

Class 230119

카카오 지도

- 시작 하기

Kakao 지도 API

Kakao 지도 Javascript API 는 키 발급을 받아야 사용할 수 있습니다. 그리고 키를 발급받기 위해서는 카카오 계정이 필요합니다. 키 발급에는 아래 과정이 필요합니다. 1. (<https://developers.kakao.com>) 접속2. 개발자 등록 및 앱 생성3. 웹 플랫폼 추가: 앱 선택 - [플랫폼] - [Web 플랫폼 등록] - 사이트 도메인 등록4. 사이트 도메인 등록:

📍 <https://apis.map.kakao.com/web/guide/>



- 전체 문서

Kakao 지도 API

지도 객체 `kakao.maps.Map(container, options)` Parameters `setCenter(latlng)` 지도의 중심 좌표를 설정한다. 지도 이동시키기 샘플보기 Parameters `getCenter()` 지도의 중심 좌표를 반환한다. 지도 정보 얻어오기 샘플보기 Returns `setLevel(level, options)` Parameters `getLevel()` 지도의 확대 수준을 반환한다. 지도 정보 얻어오기 샘플보기

📍 <https://apis.map.kakao.com/web/documentation/>



- 샘플 코드

Kakao 지도 API

Kakao 지도 API를 이용해보세요. 다양한 샘플과 상세한 매뉴얼로 개발자들의 시간을 아껴드리겠습니다!

📍 <https://apis.map.kakao.com/web/sample/>



지도

- ▼ 지도 생성하기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>Kakao 지도 시작하기</title>
</head>
<body>
  <div id="map" style="width:700px;height:700px;"></div>
  <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb378f1cb2"></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>
</body>
</html>
```

Parameters

- **container** Node : 지도가 표시될 HTML element
- **options** Object

- `center` *LatLng* : 중심 좌표 (필수)
- `level` *Number* : 확대 수준 (기본값: 3)
- `mapTypeId` *MapTypeId* : 지도 종류 (기본값: 일반 지도)
- `draggable` *Boolean* : 마우스 드래그, 휠, 모바일 터치를 이용한 시점 변경(이동, 확대, 축소) 가능 여부
- `scrollwheel` *Boolean* : 마우스 휠, 모바일 터치를 이용한 확대 및 축소 가능 여부
- `disableDoubleClick` *Boolean* : 더블클릭 이벤트 및 더블클릭 확대 가능 여부
- `disableDoubleClickZoom` *Boolean* : 더블클릭 확대 가능 여부
- `projectionId` *String* : 투영법 지정 (기본값: kakao.maps.ProjectionId.WCONG)
- `tileAnimation` *Boolean* : 지도 타일 애니메이션 설정 여부 (기본값: true)
- `keyboardShortcuts` *Boolean | Object* : 키보드의 방향키와 +, - 키로 지도 이동, 확대, 축소 가능 여부 (기본값: false)
 - `speed` *Number* : 지도 이동 속도

▼ 지도 레벨 바꾸기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>Kakao 지도 레벨</title>
</head>
<body>
  <div id="map" style="width:700px;height:700px;"></div>
  <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>
  <script>
    var container = document.getElementById('map');
    var options = {
      center: new kakao.maps.LatLng(33.450701, 126.570667),
      level: 15 // 가까이 표시 << 1 ~ 15 >> 멀리 표시
    };

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

▼ 지도 정보 얻어오기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>Kakao 지도 레벨</title>
</head>
<body>
  <div id="map" style="width:700px;height:700px;"></div>
  <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>
  <script>
    var container = document.getElementById('map');
    var options = {
      center: new kakao.maps.LatLng(33.450701, 126.570667),
      level: 12
    };

    var map = new kakao.maps.Map(container, options);
    getInfo();
    function getInfo() {
      // 지도에서 현재 중심 좌표값 얻기
      var center = map.getCenter();
      // 지도에서 현재 레벨 얻기
      var level = map.getLevel();
      // 지도에서 현재 영역 얻기
      var bounds = map.getBounds();
      // 지도의 영역 정보 얻기 - 남서 북동
      var boundsStr = bounds.toString();

      var msg = '위도 : ' + center.getLat();
      msg += '경도 : ' + center.getLng();
      msg += '레벨 : ' + level;
    }
  </script>
</body>
</html>
```

```

        msg += '영역 정보 : ' + boundsStr;

        console.log(msg);
    }

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

```

▼ 지도에 컨트롤 올리기

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Kakao 지도 레벨</title>
</head>
<body>
    <div id="map" style="width:700px;height:700px;"></div>
    <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>
    <script>
        var container = document.getElementById('map');
        var options = {
            center: new kakao.maps.LatLng(33.450701, 126.570667),
            level: 12
        };

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

        // 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);
    </script>
</body>
</html>

```

▼ 지도 이동 막기

