

POOL:US

축제사이트 클론코딩

ARTIST PAGE

GARDEN ISLAND



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

DAY1

DAY2

**소수빈**
18:50

DAY1

DAY2

**권진아**
20:00

DAY1

DAY2

**LUCY**
20:40

DAY1

DAY2

**블블간**
사춘기
21:30

ARTIST PAGE


축제사이트 클론코딩

ARTIST PAGE

GARDEN ISLAND


DAY1

DAY2

**EZUZ**
20:40


DAY1

DAY2

**BewhY**
21:20

DAY1

DAY2

**10CM**
22:00



DAY1, DAY2 페이지
이동 가능한 버튼

동일한 덩어리의 연속!
한 덩어리만 잘 만들어 놓으면
그 다음은 쉽다.



LIKELION SKU

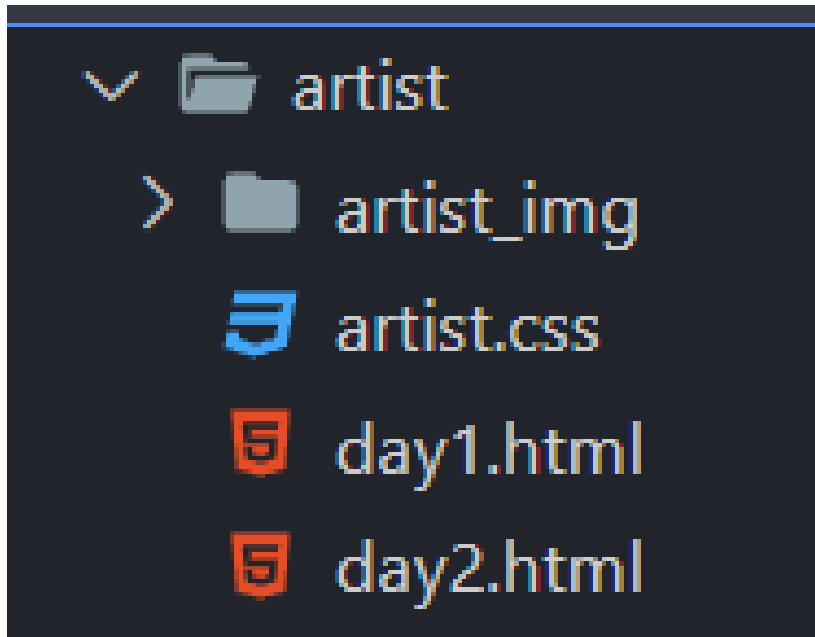
GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

기본 준비 & 레이아웃 잡기

ARTIST PAGE

GARDEN ISLAND

기본 준비



👉 **artist 폴더 생성**

👉 **artist_img 폴더 내 이미지 다운로드**

👉 **day1.html 파일 생성**

👉 **day2.html 파일 생성**

👉 **artist.css 파일 생성**

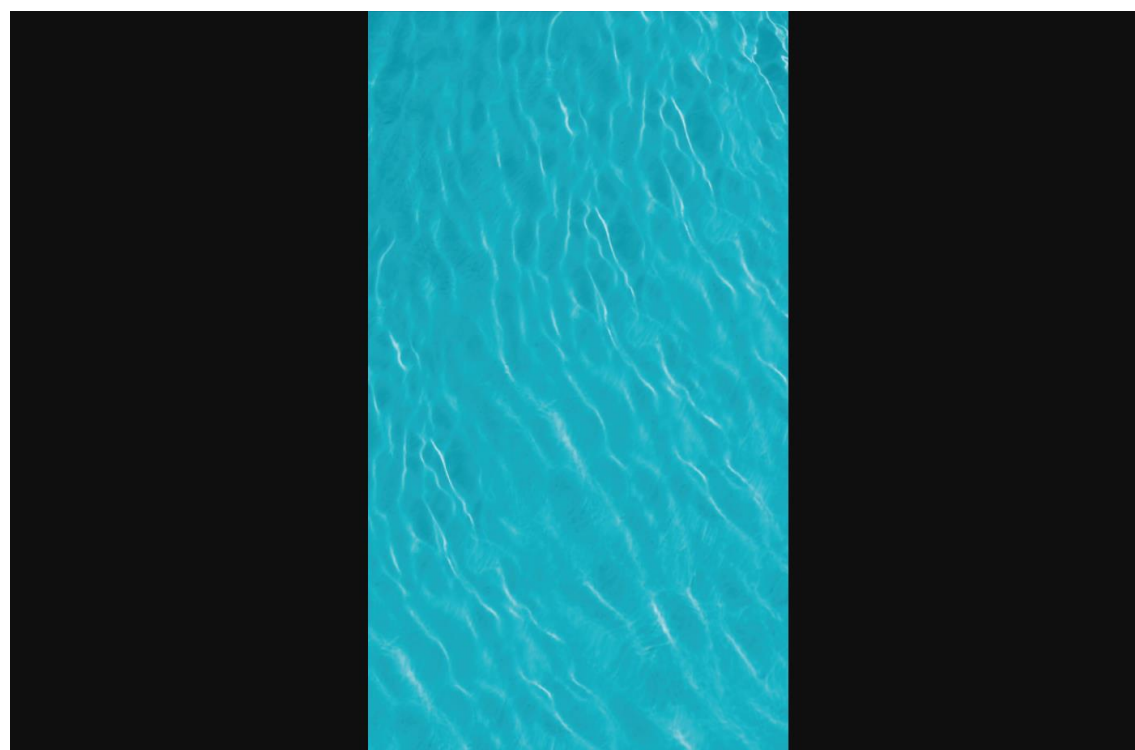
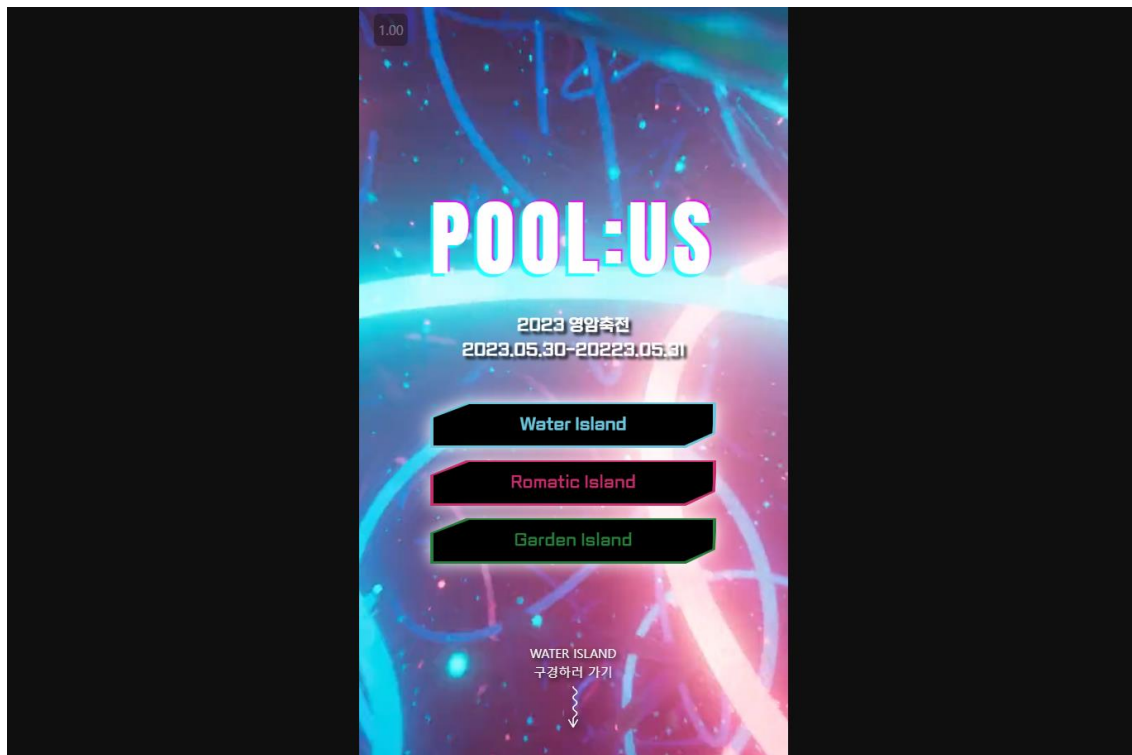
👉 **html 파일과 css 파일 연결하기**

