



Fabcar Application Front-end

아주대학교 Distributed&Mobile Computing Lab

kimgahyun1@ajou.ac.kr

HTML? CSS? JS?

HTML

Hypertext Markup Language

웹의 구조

CSS

Cascading Style Sheet

웹의 디자인

JS

Javascript 프로그래밍 언어

웹의 동작

HTML 기본 구조와 태그

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>제목</title>
5      <meta charset="utf-8" />
6    </head>
7    <body>
8      <h1>Hello</h1>
9      <div>
10       <p>프론트엔드 스터디를 해봅시다!</p>
11       
12     </div>
13   </body>
14 </html>
```

Combo
Max 1623

A pixelated green avatar, resembling a character from a retro game, positioned next to the Combo and Max scores.

HTML 기본 구조와 태그

태그의 형식

<여는 태그 속성명="속성값"> 내용 </닫는 태그>

열고 닫는 순서를 지켜줘야 함

<div> <p> 좋은 예 </p> </div>


<div> <p> 에러 </div> </p>

HTML 기본 구조와 태그

HTML 5 버전을 사용한다고 브라우저에게 알려줌

```
1 <!DOCTYPE html>
2 <html> 전체 문서를 감싸는 태그
3   <head> 문서의 정보를 나타내는 태그, 기타 라이브러리(CDN 등)를 포함시키는 곳. 예: 부트스트랩, 제이쿼리, 뷰 등
4     <title>제목</title> 문서 제목
5     <meta charset="utf-8" />
6   </head> 한글 인코딩을 요청하는 부분, 이거 없으면 한글 깨짐
7   <body> 실제로 웹상에 보여지는 부분
8     <h1>Hello</h1> Header의 줄임말, 글을 띄워줌. H 뒤의 숫자가 커질수록 글자는 작아짐 (h1>h2>...>h5)
9     <div> Division의 줄임말, 구역을 표시하는 태그. P와 Img를 한 구역으로 묶어줌
10    <p>프론트엔드 스터디를 해봅시다!</p> Paragraph, 문단 표시
11    
12  </div> 이미지 태그, 소스 뒤에 주소(상대경로 or 절대경로)를 넣어야 함, 닫는 태그 없음
13 </body>
14 </html>
```

Combo
Max 1623



Paragraph, line break, horizontal rule, headings

`<p>` 문단입니다. p 태그 안에서는 엔터를 쳐도

줄이 바뀌지 않아서 줄바꿈 태그(br) 를 써야 합니다`</p>`

`
` 한 줄 띄기

`<hr>` 구분선 긋기

`<h1>` 큰 제목 `</h1>`

`<h2>` 두번째로 큰 제목 `</h2>`

`<h3>` 세번째로 큰 제목 `</h3>`

`<h4>` 네번째로 큰 제목 `</h4>`

`<h4>` 다섯번째로 큰 제목 `</h4>`

anchor 태그, 리스트

`` 링크를 걸고 싶을때 `<a>`

``

` 첫번째 리스트 `

` 두번째 리스트 `

` 세번째 리스트 `

``

폼(Form) 태그

- Static Web(정적 웹), Dynamic Web (동적 웹)
- 폼 태그는 동적 웹에서 사용됨
- 예를 들어 로그인, 회원가입을 할 때
- 또는 버튼을 누르면 어떤 반응이 돌아올 때

폼(Form) 태그

기본형식

```
<form action="/api/join" method="post"></form>
```

폼에 담긴 정보를 보낼 url주소를 적는 곳

폼에 담긴 정보를 어떻게 보낼 것인가?

크게 post방식, get방식이 있음

Get은 주소를 통해서 query를 보내는 방식

Post는 req.body에 담겨서 보냄

쉽게 말하면 get은 주소에 보이고, post는 주소에 안보여서
좀더 보안이 필요할때 쓰임(로그인, 회원가입 등)

input 태그

Form 안에 어떤 정보를 받을 것인지 input을 통해 받는다

```
<input type="text" placeholder="아이디 입력" id="userid" name="userid" />  
<input type="password" id="password" name="password" />  
<input type="radio" name="gender" value="m" />남  
<input type="radio" name="gender" value="f" />여  
<input type="checkbox" name="subs" />구독?
```

Type: 어떤 형식의 정보를 받을 것인가

Value: input 안에 담긴 정보 값

Id: 나중에 배울 label과 연결할 값

Name: form안에 담긴 데이터를 보냈을 때 서버에서 받을 이름

input 태그

특수한 Input 태그들

```
<select name="area">
  <option value="seoul">서울</option>
  <option value="gyeonggi">경기</option>
  <option value="jeolla">전라</option>
  <option value="gyeongsang">경상</option>
</select>
<textarea placeholder="긴 글 입력" name="textarea"></textarea>
```

select: 여러 옵션 중 선택 가능

Textarea: 글이 길 때 한번에 많은 글 입력 받을 수 있는 태그

버튼 태그

버튼의 속성은 type이 있다

```
<button>클릭</button>  
<button type="submit">제출</button>  
<button type="reset">취소</button>
```

submit: 폼을 제출하는 버튼

Reset: 폼 초기화

CSS & Bootstrap



Cascading Style Sheet

마크업 언어가 실제로 표시되는 방법을 표현해주는 언어



Bootstrap

Bootstrap 4.x

CSS들의 모임. 자주 쓰이는 컴포넌트들을 모듈화 해놓아서
빠르고 편리하게 웹 스타일링을 할 수 있음

CSS & Bootstrap

Atom editor

