Final Project

2021년 12월 14일

2021년 2학기 소프트웨어시스템실습

담당교수: 최성종

서정민

profeimd7@office.uos.ac.kr

서울시립대학교 전자전기컴퓨터공학부

차 례

[1. 요약 1](#_Toc90382246)

[2. 요구사항 도출 2](#_Toc90382247)

[2.1. 프로젝트 스케치 2](#_Toc90382248)

[2.2. 기능적 요소 문제 제시 2](#_Toc90382249)

[2.3. 비기능적 요소 문제 제시 2](#_Toc90382250)

[3. 설계 3](#_Toc90382251)

[3.1. 코로나19 감염 현황 3](#_Toc90382252)

[3.2. NAVER MAP 3](#_Toc90382253)

[3.3. 긴급재난문자 4](#_Toc90382254)

[4. 구현 5](#_Toc90382255)

[4.1. 코로나19 감염 현황 5](#_Toc90382256)

[4.2. NAVER MAP 6](#_Toc90382257)

[4.3. 긴급재난문자 6](#_Toc90382258)

[5. 토론 8](#_Toc90382259)

[6. 결론 10](#_Toc90382260)

[7. 부록 12](#_Toc90382261)

[7.1. XML-Ajax 12](#_Toc90382262)

[7.2. JSON 12](#_Toc90382263)

[7.3. 소스코드(.html) 13](#_Toc90382264)

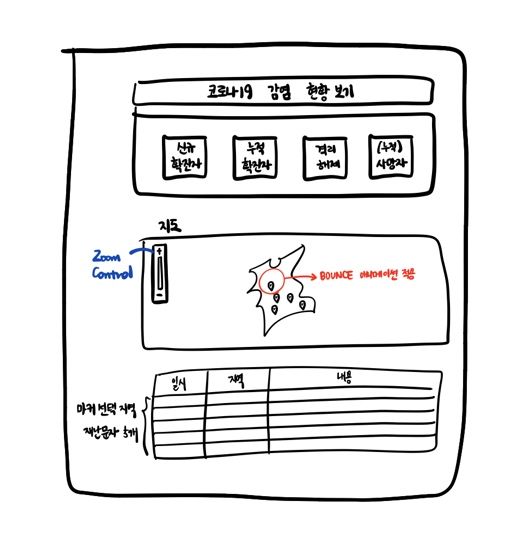
[8. 참고자료 30](#_Toc90382265)

# 요약

이번 기말과제에서는 ‘코로나19 감염 현황’을 요약하여 보여주고, 지도에서 원하는 지역을 선택하여 해당 지역의 ‘긴급재난문자’를 확인할 수 있는 웹 페이지를 제작한다. 이에 필요한 데이터는 공공데이터포털의 Open API를 활용하며, 지도는 NAVER MAP API를 활용한다. 이 과정에서 XML, Ajax, jQuery, JSON 등 기술에 대해 학습하고 적용해본다. 뿐만 아니라, 이전에 학습한 HTML DOM, CSS, JavaScript 등 다양한 기술을 적용하며 기말과제를 진행한다.

# 요구사항 도출

## 프로젝트 스케치

 텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명

## 기능적 요소 문제 제시

* 코로나19 감염 현황 정보를 ‘버튼’ 클릭 시 테이블 형태로 제공한다.
* 모든 지방자치단체(17개의 시/도)를 지도에서 ‘마커(Marker)’를 이용하여 선택할 수 있도록 한다.
* 지도 왼쪽 상단에 지도 확대/축소 조정이 가능한 설정을 배치한다.
* 지도에서 마커 클릭 시 Bounce 애니메이션을 적용한다.
* 지도에서 마커 클릭 시 하단에 테이블 형태로 해당 지역의 긴급재난문자를 출력한다.

## 비기능적 요소 문제 제시

* ‘코로나19 감염 현황 보기’ 버튼을 디자인한다.
* 웹페이지가 깔끔하게 보이도록 버튼, 테이블의 사이즈를 통일한다.

