



HTML / CSS 를 이용한 페이지 제작

Web 개발 기초





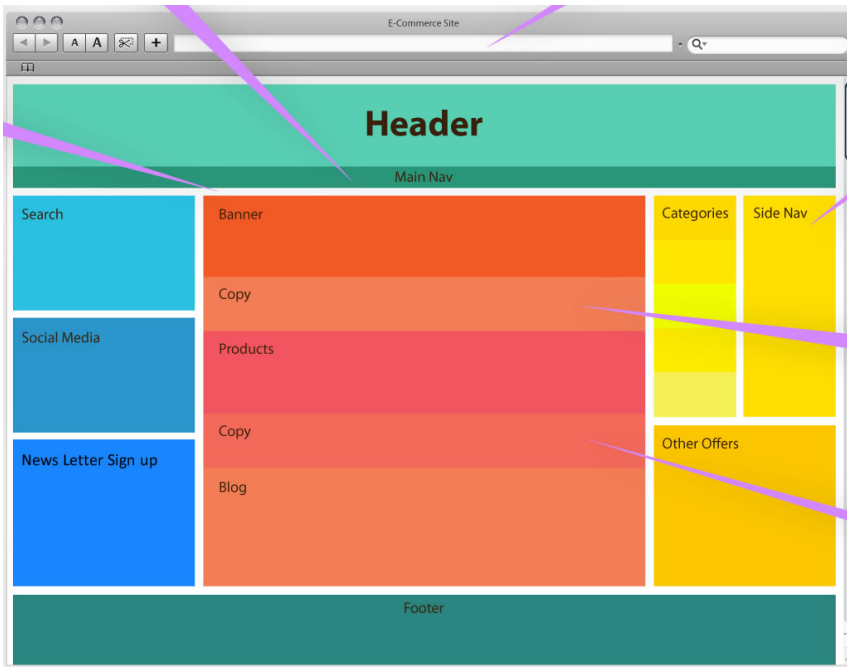
1. 기초 형태 만들기

2. 디테일 한 형태 만들고 겉모양 완성



1. 기초 형태 만들기

큰 틀 잡기



위와 같이 website의 기본 골격을 먼저 생각해야 한다.