👉 **bootstrap 연결하기**

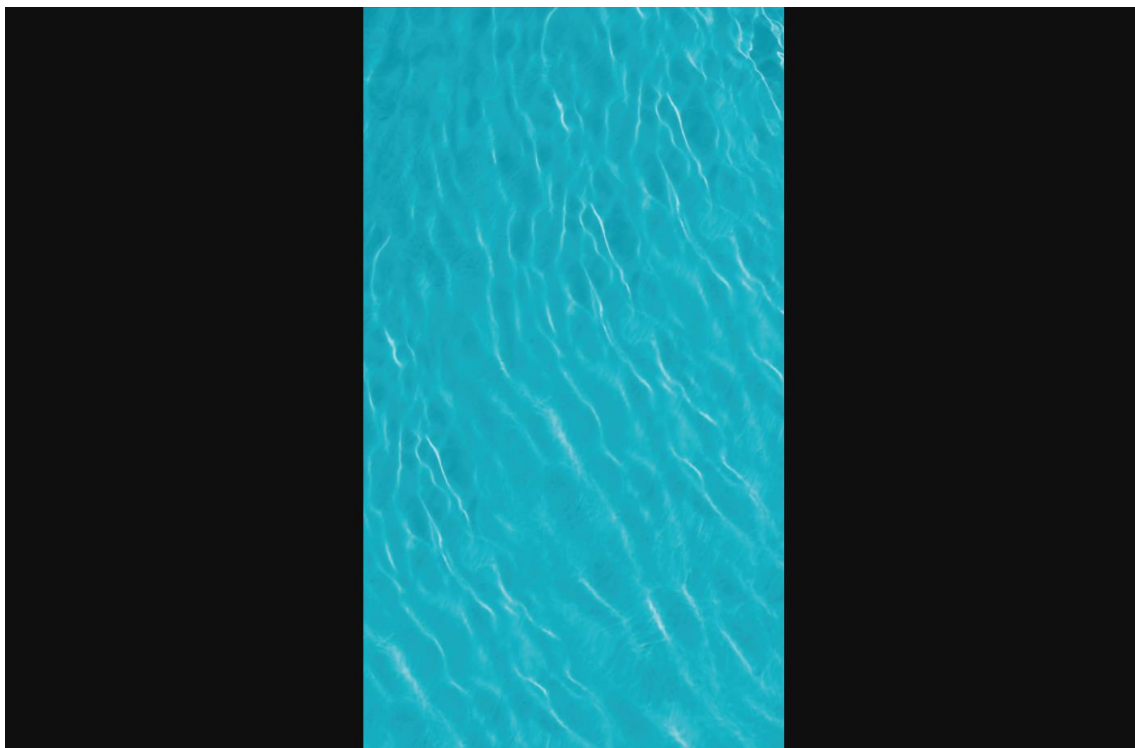


```
1 <!doctype html>
2 <html lang="ko">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>POOL:US</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
9     <link rel="stylesheet" href="./artist.css">
10 </head>
11
12 <body>
13     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
14 </body>
15
16 </html>
```

모든 페이지의 레이아웃을 동일하게 맞춰주자!



동일하게 만들어보기



기존 **display-container** 활용
배경 이미지 활용으로 간단하게 할 수 있다!


```
body >  
<div class="display-container">  
</div>  
<script src="https://cdn.jsdelivr.net" data-bbox="39 493 427 647">
```

```
body {  
  background-color: #0f0f0f;  
}  
  
.display-container {  
  box-shadow: 0 0 15px 0 hsla(0, 0%, 100%, .75);  
  max-width: 415px;  
  margin-left: auto;  
  margin-right: auto;  
  min-height: 100vh;  
  background-image: url('./artist_img/waterback.png');  
  background-size: cover;  
}
```

아 그리고 폰트 먼저 설정해줍시다.

‘롯데리아 딱붙어체’ 검색 후 **Css**에 적용해보기
눈누 사용 권장 ~

저번 시간에 **폰트를 클래스화** 해서 사용했었죠?
동일한 방법으로 적용해보세요 !



눈누에서
복붙

```
@font-face {  
  font-family: 'LOTTERIADDAG';  
  src: url('https://cdn.jsdelivr.net/gh/projectnoon');  
  font-weight: normal;  
  font-style: normal;  
}
```

```
.ddag_font {  
  font-family: 'LOTTERIADDAG';  
}
```

```
dy>  
<div class="display-container ddag_font">  
</div>
```

클래스화

DAY 버튼 직접 만들어보기!



정답은 없습니다.
먼저 자신이 원하는 UI에 맞게 구현해보세요.

☞ `<div><a>DAY1</div>` 형태

☞ 하이퍼링크의 기본속성 지워야겠죠?

☞ DAY1, DAY2 일때 활성화 상태 체크(투~ 명도 조절~)

☞ 부트스트랩 사용하기

☞ d-flex~?

☞ px~ py~ ? 패딩과 마진 이용해보기



```
day/
<div class="display-container ddag_font">
  <!-- Day button -->
  <nav class="d-flex justify-content-center fixed-top">
    <div class="day p-4 mx-3"><a href="#">DAY1</a></div>
    <div class="day DayUnAct p-4 mx-3"><a href="./day2.html">DAY2</a></div>
  </nav>
</div>
```

```
.day {
  border-bottom: 5px solid black;
  font-size: 20px;
  color: white;
}

.DayUnAct {
  border-bottom: 5px solid rgb(0, 0, 0, 0.5);
  color: rgb(255, 255, 255, 0.8);
}

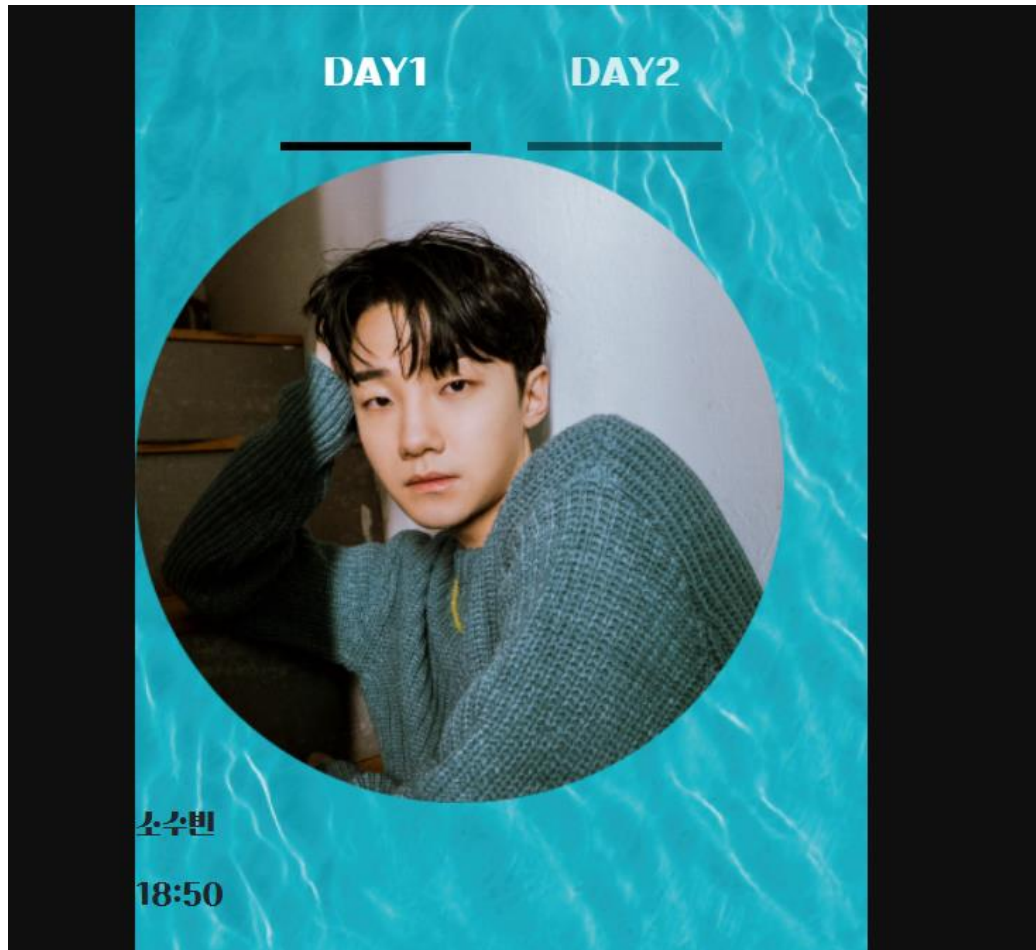
.day a {
  text-decoration: none;
  color: inherit;
}
```

