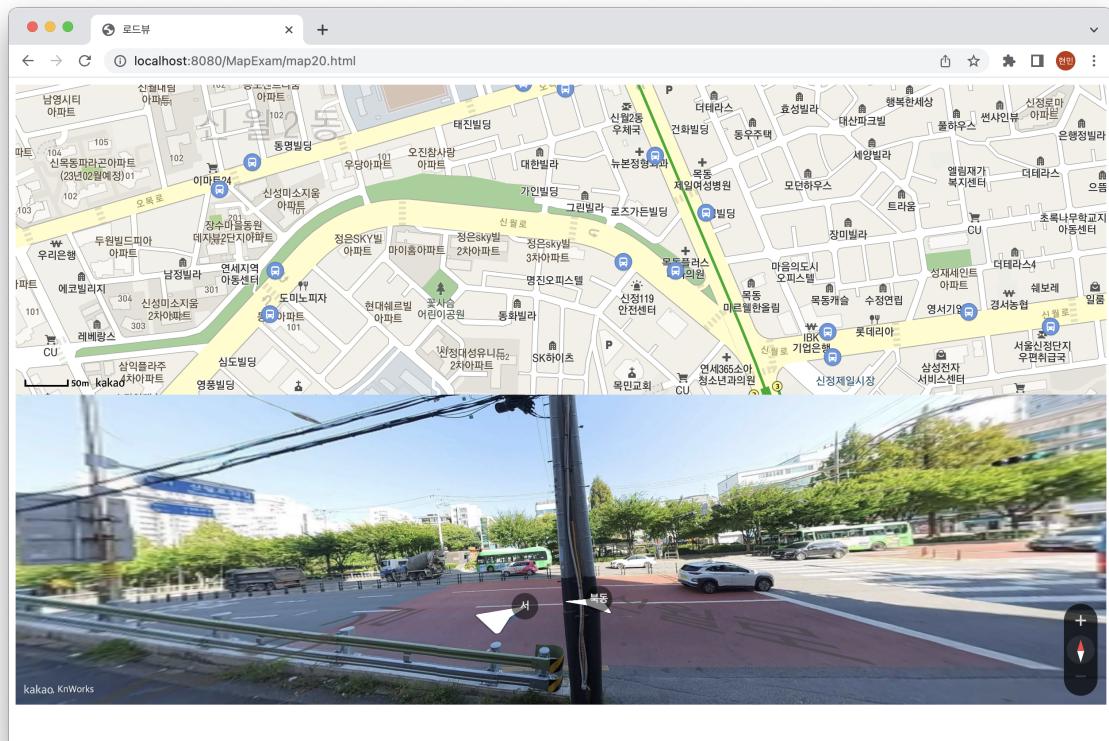
//로드뷰 객체 생성
var roadview = new kakao.maps.Roadview(roadviewContainer);

//해당 좌표로부터 로드뷰 파노라마 ID를 가져올 로드뷰 helper 객체
var roadviewClient = new kakao.maps.RoadviewClient();

var position = new kakao.maps.LatLng(37.522262, 126.850055);