```
map.dwtDraggable(false);
```

▼ 지도 확대 축소 막기

```
map.setZoomable(false);
```

▼ 지도 범위 재설정 하기

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Kakao 지도 범위 재설정</title>
</head>
<body>
    <div id="map" style="width:700px;height:700px;"></div>
    <button onclick="setBounds()">범위 재설정</button>
    <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>
    <script>
        var container = document.getElementById('map');
        var options = {
            center: new kakao.maps.LatLng(33.450701, 126.570667),
            level: 12
        };

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

        // 표시할 좌표 값을 변수 설정
        var points = [
            new kakao.maps.LatLng(33.452278, 126.567803),
            new kakao.maps.LatLng(33.452645, 126.523465),
            new kakao.maps.LatLng(33.452345, 126.544564),
        ];

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

        //세 개의 좌표 값을 지도에 올리기
        var i, marker;
        for (i = 0; points.length; i++) {

```

```

//하나씩 지도에 추가
marker = new kakao.maps.Marker({
  position:points[i]
});
marker.setMap(map);

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

// 범위 재설정 처리 함수
function setBounds() {
  map.setBounds(bounds);
}
</script>
</body>
</html>

```

▼ 중심좌표 변경 이벤트 등록하기 (center_changed 이벤트)

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>kakao 지도 중심 좌표값 변경</title>
</head>
<body>

<div id="map" style="width:700px;height:700px;"></div>
<div id="result"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>

<script>
var container = document.getElementById('map');
var options = {
  center: new kakao.maps.LatLng(33.450701, 126.570667),
  level: 3 // 지도의 크기를 설정 예) 10 이 서울 전체, 제주 전체
};

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

// 지도 변경 시 중심 좌표
kakao.maps.event.addListener(map, 'center_changed', function() {
  // 지도의 레벨을 얻어옴
  var level = map.getLevel();
  // 지도의 중심좌표를 얻어옴
  var latlng = map.getCenter();

  var message = '<p>지도 레벨은 ' + level + ' 이고</p>';
  message += '<p>중심 좌표는 위도 ' + latlng.getLat() + ', 경도 ' + latlng.getLng() + ' 입니다</p>';

  var resultDiv = document.getElementById('result');
  resultDiv.innerHTML = message;
}
);
</script>

</body>
</html>

```

▼ 타일로드 이벤트 등록하기 (tilesloaded 이벤트)

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>kakao 지도 타일로드 이벤트 등록</title>
</head>
<body>

<div id="map" style="width:700px;height:700px;"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>

<script>
var container = document.getElementById('map');
var options = {
  center: new kakao.maps.LatLng(37.5494147, 127.0818126),
  level: 3
}

```

```

};

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

// 마커 객체 생성
var marker = new kakao.maps.Marker();
// 마커 지도에 표시
kakao.maps.event.addListener(map, 'tilesloaded', displayMarker);
function displayMarker() {
    marker.setPosition(map.getCenter());
    marker.setMap(map);
}
</script>

</body>
</html>

```

오버레이

▼ 마커 생성하기

```

// 마커를 표시할 위치
var markerPosition = new kakao.maps.LatLng(37.5494147, 127.0818126);
// 마커 이미지 아이콘 생성
var icon = new kakao.maps.MarkerImage(
    'db.png',
    new kakao.maps.Size(31, 35),
    {
        offset: new kakao.maps.Point(16, 34),
        alt: "마커 이미지",
        shape: "poly",
        coords: "1,20,1,9,5,2,10,0,21,0,27,3,30,9,30,20,17,33,14,33"
    }
);
// 마커 생성
var marker = new kakao.maps.Marker({
    map: map, // 맵 지정
    position: markerPosition, // 위치 지정
    draggable: true, // 마커 드래그 가능하게 함
    image: icon // 마커 이미지 변경
});

```

Parameters

- options** *Object*
 - map** *Map* | *Roadview* : 마커가 올라갈 지도 또는 로드뷰
 - position** *LatLng* | *Viewpoint* : 마커의 좌표 또는 로드뷰에서의 시점
 - image** *MarkerImage* : 마커의 이미지
 - title** *String* : 마커 엘리먼트의 타이틀 속성 값 (툴팁)
 - draggable** *Boolean* : 드래그 가능한 마커, 로드뷰에 올릴 경우에는 유효하지 않다
 - clickable** *Boolean* : 클릭 가능한 마커
 - zIndex** *Number* : 마커 엘리먼트의 z-index 속성 값
 - opacity** *Number* : 마커 투명도 (0-1)
 - altitude** *Number* : 로드뷰에 올라있는 마커의 높이 값(m 단위)
 - range** *Number* : 로드뷰 상에서 마커의 가시반경(m 단위), 두 지점 사이의 거리가 지정한 값보다 멀어지면 마커는 로드뷰에서 보이지 않게 된다