전체적인 구조를 먼저 잡자
먼저 박스 형태로 볼 수 있는
눈이 필요함!



```
<section class="artist_section">
  <div>
    <div></div>
    <p>소수빈</p>
    <p>18:50</p>
  </div>
</section>
```

💡 구조를 잡을 때 인라인 요소들을 최대한 사용하지 않으려고 한다.
블록 요소 형태가 훨씬 레이아웃을 쉽게 잡을 수 있기 때문



Css 없이 기본적인 구조만 잡았을 때의 상태
Css로 예쁘게 디자인 해보자

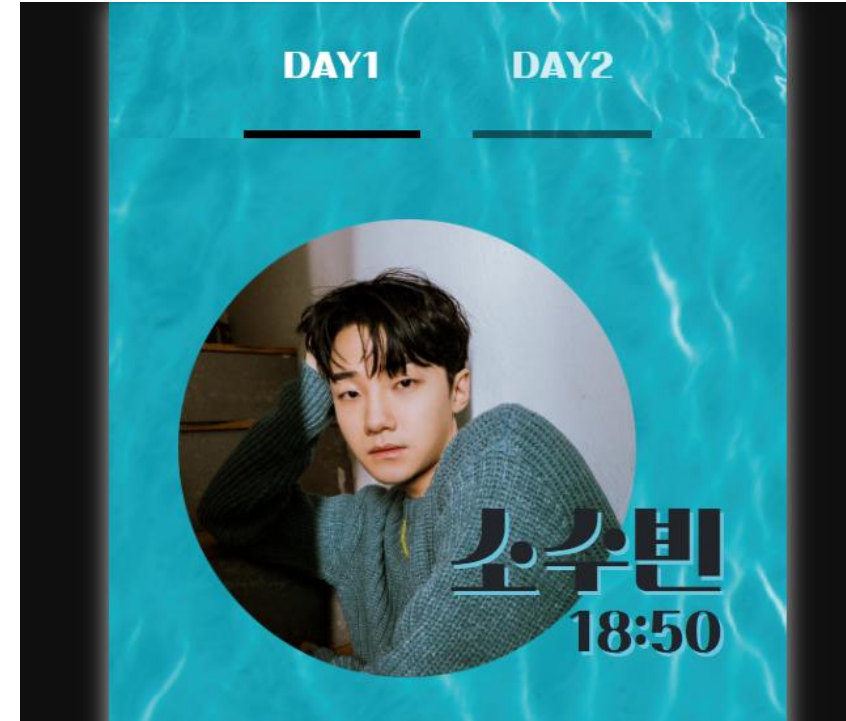
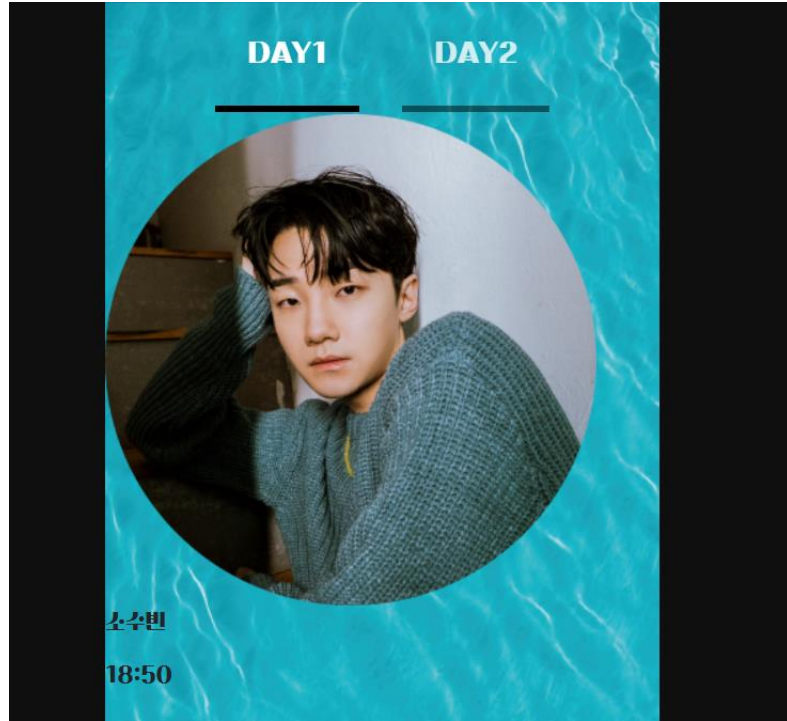
이미지, 글씨 크기와 여백 등을 고려하여
자신의 스타일대로 변경해보기

💡 **사용가능한 속성**
Padding, margin, position
Width, height 등등



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.



Position 속성 이용시 개발자 도구 활용을 적극 권장합니다.
시각적으로 직접 보고 조절한 후에
적용하는 것이 아주 편리하기 때문입니다. haha



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

```
<section class="artist_section">
  <div class="artist_text-center py-5">
    <div class="artist_img"></div>
    <p class="artist_name">소수빈</p>
    <p class="artist_time">18:50</p>
  </div>

  <div class="artist_text-center py-5">
    <div class="artist_img_r"></div>
    <p class="artist_name_r">권진아</p>
    <p class="artist_time_r">20:00</p>
  </div>

  <div class="artist_text-center py-5">
    <div class="artist_img"></div>
    <p class="artist_name">LUCY</p>
    <p class="artist_time">20:40</p>
  </div>

  <div class="artist_text-center py-5">
    <div class="artist_img_r"></div>
    <p class="artist_name_r" style="font-size:45px; margin-bottom:20px">볼빨간 사춘기</p>
    <p class="artist_time_r">21:30</p>
  </div>
</section>
```

```
.artist {
  position: relative;
}

.artist_img img {
  width: 280px;
  margin-right: 50px;
}

.artist_name {
  font-size: 60px;
  position: absolute;
  right: 38px;
  bottom: 63px;
  text-shadow: 2px 2px #6BC6DD;
}

.artist_time {
  font-size: 30px;
  position: absolute;
  right: 40px;
  bottom: 37px;
  text-shadow: 2px 2px #6BC6DD;
}
```



이렇게 한 덩어리를 만들었다면 다 끝난 것 !
이미지 변경, 텍스트 변경
세부적인 속성 변경 후 나열하면 끝 !!!

먼저 복붙으로 여러 개의 덩어리를 만들어주자



문제 발생

Nav 아래로 콘텐츠들이 표시된다.
스크롤 시에도
어떻게 해결할 것인가?

방법1.
Nav와 section이 겹치지 않도록
Section의 바깥 여백을 준다.

방법2.
Nav에 배경을 준다.



방법1.

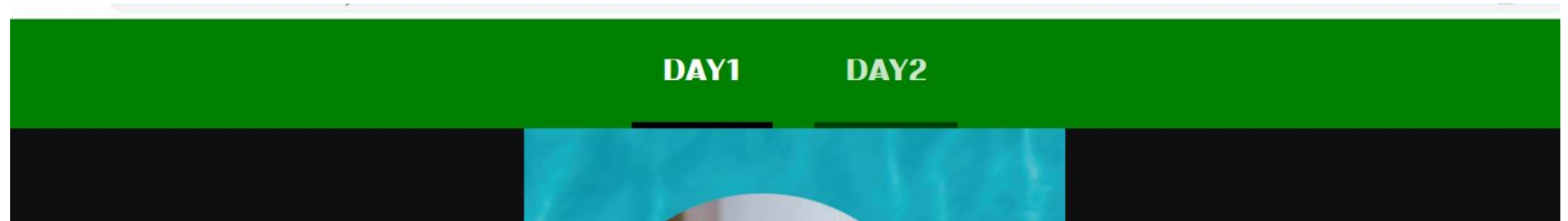
Nav와 section이 겹치지 않도록
Section의 바깥 여백을 준다.

```
.artist_section {  
  padding-top: 85px;  
}
```

방법2.

