예제 코드 분석

2021년 11월 30일

2021년 2학기 소프트웨어시스템실습

담당교수: 최성종

윤성연

allsmile99@office.uos.ac.kr

서울시립대학교 전자전기컴퓨터공학부

차 례

[1. GeoJSON데이터 레이어 표시하기 1](#_Toc89186565)

[1.1. 예제코드 1](#_Toc89186566)

[1.2. 코드에서 사용한 클래스 2](#_Toc89186567)

[1.3. 실습 코드 분석 10](#_Toc89186568)

# GeoJSON데이터 레이어 표시하기

## 예제코드



## 코드에서 사용한 클래스

### 클래스의 주된 목적

클래스(class)란 객체를 정의하는 틀 또는 설계도와 같은 의미로 사용된다. 이러한 설계도인 클래스를 가지고, 여러 객체를 생성하여 사용한다. 클래스는 객체의 상태를 나타내는 필드(field)와 객체의 행동을 나타내는 메소드(method)로 구성된다. 즉, 필드(field)란 클래스에 포함된 변수(variable)를 의미한다. 또한, 메소드(method)란 어떠한 특정 작업을 수행하기 위한 명령문의 집합이다.

이러한 클래스를 통해 설계도를 활용하여 주된 기능을 손쉽게 활용할 수 있다.

### 코드에서 사용한 클래스, 프로퍼티, 메소드 기능분석

**var** map = **new** naver.maps.Map(document.getElementById('map'), {

zoom: 18,

mapTypeId: 'normal',

center: **new** naver.maps.LatLng(37.3586524, 127.1060678)

});

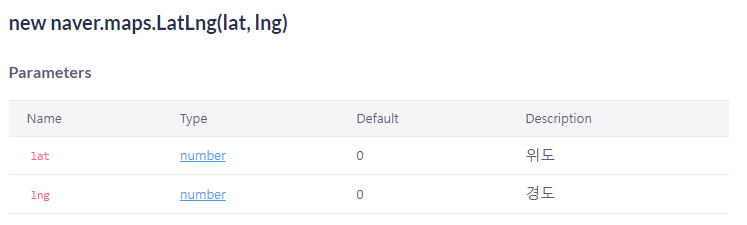
텍스트이(가) 표시된 사진

자동 생성된 설명



텍스트이(가) 표시된 사진

자동 생성된 설명



naver.maps.Event.once(map, 'init\_stylemap', **function** () {

$.ajax({

url: HOME\_PATH +'/data/naver.json',

dataType: 'json',

success: startDataLayer

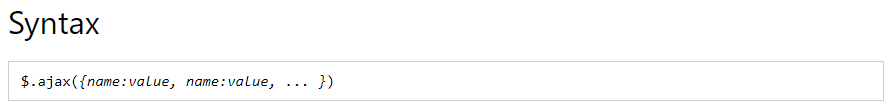
});

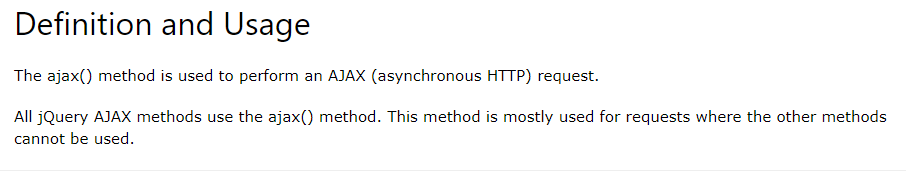
텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명











**function** **startDataLayer**(geojson) {

map.data.addGeoJson(geojson);

map.data.setStyle(**function**(feature) {

**var** color = 'red';

**if** (feature.getProperty('isColorful')) {

color = feature.getProperty('color');

}

**return** {

fillColor: color,

strokeColor: color,

strokeWeight: 2,

icon: null

};

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명



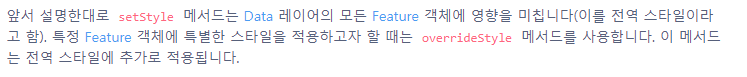
텍스트이(가) 표시된 사진

자동 생성된 설명

map.data.addListener('click', **function**(e) {

e.feature.setProperty('isColorful', true);

});



map.data.addListener('dblclick', **function**(e) {

**var** bounds = e.feature.getBounds();

**if** (bounds) {

map.panToBounds(bounds);

}

});

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

map.data.addListener('mouseover', **function**(e) {

map.data.overrideStyle(e.feature, {

strokeWeight: 8,

icon: HOME\_PATH +'/img/example/pin\_spot.png'

});

});

map.data.addListener('mouseout', **function**(e) {

map.data.revertStyle();

});

}



텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

## 실습 코드 분석

**var** map = **new** naver.maps.Map(document.getElementById('map'), {

zoom: 18,

mapTypeId: 'normal',

center: **new** naver.maps.LatLng(37.3586524, 127.1060678)

});

//map을 참조변수로 가지는 Map의 객체 생성

//불러온 지도를 객체로 받아들이고, 이에 대한 프로퍼티와 메소드를 정의

//zoom프로터피는 18, mapTypeId프로퍼티는 normal, center프로퍼티는 LatLng객체의 (37.3586524, 127.1060678)에 해당하는 객체

**var** HOME\_PATH = window.HOME\_PATH || '.';

naver.maps.Event.once(map, 'init\_stylemap', **function** () {

$.ajax({

url: HOME\_PATH +'/data/naver.json',

dataType: 'json',

success: startDataLayer

});

});

//주소의 경로를 받는 변수지정

//Event객체의 once메소드를 호출

//ajax로 받아온다

**function** **startDataLayer**(geojson) {

map.data.addGeoJson(geojson);

map.data.setStyle(**function**(feature) {

**var** color = 'red';

**if** (feature.getProperty('isColorful')) {

color = feature.getProperty('color');

}

**return** {

fillColor: color,

strokeColor: color,

strokeWeight: 2,

icon: null

};

});

//완료되면 함수실행

//기본설정지정

map.data.addListener('click', **function**(e) {

e.feature.setProperty('isColorful', true);

});

//click이벤트 발생시, 이벤트 함수 실행

//이벤트함수의 feature의 setProperty메소드에 의해서 색 변경

map.data.addListener('dblclick', **function**(e) {

**var** bounds = e.feature.getBounds();

**if** (bounds) {

map.panToBounds(bounds);

}

});

//더블클릭 이벤트 발생시, 이벤트 함수 실행

//이벤트 함수의 feature의 getBounds메소드에 의해서 형상 정보의 좌표 경계를 받아옴

//글자 좌표 경계를 포함하는 위치로 지도를 부드럽게 이동(확대됨)

map.data.addListener('mouseover', **function**(e) {

map.data.overrideStyle(e.feature, {

strokeWeight: 8,

icon: HOME\_PATH +'/img/example/pin\_spot.png'

});

});

map.data.addListener('mouseout', **function**(e) {

map.data.revertStyle();

});

}

//마우스오버 이벤트 발생시, 이벤트 함수 실행

//overrideStyle메소드로 지정된 곳의 아이콘 변경

//마우스아웃 이벤트 발생시, 이벤트 함수 실행

//revertStyle 메소드로 지정된 곳의 변경사항 원상복귀