# 설계

## 코로나19 감염 현황

* <style> 태그에서 버튼 및 테이블의 스타일 진행
* 버튼 Form 입력과 테이블 출력을 위한 태그 및 id 필요
* Ajax XMLHttpRequest Object기술을 통해 공공데이터포털의 ‘코로나19 시, 도별 발생 현황’ Open API를 활용
* 공공데이터포털에서 받아온 데이터를 활용하여 신규 확진자, 누적 확진자, 격리 해제자, 누적 사망자 수를 계산해내어 테이블로 출력

## NAVER MAP

* 서울 시청을 중심으로 출력되도록 지도 생성
* 지도 왼쪽 상단 zoomControl 가능하도록 설정
* 17개의 각 시/도 지방자치단체기관의 위도 및 경도를 모두 파악하여 지도 상에 마커로 표시
* 각 마커를 클릭하여 해당 지역의 긴급재난문자를 출력하는 처리를 하나의 함수로 구현하기 위해, 마커 생성 시 ‘title’을 이용하여 해당 지역의 시/도 명칭 저장
* 마커 클릭 시 해당 지역이 선택되었음을 알 수 있도록 애니메이션 적용

## 긴급재난문자

* 마커 클릭(입력) 시 해당 지역의 명칭을 매개변수로 받아 그 지역의 긴급재난문자를 테이블 형태로 출력
* Ajax XMLHttpRequest Object기술을 통해 공공데이터포털의 ‘재난문자방송 발령 현황’Open API를 활용
* 공공데이터포털에서 받아온 데이터로부터 문자 발송 일시, 지역, 내용을 추출하여 테이블로 출력

# 구현

## 코로나19 감염 현황

* 버튼, 테이블은 width 80%, 좌우 여백 10%로 설정을 통일하여 테이블 출력 시 웹페이지가 깔끔하게 보이도록 스타일 지정
* 버튼은 배경 색상, 테두리 색상, 글자 색상, 글자 크기, 버튼 여백, 테두리 둥글게 처리 등을 진행하여 깔끔하게 디자인

지도이(가) 표시된 사진

자동 생성된 설명

* 버튼 클릭 시 ‘buildTable1()’ 함수 실행되도록 설정
* ‘buildTable1()’함수는 XMLHttpRequest 객체를 이용하여 공공데이터포털의 Open API를 받아오고 myFunction()을 수행한다.
* myFunction()은 공공데이터포털에서 받은 xml로부터 각 지역의 신규 확진자, 누적 확진자, 격리 해제자, 누적 사망자의 수를 더해준다. 이때 17개 시/도 자치단체 뿐만 아니라 ‘검역’ 단계에서 나온 확진자까지 포함해야 한다. 따라서, 반복문은 17번이 아닌 18번 수행되도록 코드를 작성한다. 또한, xml로부터 받은 자료는 string형태로 덧셈을 하기 위해서는 parseInt()를 이용하여 type casting을 해줘야 한다. DOM과 테이블의 ID를 이용하여 계산 결과를 innerHTML 프로퍼티에 저정하여 테이블이 출력될 수 있도록 한다.

## NAVER MAP

* 서울 시청을 중심으로 하고, zoomControl이 가능하도록 Map 객체를 생성한다.
* 각 시/도 자치단체의 위도, 경도를 파악하여 17개의 마커를 생성한다. 이때, ‘title’에 해당 지역의 명칭을 같이 작성한다. 뒤에서 지역별 재난문자방송 발령현황을 받아올 때 이를 이용한다.
* 마커 클릭 시, 적용된 애니메이션이 없다면 Bounce 애니메이션을 설정하고, 이미 애니메이션이 적용되어 있다면 해당 애니메이션을 해제한다. 즉, 지도에서 특정 지역 선택 시 해당 지역이 선택되었음을 Bounce 애니메이션으로 알려줄 수 있으며, 한 번 더 클릭하여 애니메이션을 해제시킬 수 있다.

## 긴급재난문자

* 마커의 title 프로퍼티를 매개변수로 하여 buildTable2() 함수를 호출한다.
* buildTable2(location) 함수는 XMLHttpRequest 객체를 이용하여 공공데이터포털의 Open API를 받아온다. 이때, 매개변수로 받은 지역의 긴급재난문자만을 가져온다. 이후, myFunction()을 수행한다.
* myFunction()은 공공데이터포털에서 받은 xml로부터 문자 발송 일시, 지역, 내역을 추출한다. DOM과 테이블의 ID를 이용하여 계산 결과를 innerHTML 프로퍼티에 저정하여 테이블이 출력될 수 있도록 한다.