Nav에 배경을 준다.

```
nav {  
  background-color: green;  
}
```



```
dy>  
<div class="display-container ddag_font">  
  <!-- Day button -->  
  <nav class="d-flex justify-content-center fixed-top">  
    <div class="day p-4 mx-3"><a href="#">DAY1</a></div>  
    <div class="day DayUnAct p-4 mx-3"><a href="./day2.htm">DAY2</a></div>  
  </nav>  
  
  <section class="artist_section">  
    <div class="artist text-center py-5">  
      <div class="artist_img">  
      <p class="artist name">소수빈</p>  
    </div>  
  </section>  
</div>
```

분명 display-container 내부에
Nav가 있는데 왜 벗어나는 것일까

WHY????

```
nav {  
  width: 415px;  
  margin: 0 auto;  
  background-image: url('./artist_img/waterback.png');  
  background-size: cover;  
}
```

**Fixed-top으로 레이아웃에서 벗어난 상태.
어쩔 수 없다. 우리가 맞춰주자**

실습 TIME (★ ω ★)

자 이제 DAY1의 아티스트들을 수정해서 DAY1을 완성해봅시다
DAY2는 숙제 ~

GARDEN ISLAND

축제사이트 클론코딩

ARTIST PAGE

GARDEN ISLAND



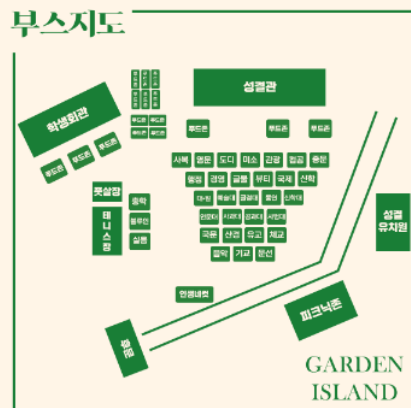
GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

GARDEN ISLAND POOL-US

📍성결관 앞 가든 아일랜드

✧ 부스 지도를 클릭하시면 크게 볼 수 있어요 ✧

부스지도



원하는 부스 또는 음식을 검색해 보세요!

원하는 부스 또는 음식을 검색해 보세요!



동아리연합회 부스

바나나 브릴레 + 아이스크림
바나나 브릴레



블루인 부스

떡볶이
아이스크림
약과 아이스크림



대의원회 부스

화채
미숫가루
냉커피



사회과학대학 부스

왕 소시지
봉봉 에이드



인문대학 부스

소프트 아이스크림
레인보우 아이스크림
허쉬 초코링 아이스크림

GARDEN MAIN

GARDEN BOOTH

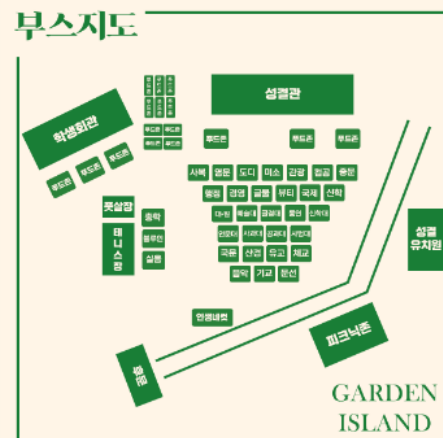


GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

GARDEN ISLAND
POOL:US

📍성결관 앞 가든 아일랜드

✧ 부스 지도를 클릭하시면 크게 볼 수 있어요 ✧



원하는 부스 또는 음식을 검색해 보세요!



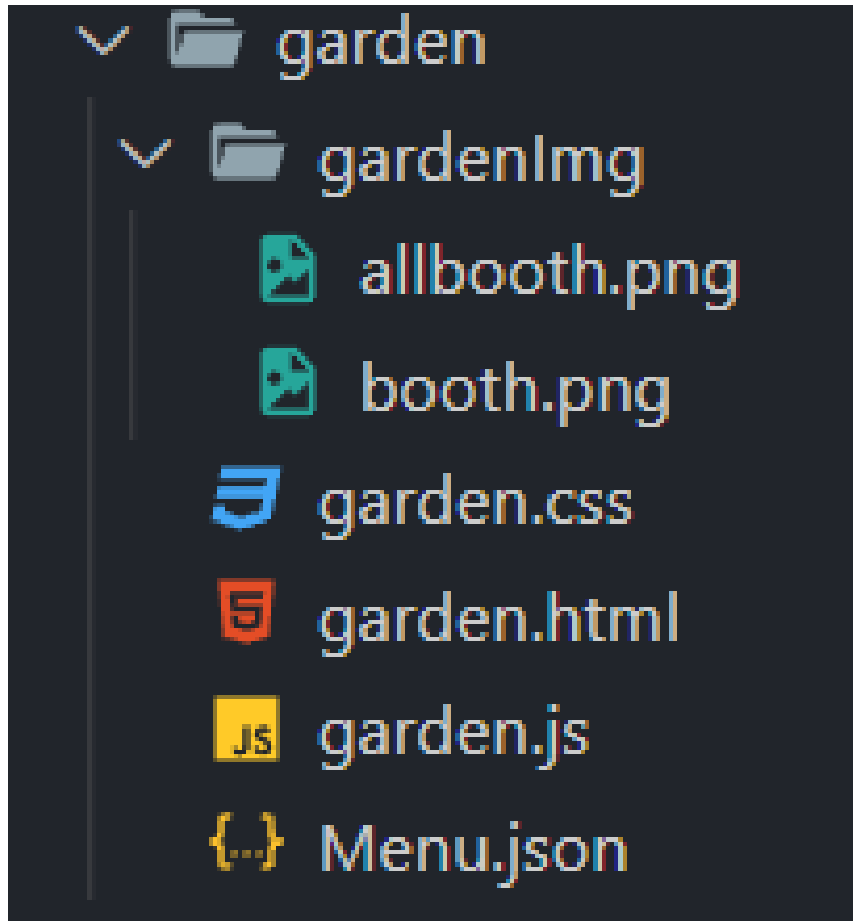
GARDEN MAIN

HTML, CSS



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.



```
den > garden.html > html > body > div.display-container
1 <!doctype html>
2 <html lang="ko">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-s
7   <title>Garden Island</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dis
9   <link rel="stylesheet" href="./garden.css">
10 </head>
11
12 <body>
13   <div class="display-container">
14
15   </div>
16   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/di
17 </body>
18
19 </html>
```

기본적인 레이아웃 잡고 갑시다 복습이니 스스로 해볼 것

배경 색상
#FEF5E8

```
body {  
  background-color: #0f0f0f;  
}  
  
.display-container {  
  box-shadow: 0 0 15px 0 hsla(0, 0%, 100%, .75);  
  max-width: 415px;  
  margin-left: auto;  
  margin-right: auto;  
  min-height: 100vh;  
  background-color: #fef5e8;  
}
```



GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

[illegible]

구조를 어떻게 정의할 것인가?
수많은 박스의 나열이라고 생각함.



