

# 최종 보고서

---

## 인터넷과웹기초

학 과	정보컴퓨터공학부
학 번	202255549
이 름	박유현
제 출 일	2022-06-06



## 내용

1. 기능 시나리오.....	3
기능 1. 서울 버스 정류장 정보 확인하기 .....	3
기능 2. 서울 지하철역 정보 확인하기 .....	3
기능 3. 기상청 날씨데이터 확인하기 .....	3
2. 최소 개발 구현 .....	4
3. 추가 구현 [가산점].....	16

---

## 1. 기능 시나리오

---

각 기능 시나리오 별로 사용법을 **개괄식**으로 작성  
사용한 Open API 반드시 명시

---

### 기능 1. 서울 버스 정류장 정보 확인하기

사용한 API(1) : <http://data.seoul.go.kr/dataList/OA-15067/S/1/datasetView.do> (서울시 버스정류소 위치정보)

사용한 API(2) : <https://apis.map.kakao.com/web/> (카카오맵 API)

'서울 버스 정류소 리스트' 버튼을 클릭하면 서울특별시의 버스 정류장 목록을 불러온다.

Ctrl + F를 눌러 원하는 버스 정류소의 이름을 찾으면, 해당 text를 복사를 하고 위로가기 버튼을 누른다.

전으로 돌아가기 버튼을 누른 후, 검색창에 정류소 이름을 검색한다.

그러면 해당 정류소의 정류소명, 정류소번호, 경도, 위도를 보여주고,

카카오맵에서 버스 정류소 키워드에 관련된 장소가 마커로 표시되고, 실시간 교통량이 표시된다.

(일반 지도와 스카이뷰도 선택 가능하다.)

### 기능 2. 서울 지하철역 정보 확인하기

사용한 API(1) : <http://data.seoul.go.kr/dataList/OA-12035/S/1/datasetView.do> (서울교통공사 지하철역 주소 및 전화번호 정보)

사용한 API(2) : <https://apis.map.kakao.com/web/> (카카오맵 API)

호선 선택에서 원하는 호선을 선택한다. 예를 들어 4호선을 골랐다고 하면,

서울특별시의 9호선 지하철역 정보를 표로 보여준다.

9호선의 원하는 역을 클릭하면 카카오맵에서 9호선의 선택한 지하철역이 마커로 표시되고, 실시간 교통량이 표시된다. (일반 지도와 스카이뷰도 선택 가능하다.)

(만약 load에 실패하면,

### 기능 3. 기상청 날씨데이터 확인하기

사용한 API(1) : <https://www.data.go.kr/data/15059093/openapi.do> (기상청\_지상(중관,ASOS)일자료\_조회서비스)

확인하고 싶은 날씨데이터의 날짜를 YYYYMMDD형식으로 입력하고 검색버튼을 누르면,

해당 날짜의 서울특별시의 평균기온, 최저 기온, 최고 기온, 일강수량, 순간 최대 풍속, 평균 상대습도가 출력된다.

---

## 2. 최소 개발 구현

---

최소 개발 기준을 어떻게 만족하였는지 **개괄식**으로 작성  
해당하는 소스코드 및 line number를 반드시 포함.

---

### ▪ 서로 다른 input 3개 이상 활용

path1.html의 25th line

```
<a onclick="location.href='/path1/SEOUL.html'"><input type="button" class="button" value="서울 버스 정류소 리스트"></a>
```

path1.html의 29th line

```
<input class="find" type="text" placeholder="검색하고 싶은 서울 정류소명 입력(기본값 종로 1가)">  
<input type="button" id="click" onclick="seoulbus(), makemap()" value="검색">
```

path3.html의 38th line

```
<input class="search" type="text" placeholder="YYYYMMDD형식으로입력">  
<input type="button" id="click" onclick="weather()" value="눌러서 확인하기">
```