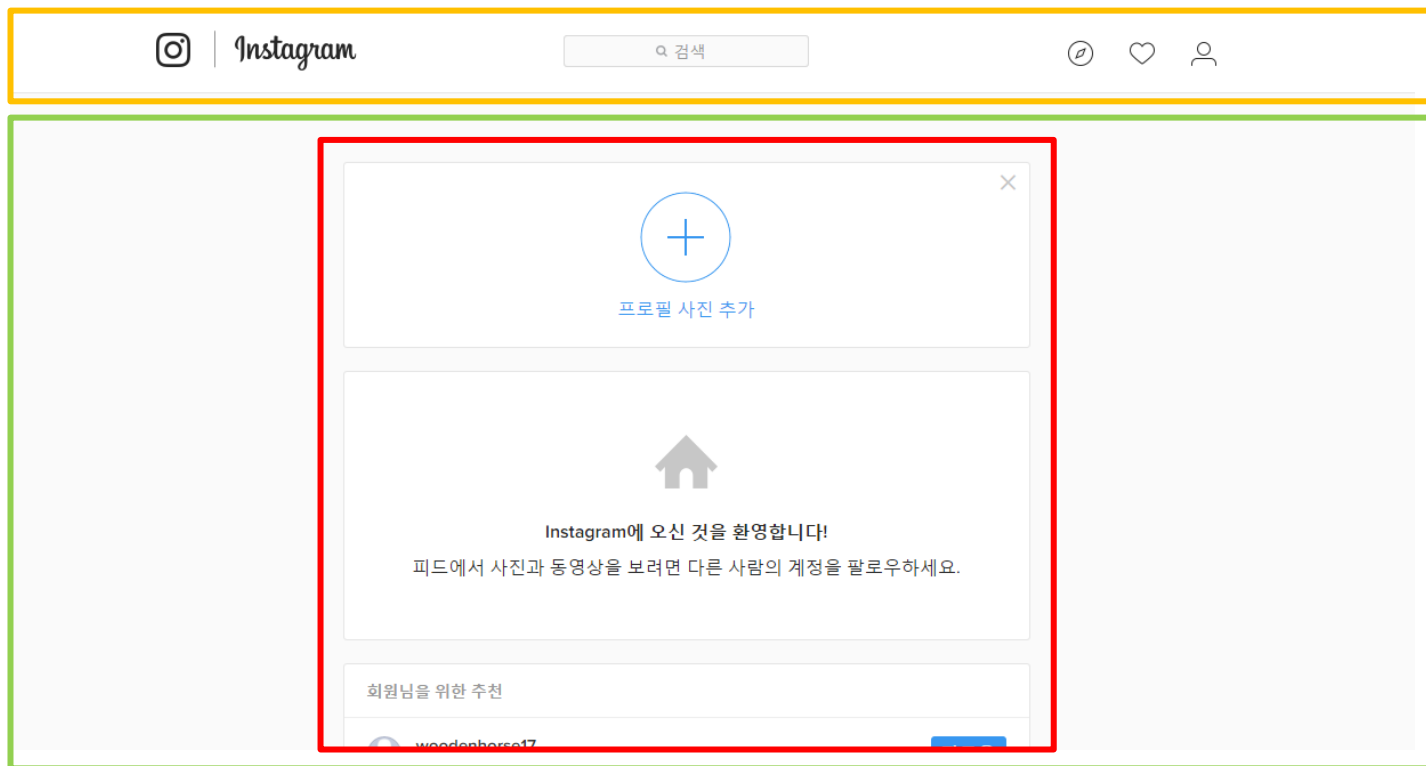
위 사진에 명시된 header, section, footer 이런 것들은 꼭 써야하는 건 아니지만, 통일성과 편의성을 위해서 사용이 권장된다.

하지만 잘 지켜지지 않는다.

1. 기초 형태 만들기

큰 틀 잡기

이제 우리가 만드려고 하는 인스타그램을 한 번 보자.



대충, 이 정도로 분석되고, 또 내부 글의 형태 또한 위와 흡사하다.

HTML으로 생각 해보면 가장 바깥의 초록색이 부모태그, 그 안의 빨간색이 자식 태그이다.

빨간색 박스 안에서 다시 나뉘어 지면 같은 박스 안에 있으므로 형제 태그이다.

1. 기초 형태 만들기

큰 틀 잡기

Atom 을 켜고 connect를 하여 views의 index.ejs를 연다

```
1  <html>
2    <head>
3      <title>
4        <%=title%>
5      </title>
6    </head>
7    <body>
8      <div>
9        안녕하세요!
10     </div>
11   </body>
12 </html>
```

Head 태그는 실제로 웹사이트 내부에 보여지는 부분은 아니다.
웹 페이지 상단의 title을 적어 놓거나 외부 javascript, css 와 같은 것들을 추가할 때 이 부분에 넣는다.

우리가 웹사이트 내부에 보여줄 내용은 Body 태그 안에 채워 넣는다.