```
<div class="container">
  container
</div>
```

Bootstrap에서
정의해 둔 Css 모음

Chrome dev tool

```
.container { bootstrap.min.css:5
  padding-right: 15px;
  padding-left: 15px;
  margin-right: auto;
  margin-left: auto;
}
```

CSS

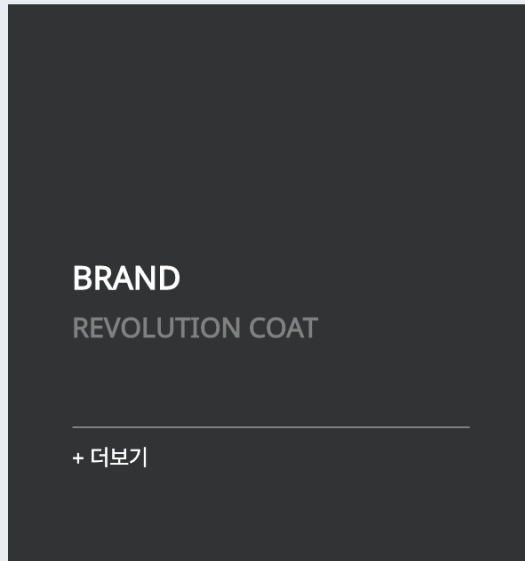
```
<div style="padding-right: 15px; padding-left: 15px;
        margin-right: auto; margin-left: auto;">
    container
</div>
```

모듈화



```
<div class="container">
    container
</div>
```

모듈화가 필요한 이유



```
<div style="background-color: white;
            height: 350px;
            padding-top: 150px;
            padding-left: 40px;
            padding-right: 40px;
            cursor: pointer;">
  <div style="font-size: 20px;
            font-weight: 900;
            margin-bottom: 0px;">
    CAREER
  </div>
  <div style="font-size: 17px;
            font-weight: 700;
            color: gray;
            text-align: left;">
    REVOLUTION COAT
  </div>
  <hr style="margin-top: 50px; color: gray !important; width: 100%;
            border: 1px solid gray; margin-bottom: 9px;">
  <span style="float: left; color: gray; "> + 더보기</span>
</div>
```


모듈화가 필요한 이유

BRAND
REVOLUTION COAT

+ 더보기

```
1 <div style="background-color: white;
2   height: 350px;
3   padding-top: 150px;
4   padding-left: 40px;
5   padding-right: 40px;
6   cursor: pointer;">
7   <div style="font-size: 20px;
8     font-weight: 900;
9     margin-bottom: 0px;">
10     CAREER
11   </div>
12   <div style="font-size: 17px;
13     font-weight: 700;
14     color: gray;
15     text-align: left;">
16     REVOLUTION COAT
17   </div>
18   <hr style="margin-top: 50px; color: gray !important; width: 100%;
19     border: 1px solid gray; margin-bottom: 9px;">
20   <span style="float: left; color: gray; "> + 더보기</span>
21 </div>
22
23
24 <div style="background-color: white;
25   height: 350px;
26   padding-top: 150px;
27   padding-left: 40px;
28   padding-right: 40px;
29   cursor: pointer;">
30   <div style="font-size: 20px;
31     font-weight: 900;
32     margin-bottom: 0px;">
33     CAREER
34   </div>
35   <div style="font-size: 17px;
36     font-weight: 700;
37     color: gray;
38     text-align: left;">
39     REVOLUTION COAT
40   </div>
41   <hr style="margin-top: 50px; color: gray !important; width: 100%;
42     border: 1px solid gray; margin-bottom: 9px;">
43   <span style="float: left; color: gray; "> + 더보기</span>
44 </div>
45
46
47
48 <div style="background-color: white;
49   height: 350px;
50   padding-top: 150px;
51   padding-left: 40px;
52   padding-right: 40px;
53   cursor: pointer;">
54   <div style="font-size: 20px;
55     font-weight: 900;
56     margin-bottom: 0px;">
57     CAREER
58   </div>
59   <div style="font-size: 17px;
60     font-weight: 700;
61     color: gray;
62     text-align: left;">
63     REVOLUTION COAT
64   </div>
65   <hr style="margin-top: 50px; color: gray !important; width: 100%;
66     border: 1px solid gray; margin-bottom: 9px;">
67   <span style="float: left; color: gray; "> + 더보기</span>
68 </div>
69
```

CAREER
REVOLUTION COAT

+ 더보기

모듈화가 필요한 이유

```
.main-container{  
  background-color: white;  
  height: 350px;  
  padding-top: 150px;  
  padding-left: 40px;  
  padding-right: 40px;  
  cursor: pointer;  
}
```

```
.main-container .title{  
  font-size: 20px;  
  font-weight: 900;  
  margin-bottom: 0px;  
}
```

```
.main-container .subtitle{  
  font-size: 17px;  
  font-weight: 700;  
  color: gray;  
  text-align: left;  
}
```