LIKELION SKU

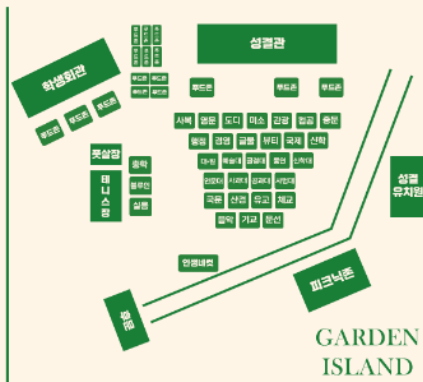
GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

GARDEN ISLAND POOL:US

📍성결관 앞 가든 아일랜드

👉 부스 지도를 클릭하시면 크게 볼 수 있어요 👈

부스지도



원하는 부스 또는 음식을 검색해 보세요!

```
<body>
  <div class="display-container">
    <!-- garden main -->
    <section>
      <div></div>
      <div></div>
      <div></div>
      <div></div>
    </section>
  </div>
  <script src="https://cdn.jsdelivr.net"
</body>
```



```
<!-- garden main -->
<section>
  <div>
    <p>GARDEN ISLAND</p>
    <p>POOL:US</p>
  </div>
  <div>
    <div>성결관 앞 가든 아일랜드</div>
    <p>부스 지도를 클릭하시면 크게 볼 수 있어요</p>
  </div>
  <div>
    <img style='width:300px;' src='./gardenImg/allbooth.png' alt="전체부스이미지">
  </div>
  <div>
    <form action="">
      <label for="boothSearch">원하는 부스 또는 음식을 검색해보세요</label>
      <input type="text" id="boothSearch">
    </form>
  </div>
</section>
</div>
```



GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

POOL:US

성결관 앞 가든 아일랜드
부스 지도를 클릭하시면 크

부스지도



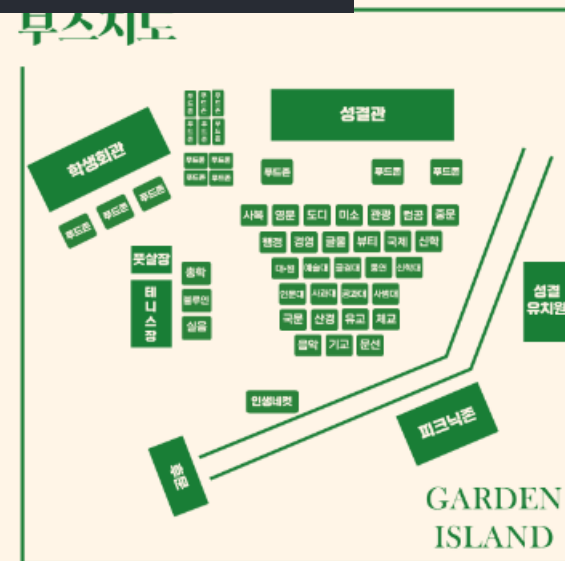
원하는 부스 또는 음식을 검색해보세요

[illegible]

POOL-115

아일랜드
크게 볼 수 있어요

부스지도



원하는 부스 또는 음식을 검색해보세요

기본적인 구조는 어느정도 잡힌 것 같다.
이제 각 요소를 디자인하자
식별가능한 클래스 이름을 주고 css를 활용하여 꾸며보자



GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

폰트 : SB 어그로

폰트 : 이사만루

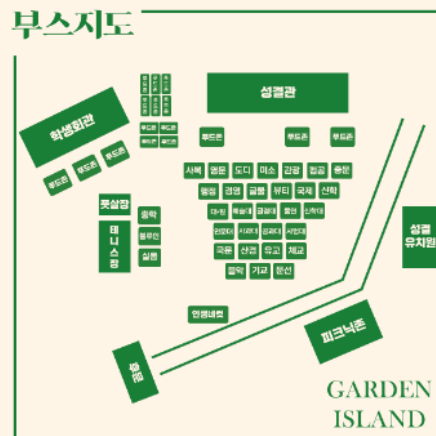
폰트 : Anton Regular

GARDEN ISLAND
POOL:US

📍 성결관 앞 가든 아일랜드

✧ 부스 지도를 클릭하시면 크게 볼 수 있어요 ✧

부스지도



원하는 부스 또는 음식을 검색해 보세요!



폰트 3가지 먼저 적용해봅시다.



dy>

```
<div class="display-container sbAggro_font">
```

```
<!-- garden main -->
```

```
<section class="boothMain d-flex flex-column text-center">
```

```
<div>
```

```
<p class="e4manro">GARDEN ISLAND</p>
```

```
<p class="Anton_font">POOL:US</p>
```

```
</div>
```

```
<div>
```

```
<div>성결관 앞 가든 아일랜드</div>
```

```
<p>부스 지도를 클릭하시면 크게 볼 수 있어요</p>
```

```
</div>
```

garden > garden.css > ...

```
1  /* 폰트 */
```

```
2
```

```
3  @import url('https://fonts.googleapis.com/css2?family=Anton&display=swap');
```

```
4  @font-face {
```

```
5      font-family: 'Anton';
```

```
        src: url('oonfonts_2108@1.1/SBAggroB.woff') format('woff');
```

```
        src: url('oonfonts_20-10@1.0/GongGothicMedium.woff') format('woff');
```

```
32
```

```
33  .Anton_font {
```

```
34      font-family: 'Anton', sans-serif;
```

```
35  }
```

```
36
```



아이콘은 bootstrap icon 사용한다.

Install

Bootstrap Icons are published to npm, but they can also be manually downloaded if needed.

Package manager

Install [Bootstrap Icons](#)—including SVGs, icon sprite, and icon fonts—with npm or Composer. Then, choose how you'd like to include the icons with the [usage instructions](#).

```
$ npm i bootstrap-icons
```



```
$ composer require twbs/bootstra
```



Download

[Releases are published on GitHub](#) and include icon SVGs, fonts, license, and readme. Our [package.json](#) is also included, though our npm scripts are primarily available for our development workflows.

[Download latest ZIP](#)

CDN

Include the icon fonts stylesheet—in your website `<head>` or via `@import` in CSS—from jsDelivr and get started in seconds. [See icon font docs](#) for examples.

```
<link rel="stylesheet" href="htt
```

Html head 안에 넣기

```
@import url("https://cdn.jsdelivriv
```



Download

Download the SVG to use or edit.

[Download SVG](#)

Icon font

Using the web font? Copy, paste, and go.

```
<i class="bi bi-geo-alt-fill" >
```

```
v>  
<div><i class="bi bi-geo-alt-fill"></i> 성결관 앞 가든 아일랜드</div>  
<p>보스 지도를 클릭하시면 크게 보 수 있어요</p>
```

 **성결관 앞 가든 아일랜드**



본격 실습 시작! () ↗

**클래스명, 폰트 크기, 둥근 테두리, 아이콘 등
자신이 알아서 조절하여 디자인 할 것
정답은 없다!!!**

[illegible]



```
body>
<div class="display-container sbAggro_font">
  <!-- garden main -->
  <section class="boothMain d-flex flex-column text-center align-items-center">
    <div class="mt-5">
      <p class="e4manro gardentitle">GARDEN ISLAND</p>
      <p class="Anton_font poolustitle">POOL:US</p>
    </div>
    <div>
      <div class="boothlocation"><i class="bi bi-geo-alt-fill"></i> 성결관 앞 가든 아일랜드</div>
      <p class="boothImgSub"><i class="bi bi-caret-down-fill"></i>부스 지도를 클릭하시면 크게 볼 수 있어요<i class="bi bi-caret-down-fill"></i></p>
    </div>
    <div>
      
    </div>
    <div>
      <form class="my-4">
        <label for="boothSearch" class="mb-2">원하는 부스 또는 음식을 검색해보세요</label>
        <p class="d-flex justify-content-center">
          <i class="bi bi-search mx-2"></i><input type="text" id="boothSearch">
        </p>
      </form>
    </div>
  </section>
</div>
```



LIKE

```
body {
  background-color: #0f0f0f;
  color: #287e3a;
}

.gardentitle {
  -webkit-text-stroke: 1px #287e3a;
  color: transparent;
  font-size: 30px;
  font-weight: 900;
  margin-bottom: -30px;
}

.poolustitle {
  color: #287e3a;
  font-size: 100px;
}

.boothlocation {
  background-color: hsla(0, 0%, 100%, .6);
  border-radius: 15px;
  font-size: 18px;
  margin: 15px auto;
  padding: 10px;
  position: relative;
  width: 250px;
}

.boothImgSub {
  font-size: 14px;
  font-weight: 700;
}
```

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

```
form label {
  font-size: 15px;
}

form p {
  border: 2px solid #287e3a;
}

form p input {
  background: none;
  border: none;
}

.bi-search::before {
  margin-top: 5px;
}
```




GARDEN ISLAND POOL:US

✧ 부스 지도를 클릭하시면 크게 볼 수 있어요 ✧



GARDEN MAIN

완성! (*◡*)



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.