### ▪ List 및 table 필수 활용

SEOUL.html의 12th line

```
<table> <!--여기서 table 활용을 했습니다. -->  
<tr>  
<td id="seoulhead">서울특별시 버스정류소 리스트 <br>Ctrl + F를 통해 원하는 정류소를 검색하실 수 있습니다.</td>  
</tr> <!--뒤에 table이 이어집니다.-->
```

path3.html의 31st line

```
<ul><!-- list 활용 -->  
<li id="avgTa"></li>  
<li id="minTa"></li>  
<li id="maxTa"></li>  
<li id="sumRn"></li>  
<li id="maxInsWs"></li>  
<li id="avgRhm"></li>
```

```
</ul>
```

path2.html의 47th line

```
<table id="apitable" class="apitable">
<tr>
<th>호선</th>
<th>역명</th>
<th>구주소</th>
<th>도로명주소</th>
<th>전화번호</th>
</tr>
</table>
```

## ▪ Semantic (header, navigation, footer) 필수 활용

path1.html의 9th, 15th, 50th line

```
<header>
<divclass="title">
<p>서울 버스 정류장 정보 확인하기<br></p>
</div>
</header>
<nav>
<p>
<a href="/main/main.html"title="go-back">메인 화면으로 돌아가기 </a>|
<a href="/path2/path2.html"title="go-subway">서울 지하철역 정보 확인하기</a>|
<a href="/path3/path3.html"title="go-weather">기상청 날씨데이터 확인하기</a>
</p>
</nav>
<footer>
<p>2022 TERM PROJECT 박유현</p>
</footer>
```

path2.html의 9th, 17th, 60th line

```
<header>
<divclass="title">
<p>서울 지하철역 정보 확인하기</p>
</div>
```

```

</header>

<nav>

<p>

<a href="/main/main.html" title="go-back">메인 화면으로 돌아가기 </a>|

<a href="/path1/path1.html" title="go-bus">서울 버스 정류장 정보 확인하기</a>|

<a href="/path3/path3.html" title="go-weather">기상청 날씨데이터 확인하기</a>

</p>

</nav>

<footer>

<p>2022 TERM PROJECT 박유현</p>

</footer>

```

path3.html의 7th, 14th, 42nd line

```

<header>

<div class="title">

<p>기상청 날씨데이터 확인하기<br></p>

</div>

</header>

<nav>

<p>

<a href="/main/main.html" title="go-back">메인 화면으로 돌아가기 </a>|

<a href="/path1/path1.html" title="go-bus">서울 버스 정류장 정보 확인하기</a>|

<a href="/path2/path2.html" title="go-subway">서울 지하철역 정보 확인하기</a>

</p>

</nav>

<footer>

<p>2022 TERM PROJECT 박유현</p>

</footer>

```

## ▪ Pseudo-class 2개 이상

main.css의 59th~64th line, 68th~73rd line

```

.hov-ani:hover.static{ /* pseudo class 활용 */

display: none;

}

```

```

.hov-ani:hover.animated{ /* pseudo class 활용 */
display: inline;
}

.div1:hover{ /* pseudo class 활용 */
background: silver;
}

.div2:hover{ /* pseudo class 활용 */
background: silver;
}

```

path1.css 25th, 34th, 37th, 40th, 58th, 61st

```

.title:hover{ /* pseudo class 활용 */
background-color: silver;
}

.nava:link{ /* pseudo class 활용 */
text-decoration: none;
}

.nava:visited{ /* pseudo class 활용 */
text-decoration: none;
}

.button:active{ /* pseudo class 활용 */
color: black;
}

.button:hover{ /* pseudo class 활용 */
background-color: silver;
}

```

## ▪ pseudo-element 2개 이상

path1.css

```

92     .div1 p::after{ /* pseudo element 활용 */
93     |         content: "(mobile)";
94     |     }
95     }
87     input::placeholder{ /* pseudo element 활용 */
88     |         color: ■rgb(238, 255, 255);
89     |     }

```

path2.css

```

87     .title p::after{ /* pseudo element 활용 */
88         content: "(mobile)";
89     }
90     td,th{

```

path3.css

```

48     .title p::after{ /*pseudo element 사용*/
49         content: " (중관기상관측일자료 데이터 조회 기능)";
50     }
51
52     input::placeholder{ /*pseudo element 사용*/
53         color: #rgb(165, 151, 151);
54     }
55