```
<div class="main-container">  
  <div class="title">INTERVIEW</div>  
  <div class="subtitle">클린씨 상생 스토리</div>  
  <span class="more"> + 더보기</span>  
</div>
```

재사용할 것 같은 component의
css 모음들을 모듈화 해 놓고

적용시키면 훨씬 간결한 코드로 구현 가능

CSS를 적용하는 방법 3가지

1. 요소 안에 직접 작성해서 적용 (inline-styling)
2. 한 html 문서 안에 style 태그로 감싸기
3. 프로젝트 안의 한 곳에 다 몰아넣고 참조해서 사용하기

(1) 요소 안에 직접 적용

```
<div style="color: red; font-size: 14px;">  
    요소 안에 직접 적용  
</div>
```

태그 안에 style 속성을 넣어서 직접 적용한다

이때 css코드는 세미콜론(;)으로 구분한다

한 요소에만 스타일링을 주고 싶을 때 이 방법을 사용

전체 코드에 이렇게 스타일링을 했을 때 코드가 지저분해 보이고 가독성이 떨어지므로 권장하지 않음.

되도록이면 모듈화를 하자.

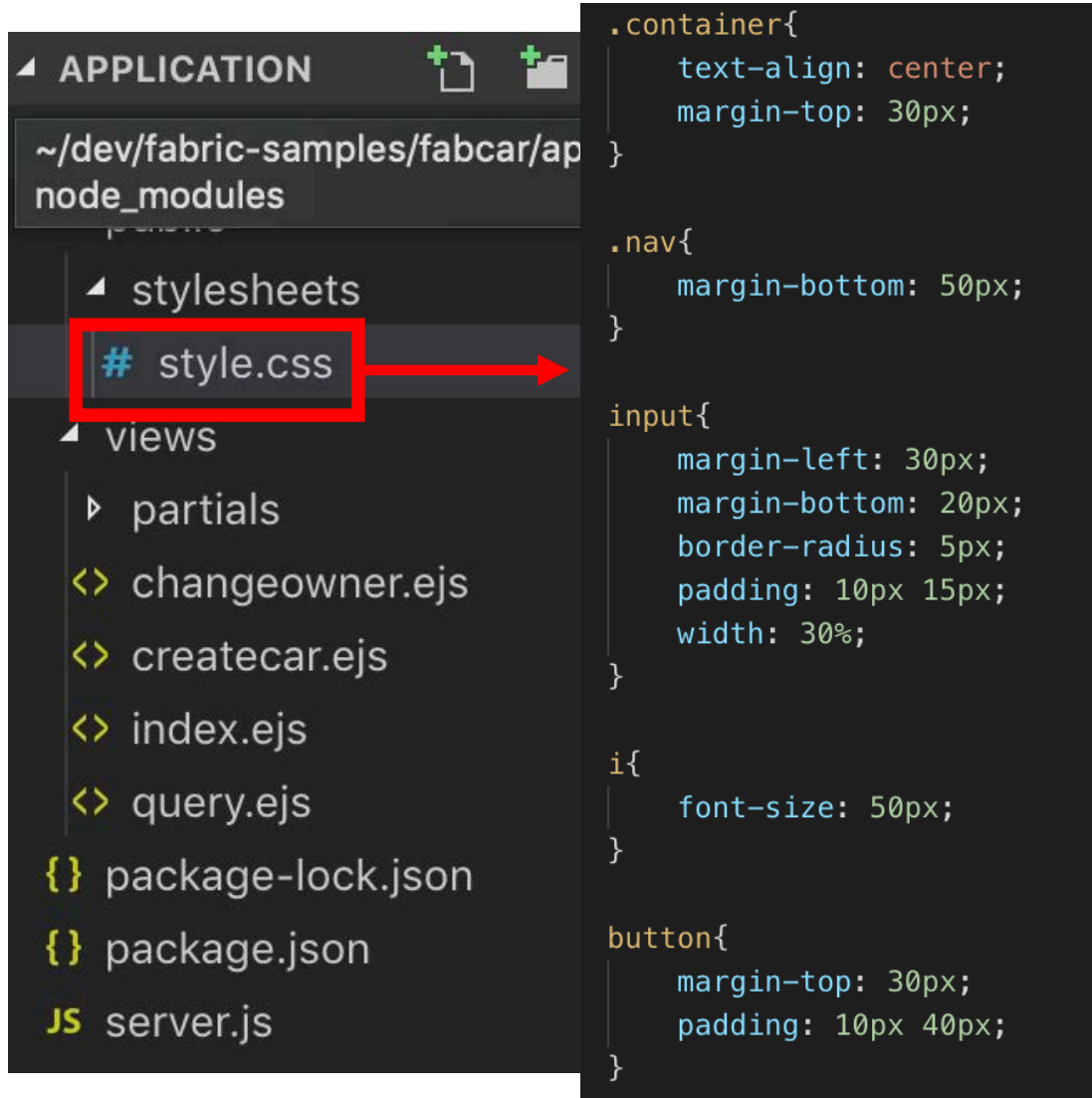
(2) 한 문서 안에 style 태그로 감싸기

```
<!DOCTYPE html>
<html>
  <head>
    <title>제목</title>
  </head>
  <body>
    <div class="my-style">
      한 문서 안에서 적용
    </div>
  </body>
</html>