1. 기초 형태 만들기

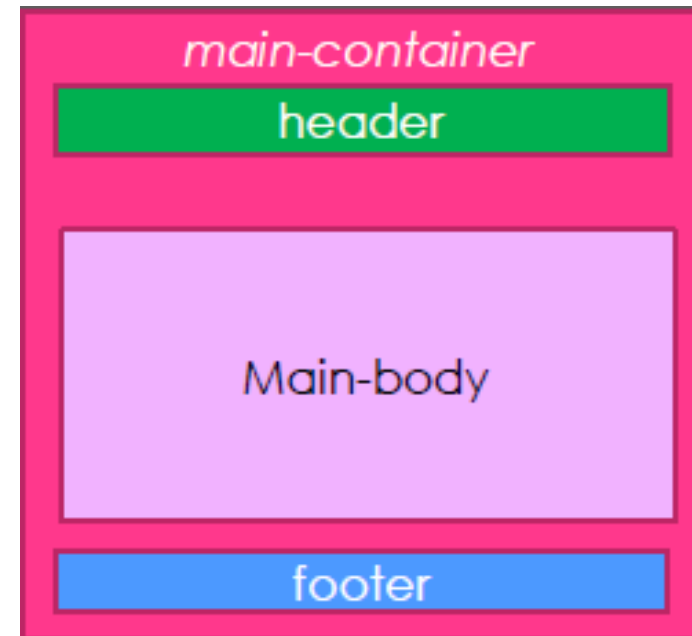
큰 틀 잡기

Body 태그 안에 우리가 아까 큰 틀 잡을 때 확인한 것에 맞게 코드를 작성한다.

이 때 우리는 class 라는 것을 이용한다.

Class는 html 태그들을 분류하기 위한 속성이다.

```
1 <html>
2   <head>
3     <title>
4       KUstagram
5     </title>
6   </head>
7   <body>
8     <div class="main-container">
9       <div class="header"></div>
10      <div class="main-body"></div>
11      <div class="footer"></div>
12    </div>
13  </body>
14 </html>
```



1. 기초 형태 만들기

큰 틀 잡기

이제 header, body, footer 로 나뉘어진 부분을 디테일하게 만들어 본다.

 | Instagram

🔍 검색



header



Instagram에 오신 것을 환영합니다!

피드에서 사진과 동영상을 보려면 다른 사람의 계정을 팔로우하세요.

body

회원님을 위한 추천

 woodenhorse17

2. 디테일 한 형태 만들고 겉모양 완성

header

먼저 header 를 만든다.

```
<div class="header">
  <div class="header-container">
    <div class="header-logo">
      <img src="" />
    </div>
    <div class="header-search">
      <input type="text" placeholder="검색하세요" />
    </div>
    <div class="header-setting">
      <span>설정</span>
    </div>
  </div>
</div>
```

설정

브라우저에서 확인해보면 이렇게 어처구니 없이 나오는 것을 알 수 있다.
이제 이미지파일과 css를 이용해 그럴 듯 하게 꾸며보자.

2. 디테일 한 형태 만들고 겉모양 완성

header

Static > img 폴더에 kustagram_logo.png 파일을 넣는다.

** Atom 으로 서버에 파일 올리는 방법

1. 먼저 올릴 파일을 구한다

(여기서는 kustagram_logo.png 를 git 이나 ppt에서 받는다)