블루인 부스

떡볶이
아이스크림
약과 아이스크림



대의원회 부스

화채
미숫가루
냉커피



사회과학대학 부스

왕 소시지
봉봉 에이드



인문대학 부스

소프트 아이스크림
레인보우 아이스크림
하쉬 초코링 아이스크림



IT 공과대학 부스

라면
치즈라면
핫바

GARDEN BOOTH

HTML, CSS, JAVASCRIPT



JS를 제대로 배우지 않았기 때문에 어려울 수 있어요.

**너무 완벽하게 이해하려 애쓰지말고
일단은 ‘아~ 이걸 이렇게 쓰이는 거구나!’라고 생각하며
가볍게 따라와주세요 화이팅**

</section>

```
▶ <button type="button" class="btn"> ... </button> =
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
▶ <button type="button" class="btn"> ... </button>
```



블루인 부스

떡볶이
아이스크림
약과 아이스크림



대의원회 부스

화채
미숫가루
냉커피



사회과학대학 부스

왕 소시지
봉봉 에이드



인문대학 부스

소프트 아이스크림
레인보우 아이스크림
하쉬 초코링 아이스크림



IT 공과대학 부스

라면
치즈라면
핫바

... ????
이걸 다?



```
</section>
```

```
<button type="button" class="btn">... </button> =
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

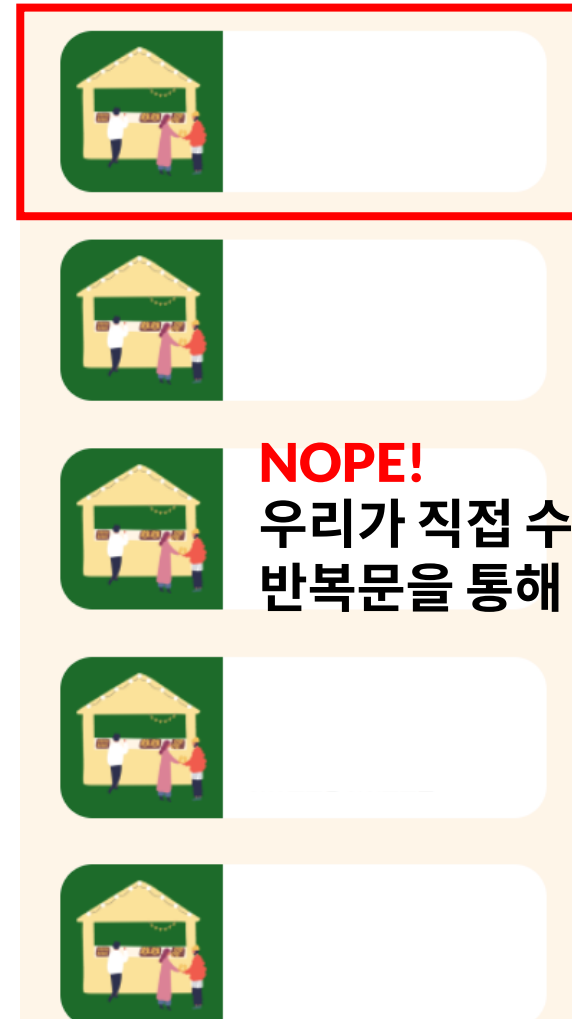
```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```

```
> <button type="button" class="btn">... </button>
```



NOPE!

우리가 직접 수작업을 하는 것이 아닌
반복문을 통해 html 요소를 반복하게 할 것이다.



블루인 부스
떡볶이
아이스크림
약과 아이스크림



대의원회 부스
화채
마숫가루
냉커피



사회과학대학 부스
왕 소시지
봉봉 에이드



인문대학 부스
소프트 아이스크림
레인보우 아이스크림
하쉬 초코링 아이스크림



IT 공과대학 부스
라면
치즈라면
핫바

음 .. ○ ○ 그래!
근데 그 안의 데이터들은 어떻게 처리할건데??
버튼은 동일하지만
그 안에 들어가야 할 데이터들은 모두 다르잖아!

그 안의 데이터들은 ?
JSON에서 데이터를 불러와 사용할 것!!

L O T....JSON이 뭐죠?

JavaScript Object Notation라는 의미의 축약어
데이터를 저장, 전송할 때 많이 사용되는 Data 교환 형식
Ex) 서버에서 클라이언트로 데이터를 전송하여 표현할 때 사용

단순히 데이터를 표시하는 표현 방법. 이라고 생각하자

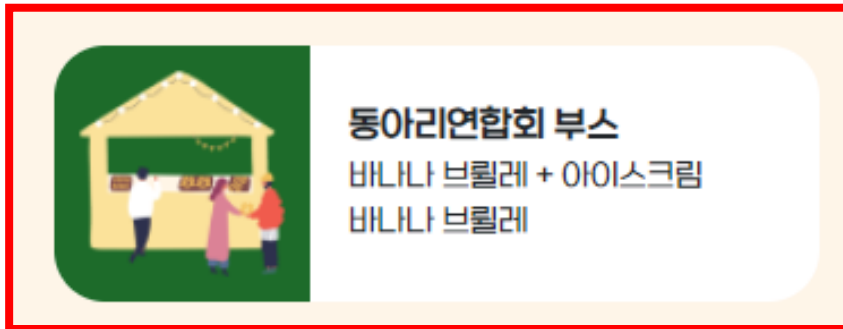
```
> { Menuison >
[
  {
    "id": 1,
    "department": "동아리연합회 부스",
    "menus": [
      { "name": "바나나 브릴레 + 아이스크림", "price": "2,000원" },
      { "name": "바나나 브릴레", "price": "1,500원" }
    ]
  },
  {
    "id": 2,
    "department": "블루인 부스",
    "menus": [
      { "name": "떡볶이", "price": "2,000원" },
      { "name": "약과 아이스크림", "price": "3,000원" },
      { "name": "떡볶이 + 약과 아이스크림", "price": "5,000원" }
    ]
  },
  {
    "id": 3,
    "department": "대의원회 부스",
    "menus": [
      { "name": "화채", "price": "3,500원" },
      { "name": "미숫가루", "price": "1,500원" },
      { "name": "냉커피", "price": "1,000원" },
      { "name": "캔 음료 S", "price": "1,000원" },
      { "name": "캔 음료 M", "price": "1,500원" },
      { "name": "캔 음료 L", "price": "2,000원" }
    ]
  }
]
```



그래 그럼 우리가 해야할 것..

- 1.반복문을 통해 HTML 요소를 반복**
- 2.JSON에서 부스의 정보(데이터)를 받아올 것!**

방금 만든 GARDEN MAIN 부분
헷갈리니까 잠깐 접어놓자~



```
<!-- garden main -->  
<section class="boothMain d-flex flex-column">  
</section>
```

```
<!-- booth section -->  
<section>  
  <button>  
    <div>  
      <div><img src="" alt=""></div>  
      <div>  
        <p></p>  
        <p></p>  
      </div>  
    </div>  
  </button>  
</section>
```



```
<!-- booth section -->
<section>
  <button>
    <div>
      <div></div>
      <div>
        <p>동아리연합회 부스</p>
        <p>
          <span>바나나 브릴레 + 아이스크림</span><br/>
          <span>바나나 브릴레</span>
        </p>
      </div>
    </div>
  </button>
</section>
```

일단 임시로 데이터를 넣고
디자인만 해보자



동아리연합회 부스

바나나 브릴레 + 아이스크림

바나나 브릴레



LIKELION SKU



GROWL TO GROWTH LIKELION SUNGKYUL UNIV.

```
<!-- booth section -->
<section class="text-center">
  <button>
    <div class="booth d-flex">
      <div class="boothimg"></div>
      <div class="boothinfo d-flex flex-column justify-content-center text-start">
        <p class="boothtitle mb-1">동아리연합회 부스</p>
        <p class="boothsub mb-0">
          <span>바나나 브릴레 + 아이스크림</span><br/>
          <span>바나나 브릴레</span>
        </p>
      </div>
    </div>
  </button>
</section>
```

```
button {
  background-color: transparent;
  border: none;
  width: 300px;
  margin-bottom: 20px;
}

.booth {
  background-color: white;
  border-radius: 15px;
}

.boothimg img {
  border-radius: 15px 0 0 15px;
  margin-right: 10px;
}

.boothinfo {
  font-size: 14px;
  padding: 10px;
}

.boothtitle {
  font-size: 14px;
}

.boothsub {
  font-size: 12px;
}
```