<style>
  .my-style{
    color: red;
    font-size: 14px;
  }
</style>
```

Html 문서 안에 <style> </style>태그로 감싸서 모듈화하기
이 또한 다른 페이지에서 해당 css를 사용할 때 어디 있는지
찾기 어렵고, 가독성이 떨어지므로 권장하지 않음

(3) 한 곳에 몰아놓고 참조(reference)해서 사용



프로젝트 디렉토리 안에 style.css를 만들고
이 css를 사용할 페이지의 head부분에서 참조하여 사용
가장 모듈화에 가깝고 전체 코드가 깔끔해지므로 권장하는 방법

외부 CSS 모듈을 참조해보자

- CDN이란? Contents Delivery Network
- 외부 모듈의 CDN을 내 애플리케이션의 `<head>` `</head>` 사이에 포함시키면 편하게 사용 가능
- 애플리케이션 디렉토리 내에 다운로드해서 파일을 가지고 있지 않아도 됨
 - 외부 모듈을 가볍게 가져다 쓸 수 있다는 장점
- Fabcar 예제에서는 bootstrap 4.0과 font-awesome 두 가지 외부 모듈을 참조

부트스트랩

<http://bootstrapk.com/getting-started/>

부트스트랩 CDN

MaxCDN 의 사람들은 고맙게도 부트스트랩의 CSS 와 자바스크립트를 CDN 으로 지원합니다. 이를 사용하려면 아래의 **부트스트랩 CDN** 링크들을 사용하세요.

```
<!-- 합쳐지고 최소화된 최신 CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">

<!-- 추가적인 테마 -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap-
theme.min.css">

<!-- 합쳐지고 최소화된 최신 자바스크립트 -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
```

Copy

이 부분을 복사해 <head> </head>사이에 넣는다

<http://bootstrapk.com/components/>

부트스트랩에 정의된 component들의 목록과
live demo가 가능한 예제를 볼 수 있음

드롭다운 사용하기

약간의 HTML 과 [드롭다운 자바스크립트 플러그인](#) 로 드롭다운 메뉴를 추가하세요.

탭형에서의 드롭다운

EXAMPLE

Home Help Dropdown ▾

```
<ul class="nav nav-tabs">
  ...
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-
expanded="false">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu" role="menu">
      ...
    </ul>
  </li>
  ...
</ul>
```

Copy

Fabcar 에서 사용할 부트스트랩 컴포넌트 (1) 버튼



```
<button class="btn btn-primary"> primary </button>
<button class="btn btn-secondary"> secondary </button>
<button class="btn btn-success"> success </button>

<button class="btn btn-?????"> 원하는 색깔의 버튼 클래스를 적용 </button>
```



아이콘 (Glyphicons)

드롭다운

버튼 그룹

버튼 드롭다운

입력 그룹

네비게이션

탭형

알약형

양쪽정렬된

비활성화된 링크

드롭다운 사용하기

네비게이션 바

Fabcar 에서 사용할 부트스트랩 컴포넌트 (2) 네비게이션

탭형

`.nav-tabs` 클래스는 `.nav` 클래스가 필요한 것을 기억하세요.

EXAMPLE

Home

Profile

Messages

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Copy

Font-awesome

- 각종 아이콘들을 이미지가 아닌 font의 형식으로 코드만 포함하면 사용할 수 있게 만들어놓은 라이브러리
- 장점: 이미지 파일이 아니기 때문에 가벼운 용량으로 아이콘을 사용할 수 있음
- <https://fontawesome.com/start>

Get Started for Free

CDN powered Kits. 1 line of code. Live in under 1 minute.


Enter your email to create your first kit!

[Already have an account?](#)

Send Kit Code

</>

[Why do you need my email address?](#)

Powered by  StackPath

Font-awesome

Welcome to your new Font Awesome Account + Kit!

Here's that new kit you requested. It's ready to use right now! Follow the steps below to get going. You can [change its settings](#) whenever you'd like.

← Your Kits

 e7a4853152

이 부분을 복사해 <head> </head> 사이에 넣는다

 How to Use

 Settings