KUstagram

2. 디테일 한 형태 만들고 겉모양 완성

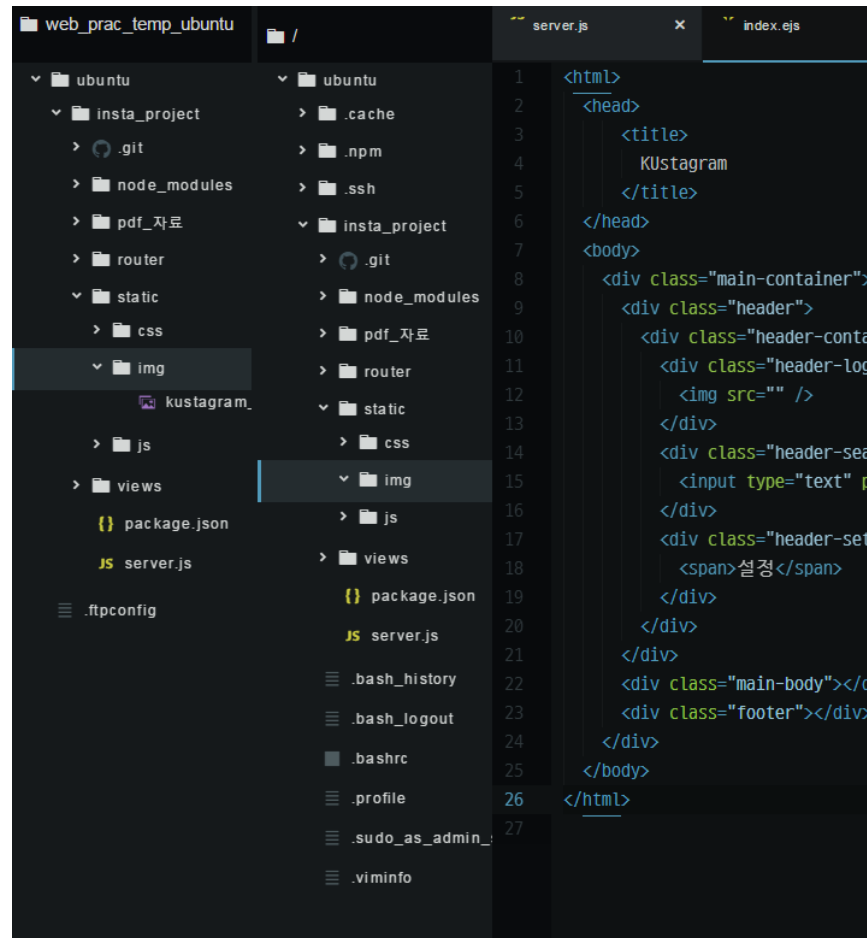
header

2. Atom 에서 서버와 연동시켰던 폴더의 디렉토리와 서버를 connect 한 디렉토리를 전부 띄운다

>>> 왼쪽에 두 개의 패널이 뜨게 된다.

만약에 connect 한 패널은 있는데 서버와 연동시켰던 폴더가 안뜨면 ctrl + \ 을 누르면 뜬다

이 부분은 계속 필요한 내용이니까 모르는 부분 질문 ㄱㄱ

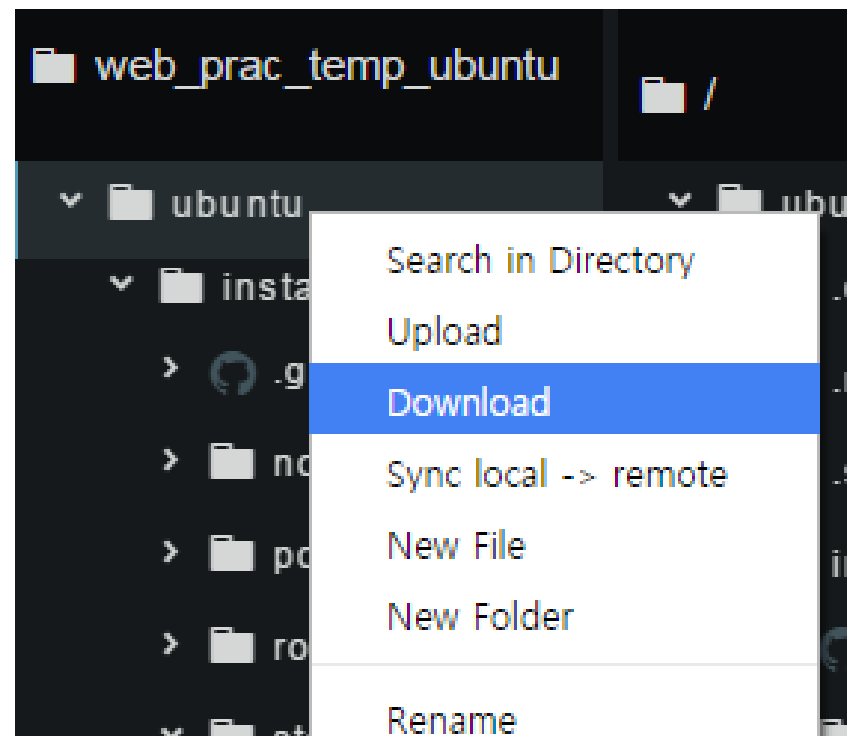


2. 디테일 한 형태 만들고 겉모양 완성

header

3. 서버와 연동시켰던 폴더에서 오른쪽 버튼을 클릭하고 Download를 누른다
그러면 서버 쪽 파일들이 연동시킨 폴더에 그대로 다운로드 된다.

>>> 현재 컴퓨터의 폴더와 서버의 폴더 내용을 같게 만들어 주는 과정이다.

