

20230119 Kakao Map

| | |
|----|---------------|
| 날짜 | @2023년 1월 19일 |
| 태그 | 지도 카카오 |

카카오 JavaScript 키 : e85f1f60d5ae784c7e84ade4beae9fd5

카카오 지도

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>) 접속 2. 개발자 등록 및 앱 생성 3. 웹 플랫폼 추가: 앱 선택 - [플랫폼] - [Web 플랫폼 등록] - 사이트 도메인 등록 4. 사이트 도메인 등록: [웹] 플랫폼

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

Maps API

Dalme 지도 API는 웹사이트와 모바일 애플리케이션에서 지도를 이용한 서비스를 제작할 수 있도록 다양한 기능을 제공하고 있습니다.



지도 API 호출

```
<div id="map" style="width:700px;height:700px;"></div>
<script type="text/javascript"
  src="//dapi.kakao.com/v2/maps/sdk.js?appkey=e85f1f60d5ae784c7e84ade4beae9fd5">
</script>
```

초기화면 설정

```
<script>
  var container = document.getElementById('map');
  var options = {
    center: new kakao.maps.LatLng(33.450701, 126.570667), // 위도, 경도 설정
    level: 3 // 축적 설정
  };
  var map = new kakao.maps.Map(container, options);
</script>
```

지도 정보 추출

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

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

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

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

지도에 컨트롤 올리기

```
<script>
  // 지도의 형태를 바꾸는 컨트롤러 추가
  var mapTypeControl = new kakao.maps.MapTypeControl();
  map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);

  // 지도의 크기를 확대/축소하는 컨트롤러 추가
  var zoomControl = new kakao.maps.ZoomControl();
  map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);
</script>
```

지도 이동, 확대/축소 막기

```

<script>
// 지도 이동 막기
map.setDraggable(false);

// 지도 확대 축소 막기
map.setZoomable(false);
</script>

```

지도 범위 재설정

```

<script>
// 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();

// 세 개의 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);

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

function setBounds() {
    map.setBounds(bounds);
}

</script>

```

중심좌표 변경 이벤트 등록

```

<script>
// 지도 변경시 중심 좌표
kakao.maps.event.addListener(map, 'center_changed', function() {
// 지도의 레벨 추출
var level = map.getLevel();

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

var message = '<p> 지도의 레벨은 ' + level + ' 이고</p>';
message += '<p> 중심좌표는 (' + latlng.getLat() + ', ' + latlng.getLng() + ')</p>';

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

});
</script>

```

타일로드 이벤트 등록

```

<script>
// 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);
}
</script>

```

마커

- Basic

```
<script>
// 마커를 표시할 위치
var markerPosition = new kakao.maps.LatLng(37.543688, 126.899759);

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

// 마커 지도에 올리기
marker.setMap(map);

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

- 사진설정

```
<script>
// 마커로 사용할 사진 설정
var imgSrc = "https://search.pstatic.net/common/?src=http%3A%2F%2Fblogfiles.naver.net%2FMjAyMjBfNDkg%2FMDAxNjQyNjY5NjMzMjgy.xBJD
imgSize = new kakao.maps.Size(64,69),
imgOption = {offset : new kakao.maps.Point(27,69)};

var markerImg = new kakao.maps.MarkerImage(imgSrc, imgSize, imgOption),
markerPosition = new kakao.maps.LatLng(37.543688, 126.899759);

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

// 마커 지도에 올리기
marker.setMap(map);
</script>
```

- 마우스 이벤트

```
<script>
// 마커 생성
// 마커를 표시할 위치
var markerPosition = new kakao.maps.LatLng(37.543688, 126.899759);

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

// 마커 지도에 올리기
marker.setMap(map);

// 마커에 마우스를 올렸을 때 정보창을 표시
// 정보창에 나타낼 정보 작성
var infoContent = '<div style = "padding:5px;">hi hello 안녕 </div>';

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

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

// 마커에서 마우스가 떨어졌을 때 이벤트
kakao.maps.event.addListener(marker, 'mouseout', function(){
    infowindow.close();
});
</script>
```

- 마커 여러개 표시

```

<script>
var container = document.getElementById('map');
var options = {
  center: new kakao.maps.LatLng(37.558043, 126.919002),
  level: 15 // 지도의 크기를 설정 예) 10 이 서울 전체, 제주 전체
};

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

var positions = [
  {
    title: '카카오',
    latlng: new kakao.maps.LatLng(37.558043, 126.919002)
  },
  {
    title: '우리집',
    latlng: new kakao.maps.LatLng(33.450701, 126.570667)
  }
];

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

```

- 마우스 클릭시 마커 생성

```

<script>
//지도를 마우스로 클릭했을때 이벤트 발생 처리

kakao.maps.event.addListener(map, 'click', function(mouseEvent) {
  // 클릭한 위치에 마커를 표시합니다
  addMarker(mouseEvent.latLng);
});