▼ 인포윈도우 생성하기

```

// 정보창 생성
var infowindow = new kakao.maps.InfoWindow({
    map: map,
    position: new kakao.maps.LatLng(37.5494147, 127.0818126),

```

```

    content: '어린이 대공원'
  });

```

Parameters

- options** *Object*
 - content** *Node | String* : 엘리먼트 또는 HTML 문자열 형태의 내용
 - disableAutoPan** *Boolean* : 인포윈도우를 열 때 지도가 자동으로 패닝하지 않을지의 여부 (기본값: false)
 - map** *Map | Roadview* : 인포윈도우가 올라갈 지도 또는 로드뷰
 - position** *LatLng* : 인포윈도우의 좌표
 - removable** *Boolean* : 삭제 가능한 인포윈도우
 - zIndex** *Number* : 인포윈도우 엘리먼트의 z-index 속성 값
 - altitude** *Number* : 로드뷰에 올라있는 인포윈도우의 높이 값(m 단위)
 - range** *Number* : 로드뷰 상에서 인포윈도우의 가시반경(m 단위), 두 지점 사이의 거리가 지정한 값보다 멀어지면 인포윈도우는 보이지 않게 된다

▼ 마커에 인포윈도우 표시하기

```

// 위치 지정
var pos = new kakao.maps.LatLng(37.5494147, 127.0818126);
// 마커 생성
var marker = new kakao.maps.Marker({
  map: map,
  position: pos
});
// 정보창 생성
var infoWindow = new kakao.maps.InfoWindow({
  position: pos,
  content: '<h3>어린이 대공원</h3>',
});
infoWindow.open(map, marker);

```

▼ 마커에 마우스 이벤트 등록하기

```

// 위치 지정
var pos = new kakao.maps.LatLng(37.5494147, 127.0818126);
// 마커 생성
var marker = new kakao.maps.Marker({
  map: map,
  position: pos
});
// 마커에 마우스 이벤트 등록
kakao.maps.event.addListener(marker, 'mouseover', function() {
  infoWindow.open(map, marker);
});
kakao.maps.event.addListener(marker, 'mouseout', function() {
  infoWindow.close(map, marker);
});
// 인포 윈도우 생성
var infoWindow = new kakao.maps.InfoWindow({
  position: pos,
  content: 'hi'
});

```

▼ geolocation으로 마커 표시하기 (크롬 전용)

```

// 현재 위치를 가져옴
navigator.geolocation.getCurrentPosition(function(position) {
  var lat = position.coords.latitude //위도
  var lng = position.coords.longitude //경도
  var locPosition = new kakao.maps.LatLng(lat, lng);
  var msg = '위도 : ' + lat + '<br>'
    + '경도 : ' + lng;
  displayMarker(locPosition, msg);
});

```

```
});

function displayMarker(locPosition, msg) {
    var marker = new kakao.maps.Marker({
        map: map,
        position: locPosition
    });

    var infoWindow = new kakao.maps.InfoWindow({
        content: msg,
        removable: true
    });
    infoWindow.open(map, marker);
    map.setCenter(locPosition)
}
}
```

▼ 여러개 마커 표시하기

- 여러개 지정해 표시

```
var positions = [
    {
        title: '우리집',
        LatLng: new kakao.maps.LatLng(37.5494147, 127.0818126)
    },
    {
        title: '카카오',
        LatLng: new kakao.maps.LatLng(33.450701, 126.570667)
    }
];
for (let i = 0; i < positions.length; i++) {
    var marker = new kakao.maps.Marker({
        map: map,
        position: positions[i].LatLng,
        title: positions[i].title
    });
}
```

- 마우스 클릭 이벤트로 표시

```
var markerArr = new Array();
// 마우스 클릭 이벤트
kakao.maps.event.addListener(map, 'click', function(MouseEvent) {
    // 마커 표시
    addMarker(MouseEvent.latLng);
});
function addMarker(position) {
    var marker = new kakao.maps.Marker({
        map: map,
        position: position
    });
    markerArr.push(marker);
}
```

▼ 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/');}
        .dotOverlay {position:relative;bottom:10px;border-radius:6px;border: 1px solid #ccc;border-bottom:2px solid #ddd;float:left;font-size:12px;}
        .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.daumcdn.net/localimg/localimages/07/mapapidoc/');}
        .distanceInfo {position:relative;top:5px;left:5px;list-style:none;margin:0;}
        .distanceInfo .label {display:inline-block;width:50px;}
        .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=2af21256d2413713f5827fb37b8f1cb2"></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 객체 입니다
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();
            deleteDistance();

        }

        // 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 표시하거
// 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 표시하는 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 getTimeHTML(distance) {

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

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

  // 자전거의 평균 시속은 16km/h 이고 이것을 기준으로 자전거의 분속은 267m/min입니다
  var bycycleTime = distance / 227 | 0;
  var bycycleHour = '', bycycleMin = '';