```

## ▪ attribute selector 2개 이상

main.css

```

47     [src$=".png"]{ /*Attribute Selector 활용*/
48         max-width:100%;
49         min-width: 100%;
50         height: 230px;
51         border-radius: 40px;
52     }
53     [src$=".gif"]{ /*Attribute Selector 활용*/
54         max-width:100%;
55         min-width: 100%;
56         height: 230px;
57         border-radius: 40px;
58     }

```

path2.css

```

49     /*Attribute Selector 활용*/
50     [title|=go]{
51         text-shadow: 1px 1px 2px #gray;
52     }

```

## ▪ combinator 5개 이상 활용

main.css

```

113     .div2 br{ /*combinator 사용 */
114         display: none !important;
115     }
116     .div2 p{ /* combinator 사용 */
117         margin: 10px 0px;
118     }
119 }

```



```

28   nav p{ /* combinator 활용 */
29       font-size: 30px;
30   }
31   nav a{ /* combinator 활용 */
32       color: skyblue;
33   }

```

path1.css

```

68   footer p{ /* combinator 활용 */
69       color: gray;
70   }

```

path2.css

```

31   footer p{ /* combinator 활용 */
32       color: gray;
33   }
34   nav p{ /* combinator 활용 */
35       font-size: 30px;
36   }
37   nav a{ /* combinator 활용 */
38       color: skyblue;
39   }

```

path3.css

```

76   select option{ /* combinator 활용 */
77       background: lightgray;
78   }

```

## ▪ Float 및 clear 활용하여 웹의 layout 구성 필수

path3.css

```

18   #weather{ /* float 활용 */
19       float: left;
20       border: 3px solid rgb(71, 155, 79);
21       width: 40%;
22       border-style: dashed;
23       margin-left: 80px;
24       margin-top: 30px;
25       margin-bottom: 30px;
26       border-radius: 40px;
27   }

```

```

82     footer{ /* clear 활용 */
83         clear: both;
84         text-align: center;
85         margin-top: 30px;
86     }

```

## ▪ 모바일 환경 대응 구현 필수 (width: 900px 이하)

main.css

```

76  @media only screen and (max-width: 900px){ /*반응형 웹(media query 사용)*/
77      .container {
78          display: grid;
79          grid-template-columns: repeat(1, 1fr);
80          grid-template-rows: repeat(4, auto);
81          row-gap: 60px;
82      }
83      .div1 {
84          font-family: fantasy;
85          font-size : 40px;
86          border: 2px solid black;
87          grid-column-start: 1;
88          grid-column-end: 1;
89          background-color: sandybrown;
90          border-radius: 40px;
91      }
92      .div1 p::after{ /* pseudo element 활용 */
93          content: "(mobile)";
94      }
95      img{
96          max-width:100%;
97          min-width: 100%;
98          height: 300px;
99          border-radius: 40px;
100     }
101     .div2 {
102         font-family: fantasy;
103         font-size : 30px;
104         border: 2px solid black;
105         text-align: center;
106         max-width: 100%;
107         max-height:100%;
108         min-width: 100%;
109         min-height:100%;
110         background-color: sandybrown;
111         border-radius: 40px;
112     }
113     .div2 br{ /*combinator 사용 */
114         display: none !important;
115     }
116     .div2 p{ /* combinator 사용 */
117         margin: 10px 0px;
118     }
119 }

```

path1.css

```

119 @media only screen and (max-width: 900px){ /*반응형 웹(media query 사용)*/
120     .container {
121         display: grid; /*grid를 사용하여 레이아웃을 구성함*/
122         grid-template-columns: repeat(1, 1fr);
123         grid-template-rows: repeat(3, auto);
124         row-gap: 50px;
125         column-gap: 50px;
126     }
127     .title {
128         width: auto;
129         max-width: 100%;
130         min-width: 100%;
131         font-family: fantasy;
132         font-size : 50px;
133         border: 4px solid black;
134         margin-bottom: 30px;
135         text-shadow: 1px 1px 2px gray;
136         background-color: sandybrown;
137         border-radius: 40px;
138     }
139     .list{
140         grid-column-start: 1;
141         grid-column-end: 2;
142     }
143     .title p::after{ /* pseudo element 활용 */
144         content: "(mobile)";
145     }
146     .item P::after{ /* pseudo element 활용 */
147         content: "(moblie)";
148     }
149     .item:nth-child(2) {order: 3;}
150     .item:nth-child(3) {order: 2;}
151 }