//특정 위치의 좌표와 가까운 로드뷰 추출하여 로드뷰를 띄운다
roadviewClient.getNearestPanoId(position, 50, function(panoId){
  roadview.setPanoId(panoId, position);
})
</script>
</body>
</html>

```



## 라이브러리

21. 키워드로 장소 검색하기

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>키워드로 장소검색</title>
</head>
<body>

<div id="map" style="width:100%;height:350px;"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=fc7abe9b9479ffa297474094a4964dcf&libraries=services"></script>

<script>
var container = document.getElementById('map');

var options = {
  center: new kakao.maps.LatLng(37.522262, 126.850055),
  level: 3
};

var map = new kakao.maps.Map(container, options);

//마커를 클릭하면 그 장소에 대한 정보를 보여주는 인포윈도우
var infowindow = new kakao.maps.InfoWindow({zIndex:1});

//장소 검색 객체 생성
var ps = new kakao.maps.services.Places();

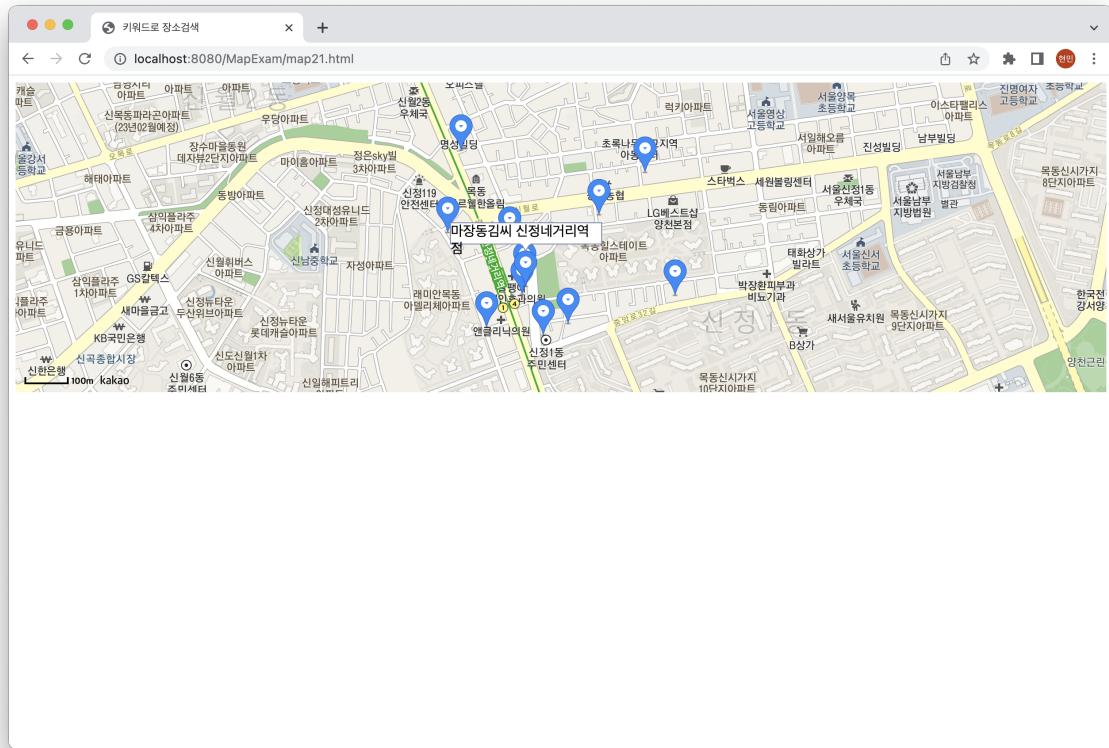
//검색할 장소
ps.keywordSearch('신정네거리 맷집', placesSearchCB);

function placesSearchCB(data, status, pagination){
  if(status==kakao.maps.services.Status.OK){
    var bounds = new kakao.maps.LatLngBounds();

    for(var i=0; i<data.length; i++){
      displayMarker(data[i]);
      bounds.extend(new kakao.maps.LatLng(data[i].y, data[i].x));
    }
    //검색된 장소들의 위치를 기준으로 지도 범위 재설정(크기가 확대/축소될 수 있다.)
    map.setBounds(bounds);
  }
}
//지도에 마커 표시 함수
function displayMarker(place){
  var marker = new kakao.maps.Marker({
    map:map,
    position: new kakao.maps.LatLng(place.y, place.x)
  });
  kakao.maps.event.addListener(marker, 'click', function(){
    infowindow.setContent(place.place_name);
    infowindow.open(map, marker);
  });
}

</script>
</body>
</html>

```



## 22. 주소로 장소 표시하기

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<% String addr = request.getParameter("addr"); %>

<div id="map" style="width:100%;height:350px;"></div>
<div id="roadview" style="width:100%;height:350px;"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=fc7abe9b9479ffa297474094a4964dcf&libraries=services"></script>
<script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

// 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();

// 주소로 좌표를 검색합니다
geocoder.addressSearch('<%=addr%>', function(result, status) {

    // 정상적으로 검색이 완료됐으면
    if (status === kakao.maps.services.Status.OK) {

        var coords = new kakao.maps.LatLng(result[0].y, result[0].x);

        // 결과값으로 받은 위치를 마커로 표시합니다
        var marker = new kakao.maps.Marker({
            map: map,
    }

```

```

        position: coords
    });

    // 인포윈도우로 장소에 대한 설명을 표시합니다
    var infowindow = new kakao.maps.InfoWindow({
        content: '<div style="width:150px;text-align:center;padding:6px 0;">우리회사</div>'
    });
    infowindow.open(map, marker);

    // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
    map.setCenter(coords);

    // 로드 뷔는 지도와는 별도로 그릴수 있다. 지도 없이 로드뷰만 생성도 가능하다.

    var roadviewContainer = document.getElementById('roadview'); //로드뷰를 표시할 div
    var roadview = new kakao.maps.Roadview(roadviewContainer); //로드뷰 객체
    var roadviewClient = new kakao.maps.RoadviewClient(); //좌표로부터 로드뷰 파노ID를 가져올 로드뷰 helper객체

/*   var position = new kakao.maps.LatLng(37.558043, 126.919002); */

    // 특정 위치의 좌표와 가까운 로드뷰의 panoId를 추출하여 로드뷰를 띄운다.
    roadviewClient.getNearestPanoId(coords, 50, function(panoId) {
        roadview.setPanoId(panoId, coords); //panoId와 중심좌표를 통해 로드뷰 실행
    });

}
</script>
</body>
</html>

```