  // 계산한 자전거 시간이 60분 보다 크면 시간으로 표시합니다
  if (bycycleTime > 60) {
    bycycleHour = '<span class="number">' + Math.floor(bycleTime / 60) + '</span>시간 '
  }
  bycycleMin = '<span class="number">' + bycycleTime % 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>' + bycycleHour + bycycleMin;
    content += '    </li>';
    content += '</ul>';

    return content;
}

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

```

로드뷰

▼ 로드뷰 생성하기

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>kakao 지도 로드뷰 생성</title>
</head>
<body>

    <div id="map" style="width:100%;height:500px;"></div>
    <div id="roadview" style="width:100%;height:500px;"></div>
    <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>

<script>
var container = document.getElementById('map');
var options = {
    center: new kakao.maps.LatLng(37.5494147, 127.0818126),
    level: 10
};

var map = new kakao.maps.Map(container, options);
// ----- 지도 생성 완료 -----//

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 helper 객체

var position = new kakao.maps.LatLng(33.450701, 126.570667);

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

```

Parameters

- **container** *Node* : 로드뷰를 생성할 엘리먼트
- **options** *Object*
 - **panoId** *Number* : 로드뷰 시작 지역의 고유 아이디 값.
 - **panoX** *Number* : panoId가 유효하지 않을 경우 지도좌표를 기반으로 데이터를 요청할 수평 좌표값.
 - **panoY** *Number* : panoId가 유효하지 않을 경우 지도좌표를 기반으로 데이터를 요청할 수직 좌표값.
 - **pan** *Number* : 로드뷰 처음 실행시에 바라봐야 할 수평 각. 0이 정북방향. (0_360)
 - **tilt** *Number* : 로드뷰 처음 실행시에 바라봐야 할 수직 각. (-90_90)
 - **zoom** *Number* : 로드뷰 줌 초기값. (-3_3)

▼ 로드뷰 도로를 이용하여 로드뷰 생성하기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>로드뷰 도로를 이용하여 로드뷰 생성하기</title>
  <style>
    .map_wrap {overflow:hidden;height:900px}
  </style>
</head>
<body>
<div class="map_wrap">
  <div id="mapWrapper" style="width:50%;height:100%;float:left">
    <div id="map" style="width:100%;height:100%"></div> <!-- 지도를 표시할 div 입니다 -->
  </div>
  <div id="rvWrapper" style="width:50%;height:100%;float:left">
    <div id="roadview" style="width:100%;height:100%"></div> <!-- 로드뷰를 표시할 div 입니다 -->
  </div>
</div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2"></script>
<script>
var mapWrapper = document.getElementById('mapWrapper'); //지도를 감싸고 있는 DIV태그

var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapCenter = new kakao.maps.LatLng(33.450422139819736 , 126.5709139924533), // 지도의 가운데 좌표
    mapOption = {
      center: mapCenter, // 지도의 중심좌표
      level: 3 // 지도의 확대 레벨
    };

// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);
map.addOverlayMapTypeId(kakao.maps.MapTypeId.ROADVIEW); //지도 위에 로드뷰 도로 올리기

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

toggleRoadview(mapCenter);

// 마커 이미지를 생성합니다.
var markImage = new kakao.maps.MarkerImage(
  'https://t1.daumcdn.net/localimg/localimages/07/2018/pc/roadview_minimap_wk_2018.png',
  new kakao.maps.Size(26, 46),
  {
    // 스프라이트 이미지를 사용합니다.
    // 스프라이트 이미지 전체의 크기를 지정하고
    spriteSize: new kakao.maps.Size(1666, 168),
    // 사용하고 싶은 영역의 좌상단 좌표를 입력합니다.
    // background-position으로 지정하는 값이며 부호는 반대입니다.
    spriteOrigin: new kakao.maps.Point(705, 114),
    offset: new kakao.maps.Point(13, 46)
  }
);

// 드래그가 가능한 마커를 생성합니다.
var rvMarker = new kakao.maps.Marker({
  image : markImage,
  position: mapCenter,
  draggable: true,
  map: map
});

//마커에 dragend 이벤트를 할당합니다
kakao.maps.event.addListener(rvMarker, 'dragend', function(mouseEvent) {
  var position = rvMarker.getPosition(); //현재 마커가 놓인 자리의 좌표
  toggleRoadview(position); //로드뷰를 토글합니다
});

//지도에 클릭 이벤트를 할당합니다
kakao.maps.event.addListener(map, 'click', function(mouseEvent){

  // 현재 클릭한 부분의 좌표를 리턴
  var position = mouseEvent.latLng;

  rvMarker.setPosition(position);
  toggleRoadview(position); //로드뷰를 토글합니다
});