// 지도에 표시된 마커 객체를 가지고 있을 배열입니다
var markers = [];

// 마커 하나를 지도위에 표시합니다
addMarker(new kakao.maps.LatLng(33.450701, 126.570667));

// 마커를 생성하고 지도위에 표시하는 함수입니다
function addMarker(position) {

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

  // 마커가 지도 위에 표시되도록 설정합니다
  marker.setMap(map);

  // 생성된 마커를 배열에 추가합니다
  markers.push(marker);
}
</script>

```

정보창

- Basic

```

<script>
// 정보창에 나타낼 정보 작성
var infoContent = '<div style = "padding:5px;">hi hello 안녕 </div>',
infoPosition = new kakao.maps.LatLng(37.543688, 126.899759);

// 창 생성
var infoWindow = new kakao.maps.InfoWindow({
  map : map,
  position : infoPosition,
  content : infoContent,
  removable : true
});

</script>

```

- 마커와 정보창 동시에 생성

```
<script>

// 마커 올리기
// 마커 위치 설정
var markerPosition = new kakao.maps.LatLng(37.543688, 126.899759);
// 마커 객체 생성
var marker = new kakao.maps.Marker({
    position : markerPosition
});
// 마커 지도에 올리기
marker.setMap(map);

// 정보창 올리기
// 정보창에 나타낼 정보 작성
var infoContent = '<div style = "padding:5px;">hi hello 안녕 </div>',
infoPosition = new kakao.maps.LatLng(37.543688, 126.899759);

// 정보창 객체 생성
var infoWindow = new kakao.maps.InfoWindow({
    position : infoPosition,
    content : infoContent,
});

// 정보창 지도에 올리기
infoWindow.open(map,marker);    // marker와 겹치지 않게 표시
</script>
```

현재 위치 불러오기

```
<script>
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;">여기에 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(33.450701, 126.570667),
        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);
}
```

```

    }
  </script>

```

▼ 직선 상의 거리 계산

```

<!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-s:
    .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:
    .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>지도로 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=e85f1f60d5ae784c7e84ade4beae9fd5"></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 커스텀오버레이입니다
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
    deleteDistnce();

    // 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 종료했을 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 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 bycycleTime = distance / 227 | 0;
    var bycycleHour = '', bycycleMin = '';

    // 계산한 자전거 시간이 60분 보다 크면 시간으로 표시합니다
    if (bycycleTime > 60) {
        bycycleHour = '<span class="number">' + Math.floor(bycycleTime / 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

```

<script>
// 로드 뷰는 지도와는 별도로 그릴 수 있다. 지도없이 로드뷰만 생성 가능

// 로드 뷰 표시 위치
var roadViewContainer = document.getElementById('road');

// 로드 뷰 객체 생성
var roadview = new kakao.maps.Roadview(roadViewContainer);

// 해당 좌표로부터 로드 뷰 파노라마 ID를 가져올 로드 뷰 helper 객체
var roadViewClient = new kakao.maps.RoadviewClient();

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

// 특정 위치의 좌표와 가까운 로드 뷰를 추출하여 로드 뷰를 띄운다
roadViewClient.getNearestPanoId(position, 50, function(panoId){
    roadview.setPanoId(panoId, position)
})

</script>

```

라이브러리

라이브러리를 사용할 때 api에 구문을 추가해야 한다

```

<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?
appkey=1d944d4a9cb22483964d7e130a9b4c2a&libraries=services"></script>

```

- 키워드로 장소 검색

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>키워드로 장소검색하기</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=1d944d4a9cb22483964d7e130a9b4c2a&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(37.566826, 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();

  // 키워드로 장소를 검색합니다
  ps.keywordSearch('이태원 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
      infowindow.setContent('<div style="padding:5px;font-size:12px;">' + place.place_name + '</div>');
      infowindow.open(map, marker);
    });
  }
</script>
</body>
</html>
```

- 주소로 장소 검색

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>주소 장소검색하기</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=1d944d4a9cb22483964d7e130a9b4c2a&libraries=services"></script>
<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(37.566826, 126.9786567), // 지도의 중심좌표
    level : 3
  }, // 지도의 확대 레벨

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

  // 주소를 좌표로 변환 객체 생성
  var geoCoder = new kakao.maps.services.Geocoder();

  geoCoder.addressSearch('서울특별시 마포구 월드컵북로 21',function(result, status){
    // 정상적으로 해당 주소를 검색 완료했다면
    if(status === kakao.maps.services.Status.OK){
      var coords = new kakao.maps.LatLng(result[0].y, result[0].x);
      var marker = new kakao.maps.Marker({
        map : map,
        position : coords
      });
      var infowindow = new kakao.maps.InfoWindow({
        content : "검색 지점"
      });
      infowindow.open(map, marker);

      // 해당 주소로 지도 화면 중심 이동
      map.setCenter(coords);
    }
  })
</script>
</body>
</html>

```