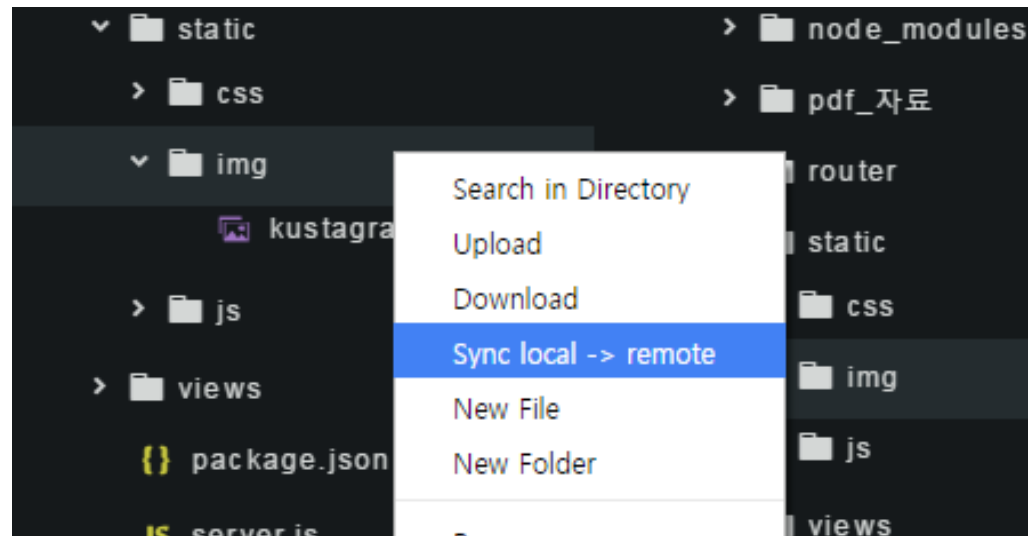
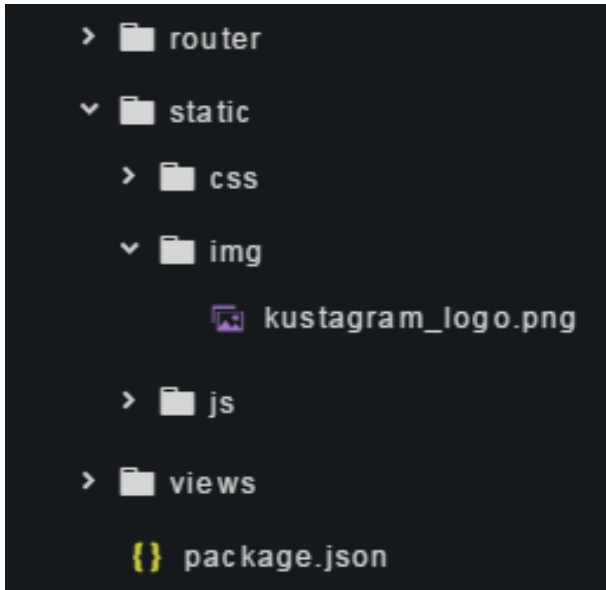


2. 디테일 한 형태 만들고 겉모양 완성

header

4. 서버와 연동시켰던 폴더의 원하는 곳에 아까 받은 파일을 넣는다
(여기서는 static > img 폴더에 kustagram_logo.png 파일을 넣는다.)