//로드뷰 토글함수
function toggleRoadview(position){
```

```

//전달받은 좌표(position)에 가까운 로드뷰의 panoId를 추출하여 로드뷰를 띄웁니다
rvClient.getNearestPanoId(position, 50, function(panoId) {
    if (panoId === null) {
        rvContainer.style.display = 'none'; //로드뷰를 넣은 컨테이너를 숨깁니다
        mapWrapper.style.width = '100%';
        map.relayout();
    } else {
        mapWrapper.style.width = '50%';
        map.relayout(); //지도를 감싸고 있는 영역이 변경됨에 따라, 지도를 재배열합니다
        rvContainer.style.display = 'block'; //로드뷰를 넣은 컨테이너를 보이게합니다
        rv.setPanoId(panoId, position); //panoId를 통한 로드뷰 실행
        rv.relayout(); //로드뷰를 감싸고 있는 영역이 변경됨에 따라, 로드뷰를 재배열합니다
    }
});
}
</script>
</body>
</html>

```

라이브러리

▼ 키워드로 장소 검색하기

```

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

</head>
<body>
<div id="map" style="width:100%;height:900px;"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2&libraries=services"></script>
// 마커를 클릭하면 장소명을 표출할 인포윈도우 입니다
var infowindow = new kakao.maps.InfoWindow({zIndex:1});

var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(37.5494147, 127.0818126), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

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

// 장소 검색 객체를 생성합니다
var ps = new kakao.maps.services.Places();

// 키워드로 장소를 검색합니다
ps.keywordSearch('홍대 맛집', placesSearchCB);

// 키워드 검색 완료 시 호출되는 콜백함수 입니다
function placesSearchCB (data, status, pagination) {
    if (status === kakao.maps.services.Status.OK) {

        // 검색된 장소 위치를 기준으로 지도 범위를 재설정하기위해
        // LatLngBounds 객체에 좌표를 추가합니다
        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 장소 위치를 기준으로 지도 범위를 재설정합니다
        map.setBounds(bounds);
    }
}

// 지도에 마커를 표시하는 함수입니다
function displayMarker(place) {
    // 마커를 생성하고 지도에 표시합니다
    var marker = new kakao.maps.Marker({
        map: map,
        position: new kakao.maps.LatLng(place.y, place.x)
    });

    // 마커에 클릭이벤트를 등록합니다

```

```

        kakao.maps.event.addListener(marker, 'click', function() {
            // 마커를 클릭하면 장소명이 인포윈도우에 표출됩니다
            console.log(place);
            infowindow.setContent('<div style="padding:5px;font-size:12px;">' + place.place_name + '</div>'
                + '<div style="padding:5px;font-size:12px;">' + place.address_name + '</div>'
                + '<div style="padding:5px;font-size:12px;">' + place.road_address_name + '</div>'
                + '<div style="padding:5px;font-size:12px;">' + place.category_name + '</div>'
            );
            infowindow.open(map, marker);
        });
    }
</script>
</body>
</html>
</body>
</html>

```

▼ 키워드로 장소검색하고 목록으로 표출하기

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>키워드로 장소검색하고 목록으로 표출하기</title>
    <style>
        .map_wrap, .map_wrap * {margin:0;padding:0;font-family: 'Malgun Gothic', dotum, '돋움', sans-serif;font-size:12px;}
        .map_wrap a, .map_wrap a:hover, .map_wrap a:active{color:#000;text-decoration: none;}
        .map_wrap {position:relative;width:100%;height:900px;}
        #menu_wrap {position:absolute;top:0;left:0;bottom:0;width:250px;margin:10px 0 30px 10px;padding:5px;overflow-y:auto;background:rgb(255,255,255);background-color:#fff;}
        #menu_wrap hr {display: block; height: 1px;border: 0; border-top: 2px solid #5F5F5F;margin:3px 0;}
        #menu_wrap .option{text-align: center;}
        #menu_wrap .option p {margin:10px 0;}
        #menu_wrap .option button {margin-left:5px;}
        #placesList li {list-style: none;}
        #placesList .item {position:relative;border-bottom:1px solid #888;overflow: hidden;cursor: pointer;min-height: 65px;}
        #placesList .item span {display: block;margin-top:4px;}
        #placesList .item h5, #placesList .item .info {text-overflow: ellipsis;overflow: hidden;white-space: nowrap;}
        #placesList .item .info{padding:10px 0 10px 55px;}
        #placesList .info .gray {color:#8a8a8a;}
        #placesList .info .jibun {padding-left:26px;background:url(https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/places_jibun.png);}
        #placesList .info .tel {color:#009900;}
        #placesList .item .markerbg {float:left;position:absolute;width:36px; height:37px;margin:10px 0 0 10px;background:url(https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/markerbg.png);background-position: 0 -10px;}
        #placesList .item .marker_1 {background-position: 0 -10px;}
        #placesList .item .marker_2 {background-position: 0 -56px;}
        #placesList .item .marker_3 {background-position: 0 -102px;}
        #placesList .item .marker_4 {background-position: 0 -148px;}
        #placesList .item .marker_5 {background-position: 0 -194px;}
        #placesList .item .marker_6 {background-position: 0 -240px;}
        #placesList .item .marker_7 {background-position: 0 -286px;}
        #placesList .item .marker_8 {background-position: 0 -332px;}
        #placesList .item .marker_9 {background-position: 0 -378px;}
        #placesList .item .marker_10 {background-position: 0 -423px;}
        #placesList .item .marker_11 {background-position: 0 -470px;}
        #placesList .item .marker_12 {background-position: 0 -516px;}
        #placesList .item .marker_13 {background-position: 0 -562px;}
        #placesList .item .marker_14 {background-position: 0 -608px;}
        #placesList .item .marker_15 {background-position: 0 -654px;}
        #pagination {margin:10px auto;text-align: center;}
        #pagination a {display:inline-block;margin-right:10px;}
        #pagination .on {font-weight: bold; cursor: default;color:#777;}
    </style>
</head>
<body>
    <div class="map_wrap">
        <div id="map" style="width:100%;height:100%;position:relative;overflow:hidden;"></div>

        <div id="menu_wrap" class="bg_white">
            <div class="option">
                <div>
                    <form onsubmit="searchPlaces(); return false;">
                        키워드 : <input type="text" value="이태원 맛집" id="keyword" size="15">
                        <button type="submit">검색하기</button>
                    </form>
                </div>
            </div>
            <hr>
            <ul id="placesList"></ul>
            <div id="pagination"></div>
        </div>
    </div>

    <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2&libraries=services"></script>