```
<script src="https://kit.fontawesome.com/e7a4853152.js"></script>
```

 Copy Kit Code

1

Add Your Kit's Code to Your Project

Copy your kit's code into the `<head>` of each template or page in your project where you want to use Font Awesome. Not sure how to do that or need more guidance? We've got an example file to show you exactly where to reference your kit.

 [Download Example File](#)

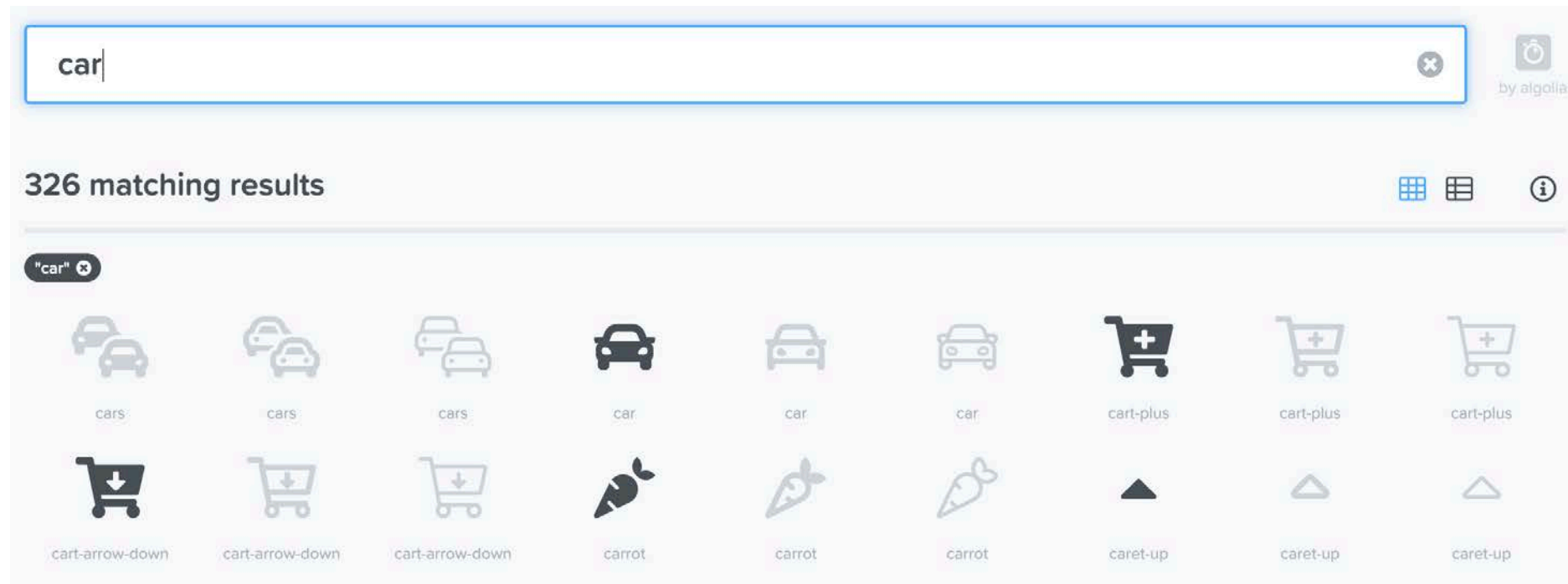
```
your-template.html

<!doctype html>
<html>
  <head>
    <!-- Place your kit's code here -->
    <script src="https://kit.fontawesome.com/e7a4853152.js"></script>
  </head>

  <body>
    <!-- Ready to use Font Awesome. Activate interlock. Dynotherms - c
  </body>
</html>
```

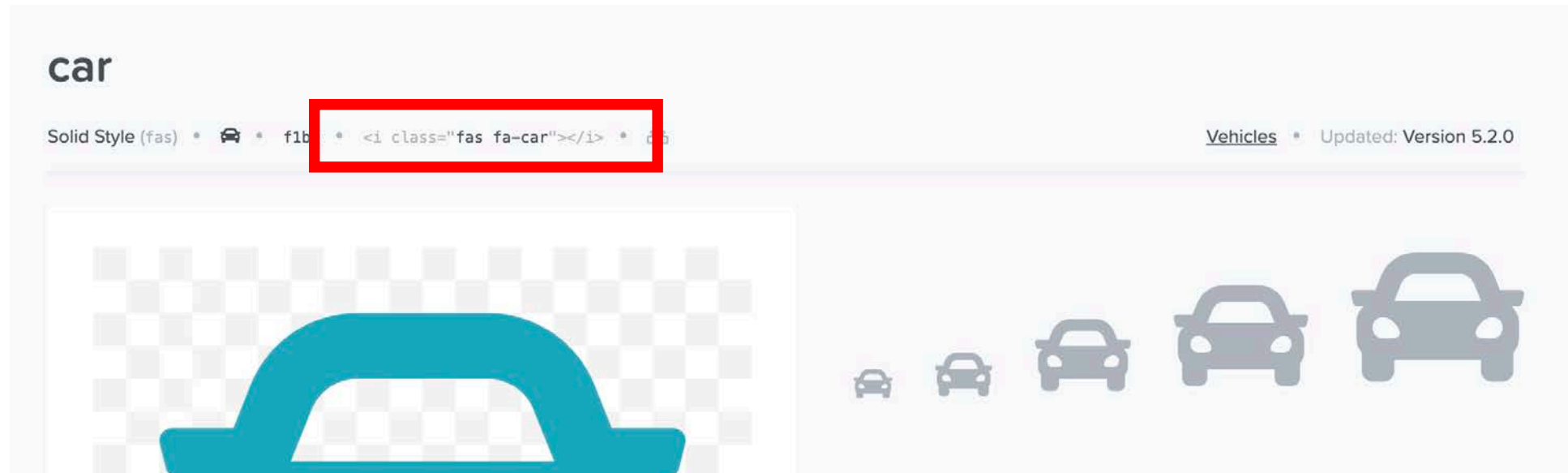
Font-awesome 사용법

<https://fontawesome.com/icons>에서 원하는 아이콘을 검색한 후 클릭
(검은색으로 활성화된 아이콘만 무료로 사용 가능)



Font-awesome 사용법

아래의 코드를 복사해서 그대로 사용하면 됨



Fabcar Application Directory 구조