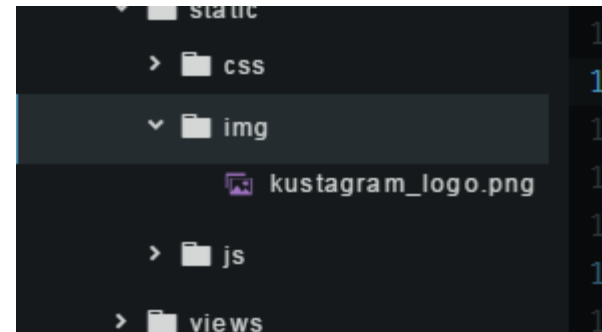
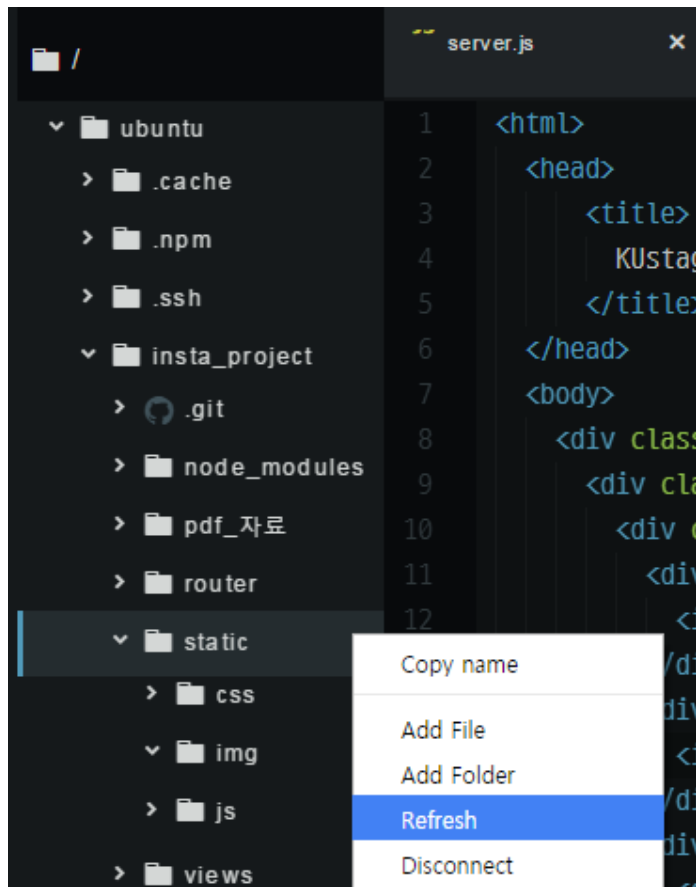
5. 그리고 Sync local -> remote 를 클릭



2. 디테일 한 형태 만들고 겉모양 완성

header

5. 서버와 connect 한 디렉토리에서 refresh를 해주면 서버의 static > img 에 이미지 파일이 이동된 것을 알 수 있다.



2. 디테일 한 형태 만들고 겉모양 완성

header

Index.ejs 에서 img 경로에 img/kustagram_logo.png 라고 경로를 적어준다.

```
<div class="header-logo">  
    
</div>
```

브라우저에서 새로고침을 해 보면 사진파일이 잘 뜨는 것을 확인할 수 있다.

KUstagram

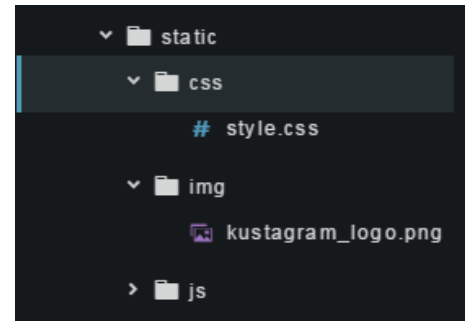
설정

그럼 이제 스타일을 적용해본다.

2. 디테일 한 형태 만들고 겉모양 완성

header

Atom 에서 css 폴더에 style.css 파일을 만들어 준다.



그리고 style.css 에 우측과 같이 입력한다.

```
1  body {  
2      background-color: #eeeeee;  
3      margin: 0;  
4  }  
5  .header {  
6      width: 100%;  
7      height: 50px;  
8      border-bottom: 1px solid #999999;  
9      text-align: center;  
10     background-color: #ffffff;  
11 }  
12 .header .header-logo img {  
13     width: 100px;  
14     vertical-align: middle;  
15 }
```

2. 디테일 한 형태 만들고 겉모양 완성

header

Index.ejs 로 돌아와 <head></head> 사이에
<link rel="stylesheet" href="css/style.css" /> 를 적어준다.

```
<head>
  <title>
    KUstagram
  </title>
  <link rel="stylesheet" href="css/style.css" />
</head>
```

브라우저에서 새로고침을 해보면

KUstagram

설정

Css 가 잘 적용되는 것을 확인 할 수 있다.

