

~~REVEAL night~~

→

|

>

GIS Javascript library

- Openlayers
- Leaflet

Quick Start

<https://wiki.wrnd.ml/open/openlayers>

→

|

>

###

- vscode

- Prettier - Code formatter
- Open HTML in Default Browser

→

|

>

OpenLayers

: <https://openlayers.org/>

→

|

>

OpenLayers

-

- Google Maps, Bing Maps, OSM
- OGC

가

.

.



<https://www.slideshare.net/jinifor/5open-layers>

→

|

>

HTML

HyperText Markup Language

- Hyper () , , ,
- Markup Language -

```
<tag>          </tag>
```

→

|

>

HTML

```
<!DOCTYPE html>
<html>
  <head>
```

```
<meta charset="utf-8">
<title>My test page</title>
</head>
<body>
  
</body>
</html>
```

https://developer.mozilla.org/ko/docs/Learn/Getting_started_with_the_web

→

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

https://developer.mozilla.org/ko/docs/Learn/Getting_started_with_the_web

<notes>

- <!DOCTYPE html> — doctype. HTML (1991 2), doctype () good HTML HTML 가
- <html></html> — <html> (root)
- <head></head> — <head> HTML (keywords , CSS,
- <body></body> — <body>
- <meta charset="utf-8"> — 가 utf-8 가 (utf-8). 가 ,
- <title></title> — <title> 가

</notes>

—→

Javascript

!

-
- ,



https://madnight.github.io/githut/#/pull_requests/2018/1

—→

|

>

Quick Start

<http://openlayers.org/en/latest/doc/quickstart.html>



—→

Put a map on a page

```
<!DOCTYPE html>
<html>
  <head>
    <title>OpenLayers example</title>
    <link rel="stylesheet"
href="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v5.3.0/css/ol.css" type="text/css">
    <script
src="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v5.3.0/build/ol.js"></script>
    <style>
      .map {
        height: 400px;
        width: 100%;
      }
    </style>
  </head>
  <body>
    <h2>My Map</h2>
    <div id="map" class="map"></div>
    <script type="text/javascript">
      var lat = 37.482923;
      var lon = 126.896016;

      var map = new ol.Map({
        target: 'map',
        layers: [
          new ol.layer.Tile({
            source: new ol.source.OSM()
          })
        ],
        view: new ol.View({
          center: ol.proj.fromLonLat([lon , lat]),
          zoom: 4
        })
      });
    </script>
  </body>
</html>
```

→



→

Vworld

```
<script type="text/javascript">
  var lat = 37.482923;
  var lon = 126.896016;
  var zoom = 17;

  var map = new ol.Map({
    target: 'map',
    layers: [
      new ol.layer.Tile({
        source: new ol.source.XYZ({
          url:
'http://xdworld.vworld.kr:8080/2d/Base/service/{z}/{x}/{y}.png'
        })
      })
    ],
    view: new ol.View({
      center: ol.proj.fromLonLat([lon, lat]),
      zoom: zoom
    })
  });
</script>
```

→



→

|

>

##

→

```
var lat = 37.482923;
var lon = 126.896016;
var zoom = 17;

var vworldLayer = new ol.layer.Tile({
  source: new ol.source.XYZ({
    url: 'http://xdworld.vworld.kr:8080/2d/Base/service/{z}/{x}/{y}.png'
  })
});

const markerSource = new ol.source.Vector();

var markerStyle = new ol.style.Style({
  image: new ol.style.Icon(({
    scale: 0.1,
    src: 'image/marker.png'
  }))
});

var markerLayer = new ol.layer.Vector({
  source: markerSource,
  style: markerStyle,
});
```

→

가

```
var map = new ol.Map({
  target: 'map',
  layers: [
    vworldLayer, markerLayer
  ],
  view: new ol.View({
    center: ol.proj.fromLonLat([lon, lat]),
    zoom: zoom
  })
});

var iconFeature = new ol.Feature({
  geometry: new ol.geom.Point(new ol.proj.fromLonLat([lon, lat]))
});

markerSource.addFeature(iconFeature);
```