# 토론

먼저, 코로나19 감염 현황의 경우 공공데이터포털에서 제공하는 데이터 자체만으로는 전국의 하루 신규 확진자를 알 수 없었다. 감염 현황 자료에서는 해당 정보를 제공하지 않고, 시/도 발생 현황에서는 각 지역별 신규 확진자 정보만을 제공하였다. 따라서 각 지역의 신규 확진자를 모두 더해주는 데이터 가공의 과정을 거쳐 원하는 정보를 얻어내었다. 이때, 17개의 자치단체 뿐만 아니라 ‘검역’ 단계에서 확진되는 인원을 빠뜨리지 않도록 주의해야 했다.

NAVER MAP은 마커를 사용하여 지역을 설정할 수 있도록 하였다. 이전 수업에서 행정구역 GeoJSON 데이터레이어를 이용하여 지역 선택을 할 수 있는 실습을 진행해보았다. 하지만 이렇게 설정해둘 경우, 직접 마우스를 가져가 움직여봐야 선택할 수 있는 지역을 확인할 수 있는 불편함이 있다. 따라서 17개 시/도 자치단체의 위도 및 경도를 모두 파악하고 입력해야 하는 번거로운 과정이 있어도, 지도에서 선택 가능한 위치를 한 눈에 파악할 수 있도록 마커를 사용하여 기말과제를 진행하였다. 또한, 선택한 위치를 알 수 있도록 마커 클릭 시 Bounce 애니메이션을 적용하였다.

긴급재난문자는 앞서 구현한 마커를 클릭하였을 때 테이블로 출력될 수 있도록 하였다. 17개의 마커에 대하여 하나의 함수로 구현하기 위해서 각 마커 생성 시 title 프로퍼티에 해당 지역의 명칭을 저장해두고, 함수 호출 시 이 프로퍼티를 매개변수로 넘겨주었다.

요구사항에 대해서는 먼저, 공공데이터포털의 재난 관련 정보로 ‘코로나19 시/도 발생 현황’ 및 ‘지역별 재난문자방송 발령 현황’ Open API를 이용하였다.

HTML은 전체적인 문서의 틀을 작성하는 데 사용하였고, CSS는 본문의 style 뿐만 아니라 버튼 디자인, 테이블 사이즈 조정 등에 사용하였다. JavaScript는 Ajax, NAVER MAP API, 함수 구현 등 가장 많이 사용한 기술이다. 또한, Ajax를 사용하고 테이블 출력을 만들어내는 과정에서 JSON과 DOM(HTML/XML)기술을 사용하였다.

입력 요구사항이었던 지도를 통한 지역 설정을 17개의 마커를 이용하였으며, Form을 활용한 입력 및 버튼 사용은 ‘코로나19 감염 현황 보기’라는 버튼을 입력으로 제작하였다.

출력 요구사항 중 테이블은 ‘코로나19 감염 현황’ 및 ‘마커 선택 지역의 긴급재난문자’로 만족시켰다. Naver Map에 대해서는 마커 클릭 시 Bounce 애니메이션을 적용하여 마커가 움직이는 출력을 나타낼 수 있도록 하였다.

이렇게 요구사항들을 만족하며 기말과제를 진행하였는데 한 가지 아쉬운 점이 있었다. 마커에 적용한 Bounce 애니메이션을 해당 지역을 다시 클릭하여 해제할 수 있도록 구현하였는데, 다른 지역을 선택할 때에도 이전에 선택한 지역의 애니메이션이 해제되도록 구현하였으면 더 좋았을 것 같다.