이는 css 를 link 태그를 통해 불러왔기 때문에 가능한 것이다.

2. 디테일 한 형태 만들고 겉모양 완성

header

이제 애네를 정렬을 시켜주자.
다시 style.css 로 가서 이어서 작성한다.

```
.header-container div {  
  display: inline-block;  
  padding-top: 10px;  
}
```

브라우저에서 확인해 보면 아주 예쁘게 가운데로 모인 것을 확인 할 수 있다.

KUstagram 설정

하지만 우리가 하고싶은 것은 모두 가운데로 모으는 것이 아니다!

인스타그램 처럼 로고는 왼쪽 끝으로, 설정은 오른쪽 끝으로 보내자.

2. 디테일 한 형태 만들고 겉모양 완성

header

Style.css 에서 이어서 쓰자.

```
.header-logo {  
  float:left;  
}  
  
.header-setting {  
  float:right;  
}
```

브라우저에서 확인해보면, 끝으로 가긴 갔는데 너무 끝으로 갔다.

KUstagram

설정

이것은 HTML/CSS 에서 잠시 언급했던 반응형 CSS 를 이용하여 처리하자
이때 바로 미디어쿼리 라는 것을 쓴다.

```
@media (min-width : 768px) {  
  .header-container {  
    width:100%;  
  }  
}  
  
@media (min-width : 992px) {  
  .header-container {  
    width:800px;  
  }  
}
```

2. 디테일 한 형태 만들고 겉모양 완성

header

이제 간격은 맞는데 왼쪽으로 치우친다.

KUstagram

설정

Css 에 추가해주면

```
.header-container {  
  margin: auto;  
}
```

가운데로 정렬이 잘 되는 것을 확인할 수 있다.

KUstagram

설정

2. 디테일 한 형태 만들고 겉모양 완성

body

가장 중요한 body 부분을 만들 차례다

우선 main-body 클래스의 div 사이에 아래와 같이 입력한다.

이제 대부분의 구성요소들을 'head-body-foot' 형태로 작성할 것이다.

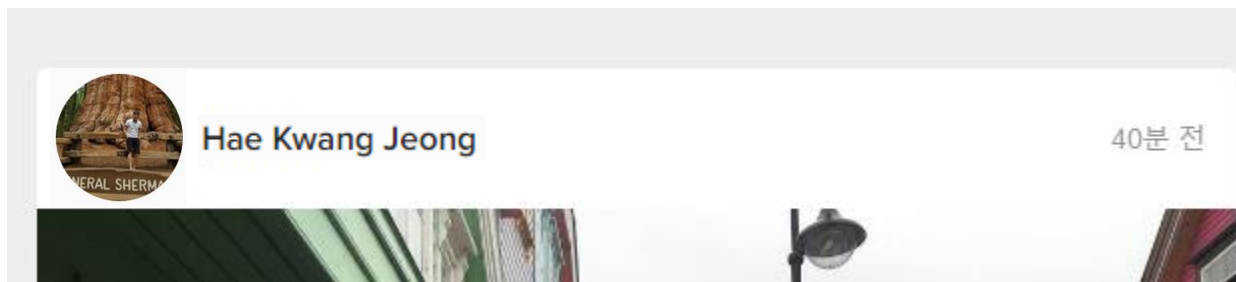
```
<div class="main-body">  
  <div class="post-header">  
  
  </div>  
  <div class="post-body">  
  
  </div>  
  <div class="post-footer">  
  
  </div>  
</div>
```

** 갑자기 생각난것 >> Atom 에서 autoclose-html 라는 외부 package 를 설치하자
<div> 라고 쓰면 자동으로 </div> 로 닫아준다!

2. 디테일 한 형태 만들고 겉모양 완성

body

인스타그램의 포스트에서 포스트 상단은 아래 형식처럼 따올것이다.
(인스타의 구성 형식은 계속해서 변화하니 일단 그냥 이 형식을 따를 것이다)



위 형식은 사진/이름/게시시간 이 포스트 상단에 표시된다.