```

## path2.css

```

86 @media only screen and (max-width: 900px){
87     .title p::after{ /* pseudo element 활용 */
88         content: "(mobile)";
89     }
90     td, th{
91         border: 1.5px solid rgb(121, 0, 0);
92         color:rgb(46, 45, 51);
93     }
94     table{
95         width: 100%;
96         border: 1px solid rgb(231, 220, 220);
97         border-style:dashed;
98         background-color: rgb(196, 176, 176);
99     }
100     .kakaomap{
101         border: 1.5px solid rgb(255, 198, 198);
102         border-style:dashed;
103         float: left;
104     }
105     .title {
106         font-family: fantasy;
107         font-size : 50px;
108         border: 4px solid black;
109         margin-bottom: 30px;
110         text-shadow: 1px 1px 2px gray;
111         background-color: sandybrown;
112         border-radius: 40px;
113     }
114
115 }
116

```

## path3.css

```

106
107 @media only screen and (max-width: 900px){
108     .title p::after{ /* pseudo element 활용 */
109         content: "(mobile)";
110     }
111     .title {
112         font-family: fantasy;
113         font-size : 50px;
114         border: 4px solid black;
115         margin-bottom: 30px;
116         text-shadow: 1px 1px 2px gray;
117         background-color: sandybrown;
118         border-radius: 40px;
119     }
120     #weather{
121         text-align: center;
122         border: 3px solid rgb(205, 205, 205);
123         width: 100%;
124         border-style: dashed;
125         margin: auto;
126         border-radius: 40px;
127     }
128     .container{
129         display: grid;
130         grid-template-columns: repeat(1, 1fr);
131         grid-template-rows: repeat(2, auto);
132     }
133     .list{
134         width: 100%;
135         text-align: center;
136     }
137 }

```

## SEOUL.css

```

63
64 @media only screen and (max-width: 900px){ /*반응형 웹(media query 사용)*/
65     #seoulhead::after{
66         content: "(moblie)";
67     }
68     #seoulhead{
69         background: rgb(162, 174, 145);
70         color: black;
71     }
72     .topmenu{
73         position: fixed;
74         font-size: 30px;
75         bottom: 0%;
76         right: 0%;
77         border: 4px solid gray;
78         text-shadow: 1px 1px 2px rgb(227, 213, 213);
79         border-style: dashed;
80     }
81     .button{
82         font-size : 30px;
83         text-shadow: 1px 1px 2px gray;
84         background-color: rgb(145, 123, 96);
85     }
86     a{
87         color: rgb(32, 30, 30);
88     }
89     body{
90         display: flex;
91         justify-content: center;
92         text-align: center;
93         background-color: rgb(246, 183, 183);
94     }
95 }

```

## ▪ flex 혹은 grid 한번 이상 필수 활용

### main.css

```

31 .container {
32     display: grid; /*grid를 사용하여 레이아웃을 구성함*/
33     grid-template-columns: repeat(3, 1fr);
34     grid-template-rows: repeat(3, auto);
35     column-gap: 60px;
36     row-gap: 40px;
37 }

```

이외에 path1.css, path2.css, path3.css의 media query, SEOUL.css 에서도 grid를 활용했습니다.

```
6 .container {
7   display: grid; /*grid를 사용하여 레이아웃을 구성함*/
8   grid-template-columns: repeat(2, 1fr);
9   grid-template-rows: repeat(2, auto);
10  row-gap: 50px;
11  column-gap: 50px;
12 }
```

```
7 .container {
8   display: grid; /*grid를 사용하여 레이아웃을 구성함*/
9   grid-template-columns: repeat(1, 1fr);
10  grid-template-rows: repeat(3, auto);
11  row-gap: 50px;
12 }
```

```
31 .container{
32   display: grid;
33   grid-template-columns: repeat(1, 1fr);
34   grid-template-rows: repeat(2, auto);
35   row-gap: 10px;
36 }
```