# 결론

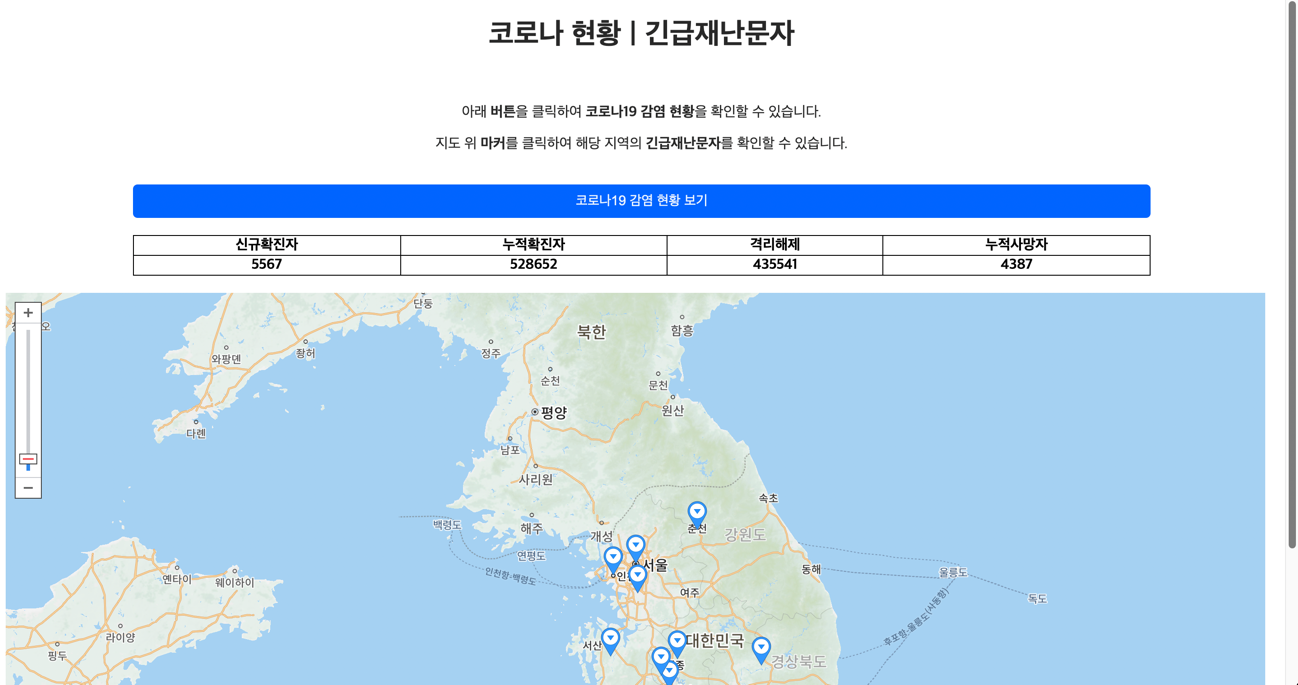
이번 기말과제에서는 단순한 웹페이지 제작을 넘어서 서비스키를 받아 Open API를 활용해보는 좋은 경험을 해볼 수 있었다. 이 과정에서 Ajax, jQuery 등의 기술을 배울 수 있었다. 또한, DOM을 충분히 활용해보고 CSS를 이용하여 직접 버튼을 디자인해보는 등 이전에 공부한 내용도 체화시킬 수 있었다. 새로 접하는 NAVER MAP API의 경우에는 문서를 찾아가며 공부하면서 기본도 더 탄탄히 다질 수 있는 계기가 되었다.

< 웹 페이지 첫 화면 >

지도이(가) 표시된 사진

자동 생성된 설명

< ‘코로나19 감염 현황 보기’ 버튼 클릭 후 화면 >



< ‘서울’ 마커 클릭 후 마커 애니메이션 적용 및 긴급재난문자 출력 화면 >

지도이(가) 표시된 사진

자동 생성된 설명

# 부록

## XML-Ajax

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

## JSON

텍스트이(가) 표시된 사진

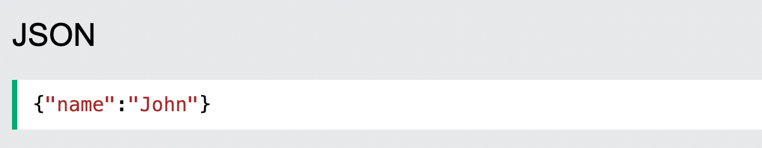
자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명