```
└─ APPLICATION
  ├── node_modules
  ├── public
  │   └─ stylesheets
  │       └─ # style.css
  ├── views
  │   ├── partials
  │   │   ├── <> head.ejs
  │   │   ├── <> navbar.ejs
  │   │   ├── <> changeowner.ejs
  │   │   ├── <> createcar.ejs
  │   │   ├── <> index.ejs
  │   │   └─ <> query.ejs
  │   ├── {} package-lock.json
  │   ├── {} package.json
  │   └─ JS server.js
```

node_modules: npm을 통해 다운받은 외부 모듈들을 저장하는 곳

public: 프로젝트 전체에 적용될 css파일을 모듈화 한 폴더

일반적으로 css library는 public/stylesheets/style.css에 저장

views: 웹 상에서 보이는 부분(뷰), html 페이지들을 모아놓은 폴더. 본 프로젝트에서는 ejs view template을 사용했기 때문에 .html이 아닌 .ejs 파일들이 있다

partials: view 파일 중에서도 공용으로 쓰이는 부분을 모듈화 해놓은 부분.

package-lock.json: node_modules 트리나 package.json에 변화가 생기면 자동으로 생성되는 파일

package.json: npm init을 통해 생성되는 프로젝트 의존 모듈 관리 파일

server.js: 서버를 구축하고 구동시키는 파일. 라우팅 기능 포함 (보통은 라우터도 모듈화 해서 따로 빼놓아야 하지만 페이지가 3개밖에 없어서 그냥 서버에서 라우팅하였음)



4개의 페이지 (1) index.ejs

 Main Page



Welcome!

Select A Nextpage

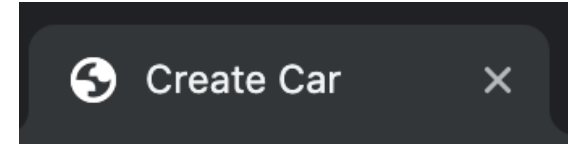
Home


[Query all cars](#)

[Create a car](#)

[Change owner](#)

4개의 페이지 (2) createcar.ejs





CreateCar Page

Create car

[Home](#) [Query all cars](#) [Create a car](#) [Change owner](#)

CarNo

Colour

Maker

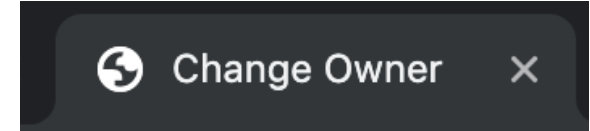
Model

Owner

SUBMIT



4개의 페이지 (3) changeowner.ejs



ChangeOwner Page

Owner change!

[Home](#)[Query all cars](#)[Create a car](#)[Change owner](#)

CarNo

Owner

SUBMIT

4개의 페이지 (4) query.ejs

```
Transaction has been evaluated, result is: [{"Key":"CAR0", "Record":{"colour":"blue", "make":"Toyota", "model":"Prius", "owner":"Tomoko"}}, {"Key":"CAR1", "Record":{"colour":"red", "make":"Ford", "model":"Mustang", "owner":"Brad"}}, {"Key":"CAR2", "Record":{"colour":"green", "make":"Hyundai", "model":"Tucson", "owner":"Jin Soo"}}, {"Key":"CAR3", "Record":{"colour":"yellow", "make":"Volkswagen", "model":"Passat", "owner":"Max"}}, {"Key":"CAR4", "Record":{"colour":"black", "make":"Tesla", "model":"S", "owner":"Adriana"}}, {"Key":"CAR5", "Record":{"colour":"purple", "make":"Peugeot", "model":"205", "owner":"Michel"}}, {"Key":"CAR6", "Record":{"colour":"white", "make":"Chery", "model":"S22L", "owner":"Aarav"}}, {"Key":"CAR7", "Record":{"colour":"violet", "make":"Fiat", "model":"Punto", "owner":"Pari"}}, {"Key":"CAR8", "Record":{"colour":"indigo", "make":"Tata", "model":"Nano", "owner":"Valeria"}}, {"Key":"CAR9", "Record":{"colour":"brown", "make":"Holden", "model":"Barina", "owner":"Shotaro"}}]
```

Json을 key-value parsing하여 ejس view에 테이블로 뿌리기

페이지 분석과 구현 계획 (1)

- 모든 페이지에서 부트스트랩 네비게이션을 공통적으로 사용한다

- 모든 페이지에서 Navbar 코드가 반복될 것이다

Navbar부분의 코드 모듈화
partials/navbar.ejs

- 모든 페이지에서 head부분에 부트스트랩 CDN을 포함시켜야 한다

- 모든 페이지에서 font-awesome 아이콘을 사용한다

- 모든 페이지에서 head 부분에 font-awesome CDN을 포함시켜야 한다.

Head에 들어갈 코드 모듈화
partials/head.ejs

- 페이지를 이동할 때마다 네비게이션의 탭이 바뀐다

- 페이지를 이동할 때마다 크롬 브라우저 제목이 바뀐다

Server에서 페이지별로
라우팅 시 쿼리를 날려서 구현

페이지 분석과 구현 계획 (2)

CarNo

Colour

Maker

Model

Owner

CarNo

Owner

- Createcar와 Changeowner 페이지에서 form과 button을 공통적으로 사용한다.
- > 공통 css를 적용하기 위해서 public/stylesheets/style.css라는 CSS 라이브러리를 모듈화 한다



Views/index.ejs

Font-awesome 아이콘



Welcome!

Select A Nextpage

Home

[Query all cars](#)

[Create a car](#)

[Change owner](#)

Bootstrap 네비게이션

server.js

- `npm install ejs` // Ejs 템플릿을 사용하기 위해 모듈 설치