이는 header를 만들 때와 다른 것이 없다.

2. 디테일 한 형태 만들고 겉모양 완성

body

적절한 프로필 사진을 구해서 서버에 올렸다는 가정 하에 아래 코드를 index.ejs 에 추가한다.

```
<div class="main-body">
  <div class="post">
    <div class="post-header">
      
      <div class="author-id">
        JOSE
      </div>
      <div class="post-reg-time">
        40분전
      </div>
    </div>
    <div class="post-body"></div>
    <div class="post-footer"></div>
  </div>
</div>
```

2. 디테일 한 형태 만들고 겉모양 완성

body

브라우저를 통해 확인해 보면 스타일이 적용되지 않은 형태로 나온다.
Css를 추가해 스타일을 적용시키자



2. 디테일 한 형태 만들고 겉모양 완성

body

아래와 같이 추가한다. (좌 > 우)

```
.post {  
  background-color: #fff;  
  margin-bottom: 10px;  
  border-radius: 5px;  
}
```

```
.post-header {  
  padding: 5px;  
  padding-left: 15px;  
  padding-right: 15px;  
  content: " ";  
  display: block;  
  clear: both;  
  height: 60px;  
}
```

```
.post-header > div {  
  display: inline;  
}
```

```
.post-header img {  
  border-radius: 50%;  
  width: 60px;  
  height: 100%;  
  float: left;  
}
```

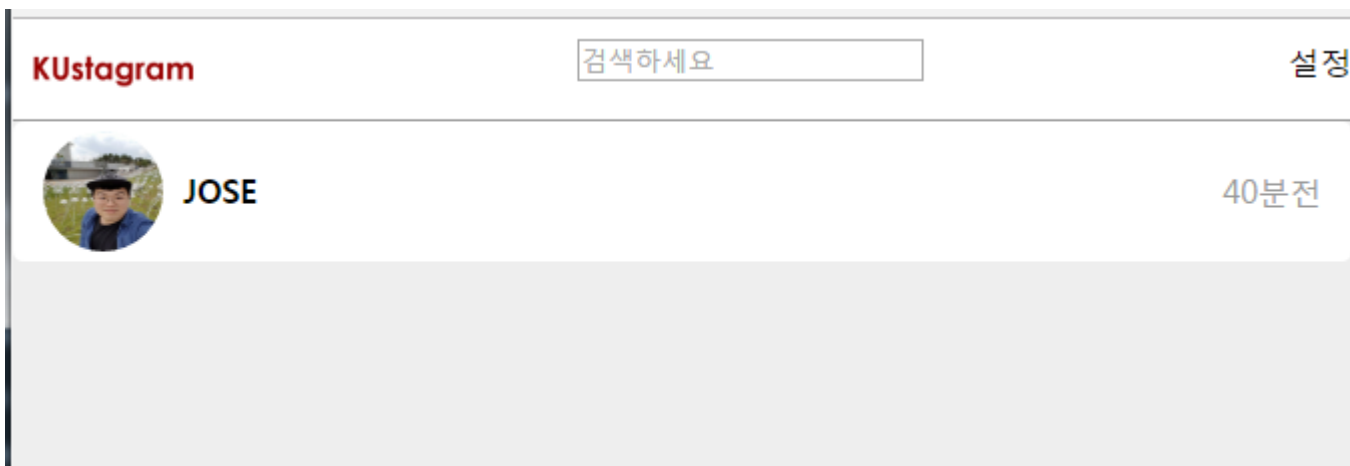
```
.post-header .author-id {  
  line-height: 60px;  
  display: inline;  
  float: left;  
  margin-left: 10px;  
  font-weight: bold;  
}
```

```
.post-header .post-reg-time {  
  float: right;  
  line-height: 60px;  
  color: #999;  
}
```


2. 디테일 한 형태 만들고 겉모양 완성

body

브라우저로 확인 해 보면 그럴 듯 하게 나왔다.





다음 시간에는...

1. HTML 템플릿 완성하기