## ▪ DOM 활용하여 HTML element 혹은 CSS property 합쳐서 5개 이상 다루기

path1.html

```
66 console.log(this.responseText);
67 data = JSON.parse(this.responseText); /* response 받는 datatype이 JSON이기 때문에 JSON 형식으로 parse 받도록 함 */
68 document.getElementById("busname").innerHTML = "정류소명 : " + data.busStopLocationXyInfo.row[0].STOP_NM;
69 document.getElementById("busno").innerHTML = "정류소번호 : " + data.busStopLocationXyInfo.row[0].STOP_NO;
70 document.getElementById("gpsx").innerHTML = "GPS X좌표(경도) : " + data.busStopLocationXyInfo.row[0].XCOD;
71 document.getElementById("gpsy").innerHTML = "GPS Y좌표(위도) : " + data.busStopLocationXyInfo.row[0].YCOD;
72 }
73 catch (e) { /* 에러 처리 예정 */ }
```

path2.html

```
72 delete_tr();
73 var select = document.getElementById("box").options[document.getElementById("box").selectedIndex].value;
74 xhr.addEventListener("readystatechange", function() {
75   if(this.readyState === 4) {
76     try{
77       console.log(this.responseText);
78       Data = JSON.parse(this.responseText);
79       for(let i=0 ; i<Data.StationAdresTelno.row.length ; i++){
80         var table = document.getElementById("apitable");
81         var row = table.insertRow();
82         var cell1 = row.insertCell(0);
83         var cell2 = row.insertCell(1);
84         var cell3 = row.insertCell(2);
85         var cell4 = row.insertCell(3);
86         var cell5 = row.insertCell(4);
87         cell1.innerHTML += Data.StationAdresTelno.row[i].LINE;
88         cell2.innerHTML += Data.StationAdresTelno.row[i].STATN_NM ;
89         cell3.innerHTML += Data.StationAdresTelno.row[i].ADRES;
90         cell4.innerHTML += Data.StationAdresTelno.row[i].RDNMADR;
91         cell5.innerHTML += Data.StationAdresTelno.row[i].TELNO ;
92         cell1.id = `table${5*i+1}`;
93         cell2.id = `table${5*i+2}`;
94         cell3.id = `table${5*i+3}`;
95         cell4.id = `table${5*i+4}`;
96         cell5.id = `table${5*i+5}`;
97         console.log(cell1)
98         console.log(cell2)
99         console.log(cell3)
100        console.log(cell4)
101        console.log(cell5)
102      }
```

path3.html

```

66 document.getElementById("tm").innerHTML = DATA.getElementsByTagName("tm")[0].textContent + "의 서울의 날씨데이터 : ";
67 document.getElementById("avgTa").innerHTML = "평균 기온은 '" + DATA.getElementsByTagName("avgTa")[0].textContent + "°C' 입니다.";
68 document.getElementById("minTa").innerHTML = "최저 기온은 '" + DATA.getElementsByTagName("minTa")[0].textContent + "°C' 입니다.";
69 document.getElementById("maxTa").innerHTML = "최고 기온은 '" + DATA.getElementsByTagName("maxTa")[0].textContent + "°C' 입니다.";
70 document.getElementById("sumRn").innerHTML = "일 강수량은 '" + sumRn + "mm' 입니다.";
71 document.getElementById("maxInsWs").innerHTML = "순간 최대 풍속은 '" + DATA.getElementsByTagName("maxInsWs")[0].textContent + "m/s' 입니다.";
72 document.getElementById("avgRhm").innerHTML = "평균 상대습도는 '" + DATA.getElementsByTagName("avgRhm")[0].textContent + "%" ' 입니다.";
73
74 tch (e) {

```

## ▪ 입력 Format, 네트워크 연결 등의 에러 처리 기능 필수 구현

path1.html

```

72     }
73     catch (e) { /* 에러 처리 구현 */
74         alert("서울에는 없는 이름의 정류소입니다. 다시 입력하여 주세요.");
75     }
76 }
77 });
78 xhr.open("GET", "http://localhost:5500/seoulapi?word=" + search);
79 xhr.send();
80 }

```

path2.html

```

101     console.log(cell15)
102 }
103 }catch (e) {
104     alert("호선을 선택하여 주세요.");
105 }
106 };
107 });
108 xhr.open("GET", "http://localhost:5500/subwayapi?word=" + select);
109 xhr.send();
110 }