```

```

// 마커를 담을 배열입니다
var markers = [];

var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(37.566826, 126.9786567), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };

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

// 장소 검색 객체를 생성합니다
var ps = new kakao.maps.services.Places();

// 검색 결과 목록이나 마커를 클릭했을 때 장소명을 표출할 인포윈도우를 생성합니다
var infowindow = new kakao.maps.InfoWindow({zIndex:1});

// 키워드로 장소를 검색합니다
searchPlaces();

// 키워드 검색을 요청하는 함수입니다
function searchPlaces() {

    var keyword = document.getElementById('keyword').value;

    if (!keyword.replace(/^\s+|\s+$/g, '')) {
        alert('키워드를 입력해주세요!');
        return false;
    }

    // 장소검색 객체를 통해 키워드로 장소검색을 요청합니다
    ps.keywordSearch( keyword, placesSearchCB);
}

// 장소검색이 완료됐을 때 호출되는 콜백함수 입니다
function placesSearchCB(data, status, pagination) {
    if (status === kakao.maps.services.Status.OK) {

        // 정상적으로 검색이 완료됐으면
        // 검색 목록과 마커를 표출합니다
        displayPlaces(data);

        // 페이지 번호를 표출합니다
        displayPagination(pagination);

    } else if (status === kakao.maps.services.Status.ZERO_RESULT) {

        alert('검색 결과가 존재하지 않습니다.');
        return;

    } else if (status === kakao.maps.services.Status.ERROR) {

        alert('검색 결과 중 오류가 발생했습니다.');
        return;

    }
}

// 검색 결과 목록과 마커를 표출하는 함수입니다
function displayPlaces(places) {

    var listEl = document.getElementById('placesList'),
        menuEl = document.getElementById('menu_wrap'),
        fragment = document.createDocumentFragment(),
        bounds = new kakao.maps.LatLngBounds(),
        listStr = '';

    // 검색 결과 목록에 추가된 항목들을 제거합니다
    removeAllChildNods(listEl);

    // 지도에 표시되고 있는 마커를 제거합니다
    removeMarker();

    for ( var i=0; i<places.length; i++ ) {

        // 마커를 생성하고 지도에 표시합니다
        var placePosition = new kakao.maps.LatLng(places[i].y, places[i].x),
            marker = addMarker(placePosition, i),
            itemEl = getListItem(i, places[i]); // 검색 결과 항목 Element를 생성합니다

        // 검색된 장소 위치를 기준으로 지도 범위를 재설정하기위해
        // LatLngBounds 객체에 좌표를 추가합니다
        bounds.extend(placePosition);

        // 마커와 검색결과 항목에 mouseover 했을때
        // 해당 장소에 인포윈도우에 장소명을 표시합니다