- 1.반복문을 통해 HTML 요소를 반복**
- 2.JSON에서 부스의 정보(데이터)를 받아올 것!**



1. 반복문을 통해 HTML 요소를 반복

반복할 HTML 요소가 어디인지 알아야 한다.
JS로 원하는 HTML 요소를 가져오기!

```
const template = `

![부스이미지](./gardenImg/booth.png)

`;
```

```
<!-- booth section -->  
<section class="boothSection text-center">  
  <!--  
    <button>  
      <div class="booth d-flex">  
        <div class="boothimg"></div>  
        <div class="boothinfo d-flex flex-column justify-content-between">  
          <p class="boothtitle mb-1"></p>  
          <p class="boothsub mb-0"></p>  
        </div>  
      </div>  
    </button> -->  
</section>
```

1. HTML 파일에 방금 디자인한 button 요소 전체를 잘라낸다.



1. 반복문을 통해 HTML 요소를 반복

반복할 HTML 요소가 어디인지 알아야 한다.
JS로 원하는 HTML 요소를 가져오기!

```
const template = `

![부스이미지](./gardenImg/booth.png)


```

\$(".boothSection").append(template);

2. JS파일의 template 변수에 할당! (백틱 사용)

section text-center">

oth d-flex">

= "boothimg"></p>

<p class="boothsub mb-0"></p>

</div>

</div>

</button> -->

</section>



1. 반복문을 통해 HTML 요소를 반복

반복할 HTML 요소가 어디인지 알아야 한다.
JS로 원하는 HTML 요소를 가져오기!

```
const template = `



`;
```

`$(".boothSection").append(template);`

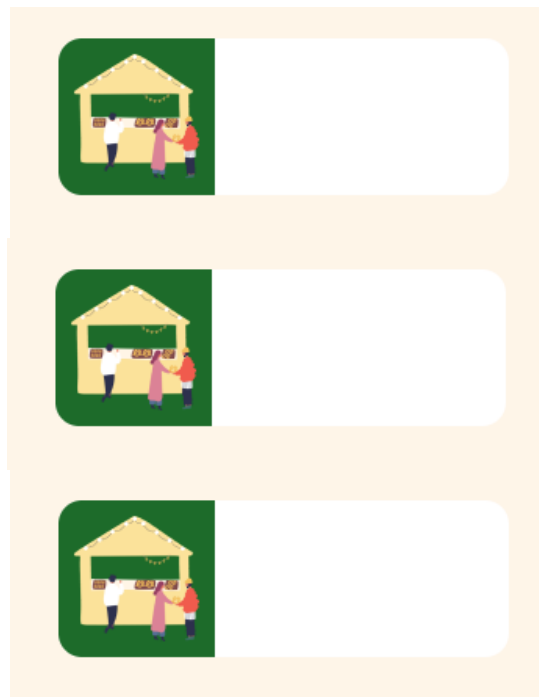
일단 결과만 보면 이전 형태랑 똑같다
하지만 해당 코드를 JS에서 사용하기 위해 작업해준 것

boothSection이라는 클래스를 가진 요소에
Template 변수에 저장된 HTML 코드를 추가하는 역할을 한다.
(jQuery 문법)

이 부분

```
!-- booth section -->  
<section class="boothSection text-center">  
  <!--  
    <button>  
      <div class="booth d-flex">  
        <div class="boothimg">  
        <div class="boothinfo d-flex flex-column justify-content-between">  
          <p class="boothtitle mb-1"></p>  
          <p class="boothsub mb-0"></p>  
        </div>  
      </div>  
    </button> -->  
</section>
```

먼저 **반복문**으로 부스 버튼 3개만 반복해보자





반복문 ForEach 연습

```
garde > for each js > myFunction
1  const students = [{
2      "name": "김현준",
3      "department": "산업경영공학과",
4  },
5  {
6      "name": "노승희",
7      "department": "미디어소프트웨어학과",
8  },
9  ]
10
11  students.forEach(myFunction);
12
13  function myFunction(student) : void {
14      console.log(student.name);
15      console.log(student.department);
16      console.log('-----')
17  }
```

1. Students 학생 정보가 들어있음.
2. 반복문 메서드 forEach를 사용하여 students 배열을 순회
3. 순회하면서 실행할 작업

```
students.forEach(
    function myFunction(student) : void {
        console.log(student.name);
        console.log(student.department);
        console.log('-----')
    }
);
```

```
[Running] node "c:\Us
김현준
산업경영공학과
-----
노승희
미디어소프트웨어학과
-----|
```



```
const items = [{
  "id": 1,
  "department": "동아리연합회 부스",
  "menus": [
    { "name": "바나나 브릴레 + 아이스크림", "price": "2,000원" },
    { "name": "바나나 브릴레", "price": "1,500원" }
  ]
},
{
  "id": 2,
  "department": "블루인 부스",
  "menus": [
    { "name": "떡볶이", "price": "2,000원" },
    { "name": "약과 아이스크림", "price": "3,000원" },
    { "name": "떡볶이 + 약과 아이스크림", "price": "5,000원" }
  ]
},
{
  "id": 3,
  "department": "대의원회 부스",
  "menus": [
    { "name": "화채", "price": "3,500원" },
    { "name": "미숫가루", "price": "1,500원" },
    { "name": "냉커피", "price": "1,000원" },
    { "name": "캔 음료 S", "price": "1,000원" },
    { "name": "캔 음료 M", "price": "1,500원" },
    { "name": "캔 음료 L", "price": "2,000원" },
  ]
}
]
```

학생 정보가 담긴 **students** 대신에
부스 정보가 담긴 **items** 를 사용

```
students.forEach(
```

```
function myFunction(student) : void {
  console.log(student.name);
  console.log(student.department);
  console.log('-----')
}
```

```
); 부스 버튼을 반복
```



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

```
items.forEach(function(item) : void {  
    const template = `  
        <button>  
            <div class="booth d-flex">  
                <div class="boothimg">  
                <div class="boothinfo d-flex flex-column justify-content-center text-center">  
                    <p class="boothtitle mb-1"></p>  
                    <p class="boothsub mb-0"></p>  
                </div>  
            </div>  
        </button>  
    `;  
    $(".boothSection").append(template);  
})
```





```
items.forEach(function(item) : void {  
  const template = `  
    <button>  
      <div class="booth d-flex">  
        <div class="boothimg"></div>  
        <div class="boothinfo d-flex flex-column justify-content-center text-start">  
          <p class="boothtitle mb-1"></p>  
          <p class="boothsub mb-0"></p>  
        </div>  
      </div>  
    </button>  
  `;  
  $(".boothSection").append(template);  
})
```

부스 정보가 들어가야 할 부분



템플릿 리터럴을 사용한 템플릿 언어

백틱(`)으로 둘러싸인 문자열 안에서 `${}` 형식으로 JS 표현식 삽입
이를 통해 동적으로 HTML 코드 생성 가능

```
items.forEach(function(item) : void {  
  const template = `  
    <button>  
      <div class="booth d-flex">  
        <div class="boothimg"></div>  
        <div class="boothinfo d-flex flex-column justify-content-center text-start">  
          <p class="boothtitle mb-1">${item.department}</p>  
          <p class="boothsub mb-0"></p>  
        </div>  
      </div>  
    </button>  
  `;  
  $(".boothSection").append(template);  
})
```



LIKELION SKU

GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

```
    "id": 1,  
    "department": "동아리연합회 부스",  
    "menus": [  
      { "name": "바나나 브릴레 + 아이스크림", "price": "2,000원" },  
      { "name": "바나나 브릴레", "price": "1,500원" }  
    ]  
  }  
]
```

```
<div class="boothinfo d-flex flex-column justify-content-center text-start">  
  <p class="boothtitle mb-1">${item.department}</p>  
  <p class="boothsub mb-0">${data.menus.map(menu => `<span>${menu.name}</span><br/>`).join('')}</p>  
</div>
```



동아리연합회 부스

바나나 브릴레 + 아이스크림
바나나 브릴레



블루인 부스

떡볶이
약과 아이스크림
떡볶이 + 약과 아이스크림



대의원회 부스

화채
미숫가루
냉커피
캔 음료 S
캔 음료 M
캔 음료 L



오로지 text만 바뀌면 됨

```
<div class="boothimg"><img src= "../gardenimg/booth.png" alt= "부스이미지"></div>
<div class="boothinfo d-flex flex-column justify-content-center text-start">
  <p class="boothtitle mb-1">동아리연합회 부스</p>
  <p class="boothsub mb-0">
    <span>바나나 브쥬레 + 아이스크림</span><br/>
    <span>바나나 브쥬레</span>
    <span>호떡</span>
    <span>초코바나나</span>
  </p>
</div>
```

메뉴의 개수가 다르기 때문에
Html요소인 span 태그를 같이 생성해주기



Foreach 구문으로 HTML요소들을 반복하고, 임시 데이터까지 넣어줌!