```

path3.html

```

72 document.getElementById("avgkkm").innerHTML = "평균 강대습
73 }
74 catch (e) {
75     alert("YYYYMMDD 형식에 맞게 다시 입력해주세요.");
76 }
77 }
78 });
79 xhr.open("GET", "http://localhost:5500/subwayapi?word=" + select);

```

index.js

```

18     });
19     request(options, function (error, response) {
20         if (error) { /* 에러 처리 구현 */
21             res.end("error");
22         } else {
23             res.send(response.body);
24         }
25     });
26 });
27
28

```

```

38     };
39     request(options, function (error, response) {
40         if (error) { /* 에러 처리 구현 */
41             res.end("error");
42         } else{
43             res.send(response.body);
44         }
45     });
46 });
47

```

```

60     request(options, function (error, response) {
61         if (error) { /* 에러 처리 구현 */
62             res.end("error");
63         }
64         else{
65             res.send(response.body);
66         }
67     });
68 });
69
70

```

## ▪ 수업시간에 배운 jQuery/AJAX/JSON 기능 활용

path2.html(jQuery)

```

117     $("#apitable").on("click", "tr", function(){
118         var tr = $(this);
119         var td = tr.children();
120         var search = td.eq(1).text() + '역' + td.eq(0).text();
121         console.log(search);
122         addEventListener("click", makemap(search));
123     });
124

```

---

### 3. 추가 구현 [가산점]

---

최소 개발 기준 이외에 추가 구현한 부분을 **개괄식**으로 작성  
해당하는 소스코드 및 line number를 반드시 포함.

---

#### ▪main.html, main.css

a tag로 div를 감싸 클릭하면 각 path로 이동하도록 함.(13, 22, 31)

```
<a onclick="location.href='/path1/path1.html'"><!-- a tag를 전체 div를 감싸서 활용, 클릭하면 이동-->
```

animation effect를 활용하여 gif와 png를 넣고, 마우스를 올리면 gif가 실행되도록 함(16, 17, 25, 26, 34, 35) 또한 대체 텍스트(alt) 명시.(css display element, pseudo class(hover) 활용)

```



59  .hov-ani:hover .static { /* pseudo class 활용 */
60      display: none;
61  }
62  .hov-ani:hover .animated { /* pseudo class 활용 */
63      display: inline;
64  }
65  .hov-ani .animated {
66      display: none;
67  }
```

media query에서 <br> tag를 제거하기 위해 display element를 활용함.

```
113  .div2 br { /*combinator 사용 */
114      display: none !important;
115  }
```

디자인을 위해 border-radius 사용.

```
28  background-color: #bisque;
29  border-radius: 40px;
30  }
```

#### ▪path1.html, path1.css

media query에서 :nth-child 를 사용하여 div의 order를 바꿔줌.(change grid layout)

```
149  .item:nth-child(2) {order: 3;}
150  .item:nth-child(3) {order: 2;}
151  }
```

input의 onclick event를 활용하여 js 함수를 불러오도록함.

```
28  <!--
29  out 사용 --> <input type="button" id="click" onclick="seoulbus(), makemap()" value="검색">
30  </--
```



변수 search를 지정하여 input의 value값 추출. 또한 변수 data를 JSON 형식으로 parse 받도록 지정.

```
57 <script>
58     var data = {}
59     function seoulbus(){ /* 서울 버스 정류소에 관한 정보를 띄게 하는 function */
60         var xhr = new XMLHttpRequest();
61         xhr.withCredentials = true;
62         var search = document.querySelectorAll("input")[1].value; /* 사용한 input
63         xhr.addEventListener("readystatechange", function() {
64             if(this.readyState === 4) {
65                 try{
66                     console.log(this.responseText);
67                     data = JSON.parse(this.responseText); /* response 받는 datatype이 JS
```