```

```

// mouseout 했을 때는 인포윈도우를 닫습니다
(function(marker, title) {
    kakao.maps.event.addListener(marker, 'mouseover', function() {
        displayInfowindow(marker, title);
    });

    kakao.maps.event.addListener(marker, 'mouseout', function() {
        infowindow.close();
    });

    itemEl.onmouseover = function () {
        displayInfowindow(marker, title);
    };

    itemEl.onmouseout = function () {
        infowindow.close();
    };
})(marker, places[i].place_name);

fragment.appendChild(itemEl);
}

// 검색결과 항목들을 검색결과 목록 Element에 추가합니다
listEl.appendChild(fragment);
menuEl.scrollTop = 0;

// 검색된 장소 위치를 기준으로 지도 범위를 재설정합니다
map.setBounds(bounds);
}

// 검색결과 항목을 Element로 반환하는 함수입니다
function getListItem(index, places) {
    var el = document.createElement('li'),
        itemStr = '<span class="markerbg marker_" + (index+1) + "></span>' +
            '<div class="info">' +
            '    <h5>' + places.place_name + '</h5>';

    if (places.road_address_name) {
        itemStr += '    <span>' + places.road_address_name + '</span>' +
            '    <span class="jibun gray">' + places.address_name + '</span>';
    } else {
        itemStr += '    <span>' + places.address_name + '</span>';
    }

    itemStr += '    <span class="tel">' + places.phone + '</span>' +
        '</div>';

    el.innerHTML = itemStr;
    el.className = 'item';

    return el;
}

// 마커를 생성하고 지도 위에 마커를 표시하는 함수입니다
function addMarker(position, idx, title) {
    var imageSrc = 'https://t1.daumcdn.net/localimg/localimages/07/mapapidoc/marker_number_blue.png', // 마커 이미지 url, 스프라이트 이
        imageSize = new kakao.maps.Size(36, 37), // 마커 이미지의 크기
        imgOptions = {
            spriteSize : new kakao.maps.Size(36, 691), // 스프라이트 이미지의 크기
            spriteOrigin : new kakao.maps.Point(0, (idx*46)+10), // 스프라이트 이미지 중 사용할 영역의 좌상단 좌표
            offset: new kakao.maps.Point(13, 37) // 마커 좌표에 일치시킬 이미지 내에서의 좌표
        },
        markerImage = new kakao.maps.MarkerImage(imageSrc, imageSize, imgOptions),
        marker = new kakao.maps.Marker({
            position: position, // 마커의 위치
            image: markerImage
        });

    marker.setMap(map); // 지도 위에 마커를 표출합니다
    markers.push(marker); // 배열에 생성된 마커를 추가합니다

    return marker;
}

// 지도 위에 표시되고 있는 마커를 모두 제거합니다
function removeMarker() {
    for ( var i = 0; i < markers.length; i++ ) {
        markers[i].setMap(null);
    }
    markers = [];
}

// 검색결과 목록 하단에 페이지번호를 표시는 함수입니다
function displayPagination(pagination) {
    var paginationEl = document.getElementById('pagination'),
        fragment = document.createDocumentFragment(),

```



```

        i;

        // 기존에 추가된 페이지번호를 삭제합니다
        while (paginationEl.hasChildNodes()) {
            paginationEl.removeChild (paginationEl.lastChild);
        }

        for (i=1; i<=pagination.last; i++) {
            var el = document.createElement('a');
            el.href = "#";
            el.innerHTML = i;

            if (i===pagination.current) {
                el.className = 'on';
            } else {
                el.onclick = (function(i) {
                    return function() {
                        pagination.gotoPage(i);
                    }
                })(i);
            }

            fragment.appendChild(el);
        }
        paginationEl.appendChild(fragment);
    }

    // 검색결과 목록 또는 마커를 클릭했을 때 호출되는 함수입니다
    // 인포윈도우에 장소명을 표시합니다
    function displayInfowindow(marker, title) {
        var content = '<div style="padding:5px;z-index:1;">' + title + '</div>';

        infowindow.setContent(content);
        infowindow.open(map, marker);
    }

    // 검색결과 목록의 자식 Element를 제거하는 함수입니다
    function removeAllChildNodes(el) {
        while (el.hasChildNodes()) {
            el.removeChild (el.lastChild);
        }
    }
}
</script>
</body>
</html>

```

▼ 주소로 장소 표시하기

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>주소로 장소 표시하기</title>
</head>
<body>
<div id="map" style="width:100%;height:900px;"></div>

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2&libraries=services"></script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(37.5494147, 127.0818126), // 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('서울특별시 마포구 월드컵북로 21', 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;">우리 학원</div>'
});
infowindow.open(map, marker);