동아리연합회 부스

바나나 브릴레 + 아이스크림
바나나 브릴레



블루인 부스

떡볶이
약과 아이스크림
떡볶이 + 약과 아이스크림



대의원회 부스

화채
미숫가루
냉커피
캔 음료 S
캔 음료 M
캔 음료 L

```
items.forEach(function(item) void {  
    //const menuToShow = data.menus.slice(0, 2);  
  
    const template = `  
        <button>  
            <div class="booth d-flex">  
                <div class="boothimg"></div>  
                <div class="boothinfo d-flex flex-column justify-content-center text-start">  
                    <p class="boothtitle mb-1">${item.department}</p>  
                    <p class="boothsub mb-0">${item.menus.map(menu => `<span>${menu.name}</span><br/>`).join('')}</p>  
                </div>  
            </div>  
        </button>  
    `;  
    $(".boothSection").append(template);  
    }  
})
```




미리보기에 모든 정보를 보여줄 필요 없다. 2개씩만 표시되도록 하자!

```
items.forEach(function(data, i) : void {  
    const menusToShow = data.menus.slice(0, 2);  
  
    const template = `  
        <button>  
            <div class="booth d-flex">  
                <div class="boothimg"></div>  
                <div class="boothinfo d-flex flex-column justify-content-center text-start">  
                    <p class="boothtitle mb-1">${data.department}</p>  
                    <p class="boothsub mb-0">${menusToShow.map(menu => `<span>${menu.name}</span>`)}</p>  
                </div>  
            </div>  
        </button>  
    `;  
    $(".boothSection").append(template);  
});
```



동아리연합회 부스
바나나 브릴레 + 아이스크림
바나나 브릴레



블루인 부스
떡볶이
약과 아이스크림



대의원회 부스
화재
미숫가루

- 1.반복문을 통해 HTML 요소를 반복 → 성공!**
- 2.JSON에서 부스의 정보(데이터)를 받아올 것!**

JSON에서 부스의 정보(데이터)를 받아오자!

임시 데이터가 아닌 Menu.json 파일에서 부스 정보들을 받아오고 싶다.



```
fetch('Menu.json')
  .then(function(response) : Promise<any> {
    if (!response.ok) {
      throw new Error('에러에러에! =');
    }
    return response.json();
  })
  .then(function(items) : void {
    items.forEach(function(data, i) : void {
      const menusToShow = data.menus.slice(0, 2);

      const template = `
        <button>
          <div class="booth d-flex">
            <div class="boothimg"></div>
            <div class="boothinfo d-flex flex-column justify-content-center text-start">
              <p class="boothtitle mb-1">${data.department}</p>
              <p class="boothsub mb-0">${menusToShow.map(menu => `<span>${menu.name}</span><br/>`).join('')}</p>
            </div>
          </div>
        </button>
      `;

      $(".boothSection").append(template);
    })
  })
  .catch(function(error) : void {
    console.log(error);
  });
```

Fetch 함수

서버로 데이터를 요청하고, 응답을 받아오는 함수

.then() :

서버로부터 응답을 받으면 .then() 메서드로 응답 처리

```
fetch('Menu.json')
  .then(function(response) : Promise<any> {
    if (!response.ok) {
      throw new Error('에러에러에러');
    }
    return response.json();
  })
  .then(function(items) : void {
    items.forEach(function(data) : void { ...
    })
  })
  .catch(function(error) : void {
    console.log(error);
  }));
```

Menu.json로 데이터를 달라는 요청 보냄

Response가 응답을 받음!

응답이 성공적이지 않을 경우, 에러를 던짐 (state : rejected)

응답을 잘 받아왔다면 응답 데이터 반환 (state : fulfilled) (다음 .then 실행)

Response.json() :

JSON 형식 데이터 -> JS 객체로 변환

Response.stringify() :

JS 객체 -> JSON 형식 데이터

받아온 응답 데이터로 반복문 실행!!!

.catch

위의 .then() 중 하나라도 에러가 발생했을 때 호출



Fetch 함수

서버로 데이터를 요청하고, 응답을 받아오는 함수

.then() :

서버로부터 응답을 받으면 .then() 메서드로 응답 처리

```
fetch('Menu.json')
  .then(function(response) : Promise<any> {
    if (!response.ok) {
      throw new Error('에러에러에러');
    }
    return response.json();
  })
  .then(function(items) : void {
    items.forEach(function(data) : void { ...
    })
  })
  .catch(function(error) : void {
    console.log(error);
  });
```

→ **Response**가 응답을 받음!

→ 응답이 성공적이지 않을 경우, 에러를 던짐 (**state : rejected**)

→ 응답을 잘 받아왔다면 응답 데이터 반환 (**state : fulfilled**)

Response : Promise 객체 형태

Promise 객체

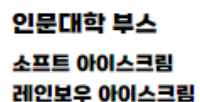
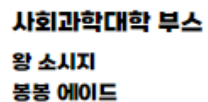
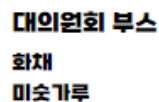
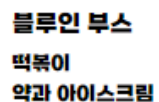
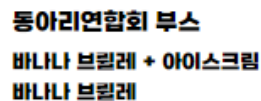
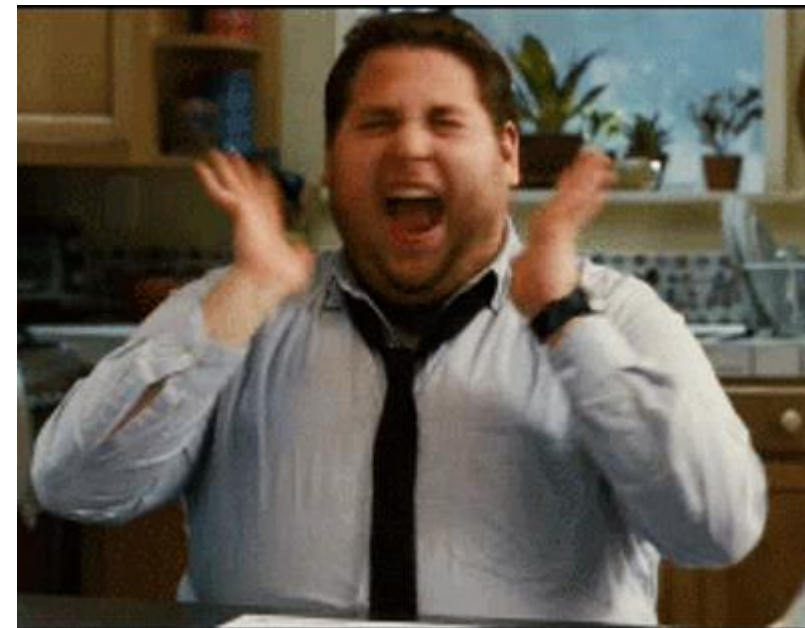
State : fulfilled, rejected

Result : 데이터(부스 정보)

- 1. 반복문을 통해 HTML 요소를 반복 → 성공!**
- 2. JSON에서 부스의 정보(데이터)를 받아올 것! → 성공!!**



GROWL TO GROWTH
LIKELION SUNGKYUL UNIV.

[illegible]

**짧은 시간에 많은 것을 배웠습니다..
꼭 복습해서 자신의 것으로 만드세요!!**
★ JS 부분 혼자서 꼭 복습해보기 ★