## 소스코드(.html)

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>코로나 현황 | 긴급재난문자</title>

<style>

h1,

p {

color: #282828;

text-align: center;

}

table,

button {

width: 80%;

margin-left: 10%;

margin-right: 10%;

}

table,

th,

td {

border: 1px solid black;

border-collapse: collapse;

}

th,

td {

paadding: 5px;

}

button {

color: #ebf5ff;

background-color: #0064ff;

border: 1px solid #0064ff;

font-size: 15px;

padding: 8px;

border-radius: 5px;

}

</style>

<!-- Naver Map API -->

<script

type="text/javascript"

src="https://openapi.map.naver.com/openapi/v3/maps.js?ncpClientId=hevdjc9qgc"

></script>

</head>

<body>

<h1>코로나 현황 | 긴급재난문자</h1>

<br />

<p>

아래 <strong>버튼</strong>을 클릭하여

<strong>코로나19 감염 현황</strong>을 확인할 수 있습니다.

</p>

<p>

지도 위 <strong>마커</strong>를 클릭하여 해당 지역의

<strong>긴급재난문자</strong>를 확인할 수 있습니다.

</p>

<br />

<button type="button" onclick="buildTable1()">

코로나19 감염 현황 보기

</button>

<br />

<br />

<table id="covid"></table>

<br />

<div id="map" style="width: 99%; height: 600px"></div>

<br />

<table id="message"></table>

<!-- 공공데이터포털: 코로나19 감염 현황 -->

<script>

function buildTable1() {

var xhr = new XMLHttpRequest();

var url =

"http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson"; /\*URL\*/

var queryParams =

"?" +

encodeURIComponent("serviceKey") +

"=" +

"TBzJLMCgF50KAwTpZ5vpHHWdkzgTOHTojD0gMZfVngqFfcx4MjZwgPjaiCM%2FchJuqKP4iMjqkLBRGnX66AMCgw%3D%3D"; /\*Service Key\*/

queryParams +=

"&" +

encodeURIComponent("pageNo") +

"=" +

encodeURIComponent("1"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("numOfRows") +

"=" +

encodeURIComponent("18"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("startCreateDt") +

"=" +

encodeURIComponent("20211212"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("endCreateDt") +

"=" +

encodeURIComponent("20211231"); /\*\*/

xhr.open("GET", url + queryParams);

xhr.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) {

myFunction(this);

}

};

xhr.send("");

}

function myFunction(xml) {

var incDec = 0; // 신규 확진자

var defCnt = 0; // 누적 확진자

var isolClearCnt = 0; // 격리 해제자

var deathCnt = 0; // 누적 사망자

var xmlDoc = xml.responseXML;

var x = xmlDoc.getElementsByTagName("item");

for (i = 0; i < 18; i++) {

incDec += parseInt(

x[i].getElementsByTagName("incDec")[0].childNodes[0].nodeValue

);

defCnt += parseInt(

x[i].getElementsByTagName("defCnt")[0].childNodes[0].nodeValue

);

isolClearCnt += parseInt(

x[i].getElementsByTagName("isolClearCnt")[0].childNodes[0].nodeValue

);

deathCnt += parseInt(

x[i].getElementsByTagName("deathCnt")[0].childNodes[0].nodeValue

);

}

var table =

"<thead><tr><th>신규확진자</th><th>누적확진자</th><th>격리해제</th><th>누적사망자</th></tr></thead>";

table +=

"<tr><th>" +

incDec +

"</th><th>" +

defCnt +

"</th><th>" +

isolClearCnt +

"</th><th>" +

deathCnt +

"</th></tr>";

document.getElementById("covid").innerHTML = table;

}

</script>

<!-- 공공데이터포털: 긴급재난문자-->

<script>

function buildTable2(location) {

var url =

"http://apis.data.go.kr/1741000/DisasterMsg4/getDisasterMsg2List"; /\*URL\*/

var queryParams =

"?" +

encodeURIComponent("serviceKey") +

"=" +

"TBzJLMCgF50KAwTpZ5vpHHWdkzgTOHTojD0gMZfVngqFfcx4MjZwgPjaiCM%2FchJuqKP4iMjqkLBRGnX66AMCgw%3D%3D"; /\*Service Key\*/

queryParams +=

"&" +

encodeURIComponent("pageNo") +

"=" +

encodeURIComponent("1"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("numOfRows") +