```
20 // static파일의 경로 찾기
21 app.use(express.static(path.join(__dirname, 'public')));
22
23 // ejs view template
24 app.set('views', path.join(__dirname, 'views'));
25 app.set('view engine', 'ejs');
26
27 // Index page
28 app.get('/', function (req, res) {
29   res.render('index', { title: "Main Page", activate: "index"});
30 });
```

Root path(/)에 접근할 경우

서버에서 title이라는 이름의 쿼리를 "main page"라는 value를 담아서 보냄

Activate라는 이름의 쿼리를 "index"라는 value를 담아서 보냄

server.js



```
32 // Query all cars page
33 app.get('/api/query', function (req, res) {
34 |   res.render('query', { title: "Query", activate: "query" });
35 });
```

`/api/query` 라는 페이지에 get방식으로 접근할 경우
서버에서 title이라는 이름의 쿼리를 "Query"라는 value를 담아서 보냄
Activate라는 이름의 쿼리를 "query"라는 value를 담아서 보냄

server.js

```
115 // Create car page
116 app.get('/api/createcar', function (req, res) {
117     res.render('createcar', { title: "Create Car", activate: "createcar" });
118 });
```

`/api/createcar` 라는 페이지에 get방식으로 접근할 경우
서버에서 title이라는 이름의 쿼리를 "Create Car"라는 value를 담아서 보냄
Activate라는 이름의 쿼리를 "createcar"라는 value를 담아서 보냄

server.js

```
169 // Change car owner page
170 app.get('/api/changeowner', function (req, res) {
171 |   res.render('changeowner', { title: "Change Owner", activate: "changeowner" });
172 });
```

`/api/changeowner` 라는 페이지에 get방식으로 접근할 경우
서버에서 title이라는 이름의 쿼리를 "Change Owner"라는 value를 담아서 보냄
Activate라는 이름의 쿼리를 "changeowner"라는 value를 담아서 보냄

Views/index.ejs

```
views ▸ <> index.ejs ▸ ...
1  <!DOCTYPE html>
2  <html>
3    <head>                                모든 페이지에 적용되는 부분이므로 partials 모듈화하여 include
4      <%- include('partials/head') %>
5    </head>
6
7    <body>                                Bootstrap에 미리 정의되어 있는 compnent
8      <div class="container">
9        <i class="fas fa-smile"></i> Font-awesome
10       <h1>Welcome!</h1>
11       <p>Select A Nextpage</p>
12       <%- include('partials/navbar') %>
13     </div>                                모든 페이지에 적용되는 부분이므로 partials 모듈화하여 include
14   </body>
15 </html>
```

Partials/head.ejs

각종 외부 모듈을 사용하기 위한 CDN들

```
1 <!-- popper.js -->
2 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KcKrR/rE9/Qpg6aAZGJwFDMVNA/C
3 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHp0f/4+1nzFpr53r
4 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi0
5
6 <!-- jquery ajax -->
7 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
8
9 <!-- bootstrap -->
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
11 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap-theme.min.css">
12 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
13
14 <!-- fontawsome -->
15 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.9.0/css/all.css">
16 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.9.0/css/v4-shims.css">
17
18 <!-- customize css --> 모듈화 해 놓은 Public/stylesheets/style.css를 읽어들인다.
19 <link rel="stylesheet" type="text/css" href="../stylesheets/style.css">
20
21 <title><%= title %></title>
22
```

서버에서 보냈던 title 이라는 쿼리를 ejs파일에서 읽어들인다.
<%= %> 로 감싸면 서버에서 보낸 쿼리를 ejs에서 읽을 수 있음

Partials/navbar.ejs

서버에서 메인 페이지로 라우팅 요청이 왔을때 보냈던
activate라는 쿼리가 index라는 값을 가지고 있다면

Home 탭을 활성화 시킨다

```
1 <% if(activate=="index"){ %>
2   <ul class="nav nav-tabs">
3     <li role="presentation" class="active"><a href="#">Home</a></li>
4     <li role="presentation"><a href="/api/query">Query all cars</a></li>
5     <li role="presentation"><a href="/api/createcar">Create a car</a></li>
6     <li role="presentation"><a href="/api/changeowner">Change owner</a></li>
7   </ul>
8 <% }else if(activate=="query"){ %>
9   <ul class="nav nav-tabs">
10    <li role="presentation"><a href="/">Home</a></li>
11    <li role="presentation" class="active"><a href="/../api/query">Query all cars</a></li>
12    <li role="presentation"><a href="/../api/createcar">Create a car</a></li>
13    <li role="presentation"><a href="/../api/changeowner">Change owner</a></li>
14  </ul>
15 <% }else if(activate=="createcar"){ %>
16   <ul class="nav nav-tabs">
17     <li role="presentation"><a href="/">Home</a></li>
18     <li role="presentation"><a href="/../api/query">Query all cars</a></li>
19     <li role="presentation" class="active"><a href="/../api/createcar">Create a car</a></li>
20     <li role="presentation"><a href="/../api/changeowner">Change owner</a></li>
21   </ul>
22 <% }else{ %>
23   <ul class="nav nav-tabs">
24     <li role="presentation"><a href="/">Home</a></li>
25     <li role="presentation"><a href="/../api/query">Query all cars</a></li>
26     <li role="presentation"><a href="/../api/createcar">Create a car</a></li>
27     <li role="presentation" class="active"><a href="/../api/changeowner">Change owner</a></li>
28   </ul>
29 <% } %>
```


Partials/navbar.ejs