→



→

|

>

Cluster

Clustered Features



—→

feature

```

var lat = 37.482923
var lon = 126.896016
var zoom = 17

var vworldLayer = new ol.layer.Tile({
  source: new ol.source.XYZ({
    url: "http://xdworld.vworld.kr:8080/2d/Base/service/{z}/{x}/{y}.png",
  }),
})

var count = 100
var features = new Array(count)

for (var i = 0; i < count; ++i) {
  var coordinates = [lon + Math.random() / 100, lat + Math.random() / 100]
  features[i] = new ol.Feature({
    geometry: new ol.geom.Point(ol.proj.fromLonLat(coordinates)),
  })
}

```

—→

```

var source = new ol.source.Vector({
  features: features,
})

var clusterSource = new ol.source.Cluster({
  distance: 20,
  source: source,
})

```

```
})
```

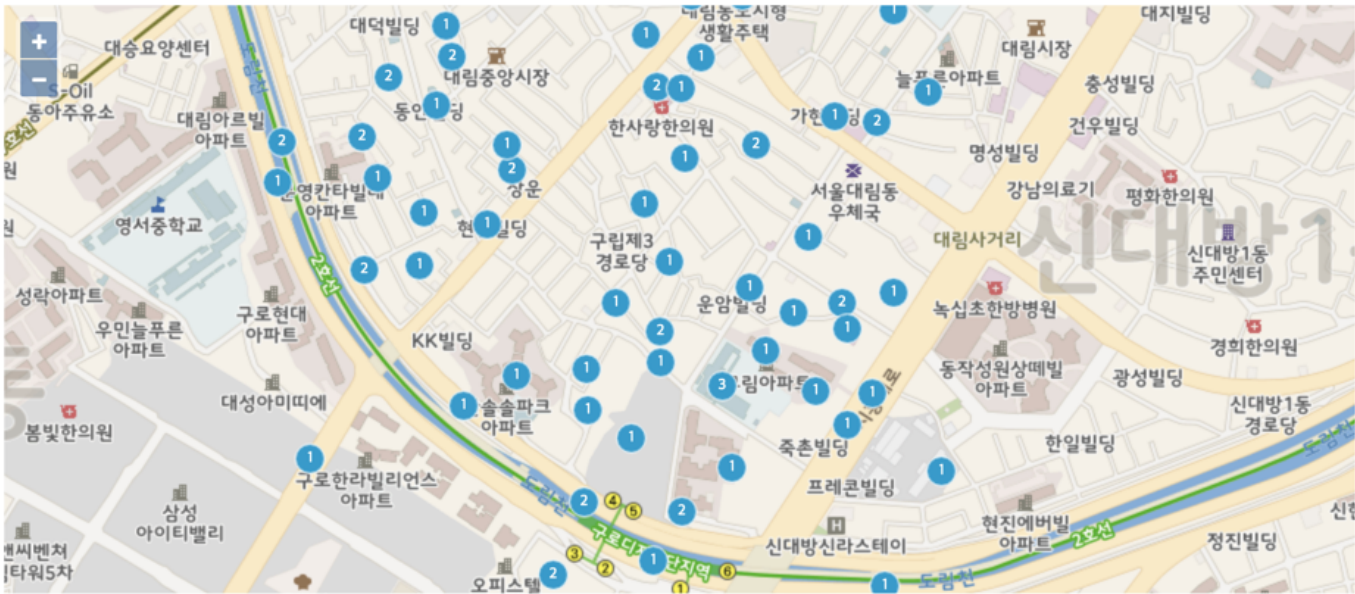
→

```
var styleCache = {}
var clusters = new ol.layer.Vector({
  source: clusterSource,
  style: function(feature) {
    var size = feature.get("features").length
    var style = styleCache[size]
    if (!style) {
      style = new ol.style.Style({
        image: new ol.style.Circle({
          radius: 10,
          stroke: new ol.style.Stroke({
            color: "#fff",
          }),
          fill: new ol.style.Fill({
            color: "#3399CC",
          }),
        }),
        text: new ol.style.Text({
          text: size.toString(),
          fill: new ol.style.Fill({
            color: "#fff",
          }),
        }),
      })
      styleCache[size] = style
    }
    return style
  },
})

var map = new ol.Map({
  layers: [vworldLayer, clusters],
  target: "map",
  view: new ol.View({
    center: ol.proj.fromLonLat([lon, lat]),
    zoom: zoom,
  }),
})
```

→

My Map



→
|
>

. —
|
>
—

Plugin Backlinks:

From:
<https://moro.kr/> - Various Ways

Permanent link:
<https://moro.kr/open/openlayers>

Last update: 2020/06/02 09:25