"=" +

encodeURIComponent("5"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("type") +

"=" +

encodeURIComponent("xml"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("create\_date") +

"=" +

encodeURIComponent("2021/06/22 00:00:00"); /\*\*/

queryParams +=

"&" +

encodeURIComponent("location\_name") +

"=" +

encodeURIComponent(location); /\*\*/

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) {

myFunction2(this);

}

};

xhttp.open("GET", url + queryParams);

xhttp.send();

}

function myFunction2(xml) {

var i;

var xmlDoc2 = xml.responseXML;

var table2 = "<tr><th>일시</th><th>지역</th><th>내용</th></tr>";

var x = xmlDoc2.getElementsByTagName("row");

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

table2 +=

"<tr><td>" +

x[i].getElementsByTagName("create\_date")[0].childNodes[0]

.nodeValue +

"</td><td>" +

x[i].getElementsByTagName("location\_name")[0].childNodes[0]

.nodeValue +

"</td><td>" +

x[i].getElementsByTagName("msg")[0].childNodes[0].nodeValue +

"</td></tr>";

}

document.getElementById("message").innerHTML = table2;

}

</script>

<!-- Naver Map-->

<script>

// 지도 생성

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

center: new naver.maps.LatLng(37.566293, 126.977944),

zoom: 7,

zoomControl: true,

mapTypeId: "normal",

});

// 서울특별시

// 마커 생성

var marker1 = new naver.maps.Marker({

position: new naver.maps.LatLng(37.566293, 126.977944),

map: map,

title: "서울특별시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker1, "click", function () {

// 애니메이션 적용

if (marker1.getAnimation() !== null) {

marker1.setAnimation(null);

} else {

marker1.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker1.title);

});

// 강원도

// 마커 생성

var marker2 = new naver.maps.Marker({

position: new naver.maps.LatLng(37.885395, 127.729776),

map: map,

title: "강원도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker2, "click", function () {

// 애니메이션 적용

if (marker2.getAnimation() !== null) {

marker2.setAnimation(null);

} else {

marker2.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker2.title);

});

// 경기도

// 마커 생성

var marker3 = new naver.maps.Marker({

position: new naver.maps.LatLng(37.274904, 127.00919),

map: map,

title: "경기도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker3, "click", function () {

// 애니메이션 적용

if (marker3.getAnimation() !== null) {

marker3.setAnimation(null);

} else {

marker3.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker3.title);

});

// 경상남도

// 마커 생성

var marker4 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.238288, 128.692398),

map: map,

title: "경상남도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker4, "click", function () {

// 애니메이션 적용

if (marker4.getAnimation() !== null) {

marker4.setAnimation(null);

} else {

marker4.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker4.title);

});

// 경상북도

// 마커 생성

var marker5 = new naver.maps.Marker({

position: new naver.maps.LatLng(36.576018, 128.505595),

map: map,

title: "경상북도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker5, "click", function () {

// 애니메이션 적용

if (marker5.getAnimation() !== null) {

marker5.setAnimation(null);

} else {

marker5.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker5.title);

});

// 광주광역시

// 마커 생성

var marker6 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.159606, 126.852349),

map: map,

title: "광주광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker6, "click", function () {

// 애니메이션 적용

if (marker6.getAnimation() !== null) {

marker6.setAnimation(null);

} else {

marker6.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker6.title);

});

// 대구광역시

// 마커 생성

var marker7 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.871539, 128.601505),

map: map,

title: "대구광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker7, "click", function () {

// 애니메이션 적용

if (marker7.getAnimation() !== null) {

marker7.setAnimation(null);

} else {

marker7.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker7.title);

});

// 대전광역시

// 마커 생성

var marker8 = new naver.maps.Marker({

position: new naver.maps.LatLng(36.350454, 127.384819),

map: map,

title: "대전광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker8, "click", function () {

// 애니메이션 적용

if (marker8.getAnimation() !== null) {

marker8.setAnimation(null);