```
1 <% if(activate=="index"){ %>
2   <ul class="nav nav-tabs">
3     <li role="presentation" class="active"><a href="#">Home</a></li>
4     <li role="presentation"><a href="/api/query">Query all cars</a></li>
5     <li role="presentation"><a href="/api/createcar">Create a car</a></li>
6     <li role="presentation"><a href="/api/changeowner">Change owner</a></li>
7   </ul>
8   <% }else if(activate=="query"){ %>
9     <ul class="nav nav-tabs">
10      <li role="presentation"><a href="/">Home</a></li>
11      <li role="presentation" class="active"><a href="/../api/query">Query all cars</a></li>
12      <li role="presentation"><a href="/../api/createcar">Create a car</a></li>
13      <li role="presentation"><a href="/../api/changeowner">Change owner</a></li>
14    </ul>
15    <% }else if(activate=="createcar"){ %>
16      <ul class="nav nav-tabs">
17        <li role="presentation"><a href="/">Home</a></li>
18        <li role="presentation"><a href="/../api/query">Query all cars</a></li>
19        <li role="presentation" class="active"><a href="/../api/createcar">Create a car</a></li>
20        <li role="presentation"><a href="/../api/changeowner">Change owner</a></li>
21      </ul>
22    <% }else{ %>
23      <ul class="nav nav-tabs">
24        <li role="presentation"><a href="/">Home</a></li>
25        <li role="presentation"><a href="/../api/query">Query all cars</a></li>
26        <li role="presentation"><a href="/../api/createcar">Create a car</a></li>
27        <li role="presentation" class="active"><a href="/../api/changeowner">Change owner</a></li>
28      </ul>
29    <% } %>
```

서버에서 메인 페이지로 라우팅 요청이 왔을때 보냈던
activate라는 쿼리가 query라는 값을 가지고 있다면

query 탭을 활성화 시킨다

Views/createcar.ejs

```
views ▸ <> createcar.ejs ▸ ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <%- include('partials/head') %>
5  </head>
6
7  <body>
8    <div class="container">
9      <i class="fas fa-car"></i>
10     <h1>CreateCar Page</h1>
11     <p>Create car</p>
12     <%- include('partials/navbar') %>
13
14     <form method="post">
15       CarNo <input type="text" name="carno" /> <br>
16       Colour <input type="text" name="colour" /> <br>
17       Maker <input type="text" name="make" /> <br>
18       Model <input type="text" name="model" /> <br>
19       Owner <input type="text" name="owner" /> <br>
20       <button type="submit" class="btn btn-warning">SUBMIT</button>
21     </form>
22   </div>
23 </body>
24 </html>
```


Partials/head.ejs

각종 외부 모듈을 사용하기 위한 CDN들

```
1 <!-- popper.js -->
2 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/C
3 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHp0f/4+1nzFpr53r
4 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi0
5
6 <!-- jquery ajax -->
7 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
8
9 <!-- bootstrap -->
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
11 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap-theme.min.css">
12 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
13
14 <!-- fontawsome -->
15 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.9.0/css/all.css">
16 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.9.0/css/v4-shims.css">
17
18 <!-- customize css --> 모듈화 해 놓은 Public/stylesheets/style.css를 읽어들인다.
19 <link rel="stylesheet" type="text/css" href="../stylesheets/style.css">
20
21 <title><%= title %></title>
22
```

서버에서 보냈던 title 이라는 쿼리를 ejs파일에서 읽어들인다.
<%= %> 로 감싸면 서버에서 보낸 쿼리를 ejs에서 읽을 수 있음

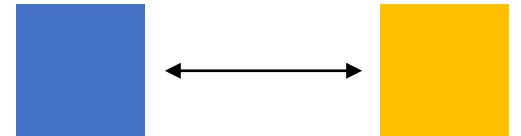
Public/stylesheets/style.css

앞에 . (comma) 를 붙이면
Class에 적용하겠다는 뜻

아무것도 없이 그냥 쓰면
Tag 자체에 적용하겠다는 뜻

```
public ▸ stylesheets ▸ # style.css ▸ ...
1  .container{      가운데 정렬
2      text-align: center;
3      margin-top: 30px;
4  }                Margin: 바깥 여백을 준다
5
6  .nav{            Top/bottom/left/right으로 방향 지정 가능
7      margin-bottom: 50px;
8  }
9
10 input{
11     margin-left: 30px;
12     margin-bottom: 20px;
13     border-radius: 5px;
14     padding: 10px 15px;
15     width: 30%;
16 }                Border-radius: 모서리를 둥글게
                  픽셀(px)값이 클수록 더 둥글어진다
17
18 i{                Padding: 안쪽 여백을 준다
19     font-size: 50px;
20 }                Margin과 마찬가지로 방향 지정 가능
                  글씨 크기 조절
21
22 button{
23     margin-top: 30px;
24     padding: 10px 40px;
25 }
```

Margin (바깥 여백)



Padding (안쪽 여백)