// 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
map.setCenter(coords);
}
});
</script>
</body>
</html>

```

▼ 좌표로 주소를 얻어내기

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>좌표로 주소를 얻어내기</title>
  <style>
    .map_wrap {position:relative;width:100%;height:900px;}
    .title {font-weight:bold;display:block;}
    .hAddr {position:absolute;left:10px;top:10px;border-radius: 2px;background:#fff;background:rgba(255,255,255,0.8);z-index:1;padding:5px;}
    #centerAddr {display:block;margin-top:2px;font-weight: normal;}
    .bAddr {padding:5px;text-overflow: ellipsis;overflow: hidden;white-space: nowrap;}
  </style>
</head>
<body>
  <div class="map_wrap">
    <div id="map" style="width:100%;height:100%;position:relative;overflow:hidden;"></div>
    <div class="hAddr">
      <span class="title">지도중심기준 행정동 주소정보</span>
      <span id="centerAddr"></span>
    </div>
  </div>

  <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2&libraries=services"></script>
  var mapContainer = document.getElementById('map'), // 지도를 표시할 div
      mapOption = {
        center: new kakao.maps.LatLng(37.566826, 126.9786567), // 지도의 중심좌표
        level: 1 // 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();

  var marker = new kakao.maps.Marker(), // 클릭한 위치를 표시할 마커입니다
      infowindow = new kakao.maps.InfoWindow({zindex:1}); // 클릭한 위치에 대한 주소를 표시할 인포윈도우입니다

  // 현재 지도 중심좌표로 주소를 검색해서 지도 좌측 상단에 표시합니다
  searchAddrFromCoords(map.getCenter(), displayCenterInfo);

  // 지도를 클릭했을 때 클릭 위치 좌표에 대한 주소정보를 표시하도록 이벤트를 등록합니다
  kakao.maps.event.addListener(map, 'click', function(mouseEvent) {
    searchDetailAddrFromCoords(mouseEvent.latLng, function(result, status) {
      if (status === kakao.maps.services.Status.OK) {
        var detailAddr = !!result[0].road_address ? '<div>도로명주소 : ' + result[0].road_address.address_name + '</div>' : '';
        detailAddr += '<div>지번 주소 : ' + result[0].address.address_name + '</div>';
        var content = '<div class="bAddr">' +
          '<span class="title">법정동 주소정보</span>' +
          detailAddr +
          '</div>';

        // 마커를 클릭한 위치에 표시합니다
        marker.setPosition(mouseEvent.latLng);
        marker.setMap(map);

        // 인포윈도우에 클릭한 위치에 대한 법정동 상세 주소정보를 표시합니다
        infowindow.setContent(content);
        infowindow.open(map, marker);
      }
    });
  });

  // 중심 좌표나 확대 수준이 변경됐을 때 지도 중심 좌표에 대한 주소 정보를 표시하도록 이벤트를 등록합니다
  kakao.maps.event.addListener(map, 'idle', function() {
    searchAddrFromCoords(map.getCenter(), displayCenterInfo);
  });

```

```

});

function searchAddrFromCoords(coords, callback) {
    // 좌표로 행정동 주소 정보를 요청합니다
    geocoder.coord2RegionCode(coords.getLng(), coords.getLat(), callback);
}

function searchDetailAddrFromCoords(coords, callback) {
    // 좌표로 법정동 상세 주소 정보를 요청합니다
    geocoder.coord2Address(coords.getLng(), coords.getLat(), callback);
}

// 지도 좌측상단에 지도 중심좌표에 대한 주소정보를 표출하는 함수입니다
function displayCenterInfo(result, status) {
    if (status === kakao.maps.services.Status.OK) {
        var infoDiv = document.getElementById('centerAddr');

        for(var i = 0; i < result.length; i++) {
            // 행정동의 region_type 값은 'H' 이므로
            if (result[i].region_type === 'H') {
                infoDiv.innerHTML = result[i].address_name;
                break;
            }
        }
    }
}
}
</script>
</body>
</html>

```

사용 연습

▼ 주소 검색으로 지도와 로드뷰 띄우기

▼ addressForm.jsp

```

<!-- addressForm.jsp -->

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>주소 검색</title>
</head>
<body>
    <form action="addr.jsp" method="get">
        주소 :
        <input type="text" name="addr">
        <button>검색</button>
    </form>
</body>
</html>

```

▼ addrMap.jsp

```

<!-- addrMap.jsp -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String addr = request.getParameter("addr");
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>주소로 장소 표시하기</title>
</head>
<body>
    <div id="map" style="width:100%;height:450px;"></div>
    <div id="roadview" style="width:100%;height:450px;"></div>
    <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=2af21256d2413713f5827fb37b8f1cb2&libraries=service:
    </script>
    var mapContainer = document.getElementById('map'), // 지도를 표시할 div
        mapOption = {
            center: new kakao.maps.LatLng(37.5494147, 127.0818126), // 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) {

            // 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;"><%=addr%></div>'
                });
                infowindow.open(map, marker);

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

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

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

                // 특정 위치의 좌표와 가까운 로드뷰의 panoId를 추출하여 로드뷰를 띄운다.
                roadviewController.getNearestPanoId(position, 50, function(panoId) {
                    roadviewController.setPanoId(panoId, position); //panoId와 중심좌표를 통해 로드뷰 실행
                });
            }
        }
    });
}
</script>
</body>
</html>

```