} else {

marker8.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker8.title);

});

// 부산광역시

// 마커 생성

var marker9 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.179789, 129.074987),

map: map,

title: "부산광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker9, "click", function () {

// 애니메이션 적용

if (marker9.getAnimation() !== null) {

marker9.setAnimation(null);

} else {

marker9.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker9.title);

});

// 울산광역시

// 마커 생성

var marker10 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.539619, 129.311527),

map: map,

title: "울산광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker10, "click", function () {

// 애니메이션 적용

if (marker10.getAnimation() !== null) {

marker10.setAnimation(null);

} else {

marker10.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker10.title);

});

// 인천광역시

// 마커 생성

var marker11 = new naver.maps.Marker({

position: new naver.maps.LatLng(37.455993, 126.705255),

map: map,

title: "인천광역시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker11, "click", function () {

// 애니메이션 적용

if (marker11.getAnimation() !== null) {

marker11.setAnimation(null);

} else {

marker11.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker11.title);

});

// 전라남도

// 마커 생성

var marker12 = new naver.maps.Marker({

position: new naver.maps.LatLng(34.816216, 126.462924),

map: map,

title: "전라남도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker12, "click", function () {

// 애니메이션 적용

if (marker12.getAnimation() !== null) {

marker12.setAnimation(null);

} else {

marker12.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker12.title);

});

// 전라북도

// 마커 생성

var marker13 = new naver.maps.Marker({

position: new naver.maps.LatLng(35.820356, 127.108726),

map: map,

title: "전라북도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker13, "click", function () {

// 애니메이션 적용

if (marker13.getAnimation() !== null) {

marker13.setAnimation(null);

} else {

marker13.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker13.title);

});

// 제주특별자치도

// 마커 생성

var marker14 = new naver.maps.Marker({

position: new naver.maps.LatLng(33.48883, 126.498079),

map: map,

title: "제주특별자치도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker14, "click", function () {

// 애니메이션 적용

if (marker14.getAnimation() !== null) {

marker14.setAnimation(null);

} else {

marker14.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker14.title);

});

// 충청남도

// 마커 생성

var marker15 = new naver.maps.Marker({

position: new naver.maps.LatLng(36.659995, 126.67228),

map: map,

title: "충청남도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker15, "click", function () {

// 애니메이션 적용

if (marker15.getAnimation() !== null) {

marker15.setAnimation(null);

} else {

marker15.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker15.title);

});

// 충청북도

// 마커 생성

var marker16 = new naver.maps.Marker({

position: new naver.maps.LatLng(36.635806, 127.491334),

map: map,

title: "충청북도",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker16, "click", function () {

// 애니메이션 적용

if (marker16.getAnimation() !== null) {

marker16.setAnimation(null);

} else {

marker16.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker16.title);

});

// 세종특별자치시

// 마커 생성

var marker17 = new naver.maps.Marker({

position: new naver.maps.LatLng(36.48013, 127.288763),

map: map,

title: "세종특별자치시",

});

// 마커 클릭: 애니메이션 적용, 긴급재난문자 테이블 생성

naver.maps.Event.addListener(marker17, "click", function () {

// 애니메이션 적용

if (marker17.getAnimation() !== null) {

marker17.setAnimation(null);

} else {

marker17.setAnimation(naver.maps.Animation.BOUNCE);

}

// 긴급재난문자 테이블 생성

buildTable2(marker17.title);

});

</script>

</body>

</html>

# 참고자료

• https://www.w3schools.com/xml/ajax\_intro.asp

• https://www.w3schools.com/xml/xml\_http.asp

• https://www.w3schools.com/js/js\_json\_intro.asp

• https://www.w3schools.com/js/js\_json\_syntax.asp

• https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15091495

• https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15043378

• https://navermaps.github.io/maps.js.ncp/docs/tutorial-2-control-options.example.html

• https://navermaps.github.io/maps.js.ncp/docs/tutorial-1-marker-simple.example.html

• https://navermaps.github.io/maps.js.ncp/docs/tutorial-marker-animation.example.html

• https://navermaps.github.io/maps.js.ncp/docs/naver.maps.Marker.html