## ▪path2.html, path2.css

select, option tag 사용(onchange event 활용하여 JS 함수 불리하도록 함.)

```
34 <select id="box" onchange="seoulsubway()">
35     <option id="select" value="호선 선택">호선 선택</option>
36     <option id="select" value="1호선">1호선</option>
37     <option id="select" value="2호선">2호선</option>
38     <option id="select" value="3호선">3호선</option>
39     <option id="select" value="4호선">4호선</option>
40     <option id="select" value="5호선">5호선</option>
41     <option id="select" value="6호선">6호선</option>
42     <option id="select" value="7호선">7호선</option>
43     <option id="select" value="8호선">8호선</option>
44     <option id="select" value="9호선">9호선</option>
45 </select>
```

O호선을 선택했다가 X호선을 선택하면 O호선에 관한 table이 사라질 수 있도록 JS 함수 생성

```
71 xhr.withCredentials
72 delete_tr();
73 var select = document
```

```
112 function delete_tr() {
113     let a = document.querySelectorAll("tr").length;
114     for(let i=1 ; i<a ; i++) document.querySelectorAll("tr")[i].remove();
115 }
116
```

JS td.eq.text/addEventListener event 활용(search값(OO역 O호선)을 카카오맵 api 키워드(parameter) 값으로 넘겨줌)

```
117 $("#apitable").on("click", "tr", function(){
118     var tr = $(this);
119     var td = tr.children();
120     var search = td.eq(1).text()+ '역 ' + td.eq(0).text();
121     console.log(search);
122     addEventListener("click", makemap(search));
123 });
```

## ▪path3.html, path3.css

a tag의 title attribute를 활용(path1, path2도 활용)

```
15 <p>
16 <a href="/main/main.html" title="go-back">메인화면으로 돌아
17 <a href="/path1/path1.html" title="go-bus">서울 버스 정류장
18 <a href="/path2/path2.html" title="go-subway">서울 지하철역
19 </p>
```

input의 onclick event를 활용하여 JS 함수를 실행하도록 함.

```

37 확인하고 싶은 날씨데이터의 날짜를 입력해주세요.<br>*확인할 수 있는 날짜의 범위
38 2로 입력" > <input type="button" id="click" onclick="weather()" value="눌러서 확인" />
39
40

```

변수 day에 input value값 설정. try/catch 사용. DOMparser.parsefromstring() 활용

```

57     try{
58         console.log(this.responseText);
59         parser=new DOMParser();
60         var DATA=parser.parseFromString(this.responseText,"text/xml");
61         console.log(DATA);
62         var sumRn = DATA.getElementsByTagName("sumRn")[0].textContent;
63         if (sumRn.length == 0){
64             sumRn += "0";
65         }
66         document.getElementById("tm").innerHTML = DATA.getElementsByTagName("tm")[0].textContent;
67         document.getElementById("avgTa").innerHTML = "평균 기온은 " + DATA.getElementsByTagName("avgTa")[0].textContent;
68         document.getElementById("minTa").innerHTML = "최저 기온은 " + DATA.getElementsByTagName("minTa")[0].textContent;
69         document.getElementById("maxTa").innerHTML = "최고 기온은 " + DATA.getElementsByTagName("maxTa")[0].textContent;
70         document.getElementById("sumRn").innerHTML = "일 강수량은 " + sumRn;
71         document.getElementById("maxInsWs").innerHTML = "순간 최대 풍속은 " + DATA.getElementsByTagName("maxInsWs")[0].textContent;
72         document.getElementById("avgRhm").innerHTML = "평균 상대습도는 " + DATA.getElementsByTagName("avgRhm")[0].textContent;
73     }
74     catch (e) {
75         alert("YYYYMMDD 형식에 맞게 다시 입력해주세요.")
76     }

```

## •SEOUL.html

SEOUL.css에서 position attribute 활용, justify-content property 활용

```

88 .topmenu{
89     position: fixed;
90     font-size: 30px;
91     bottom: 10%;
92     right: 10%;
93     border: 2px solid gray;
94     text-shadow: 1px 1px 2px gray;
95     border-style: dashed;
96 }
97
98 body{
99     display: flex;
100     justify-content: center;
101     text-align: center;
102     background-color: rgb(246, 183, 183);
103 }

```

a tag의 hypertext-reference를 활용하여 '위로 가기' 기능 구현

```

32794 <div class="topmenu">
32795     <a href="#front">위로 가기</a>
32796 </div>

```