



# 클론 코딩(2일차)

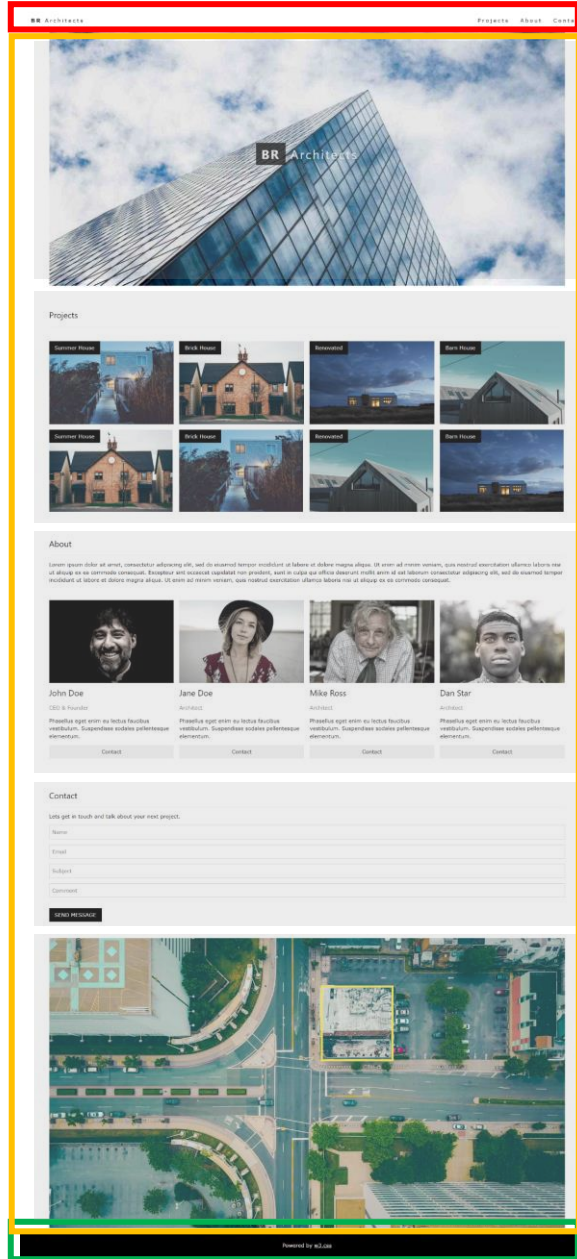
풀스택 자바 웹개발자 오후반  
방미소



# 목차

1. 레이아웃 정의
2. 레이아웃 상세
3. 구현 결과
4. 소요시간 및 후기

# 1. 레이아웃 정의 (헤더, 푸터)



Section0

Section1

Section2

Section3

Section4

## 헤더

- 브라우저를 스크롤해도 상단에 고정

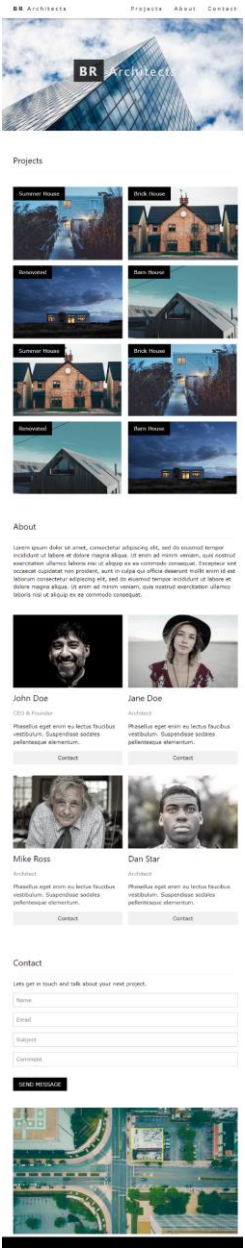
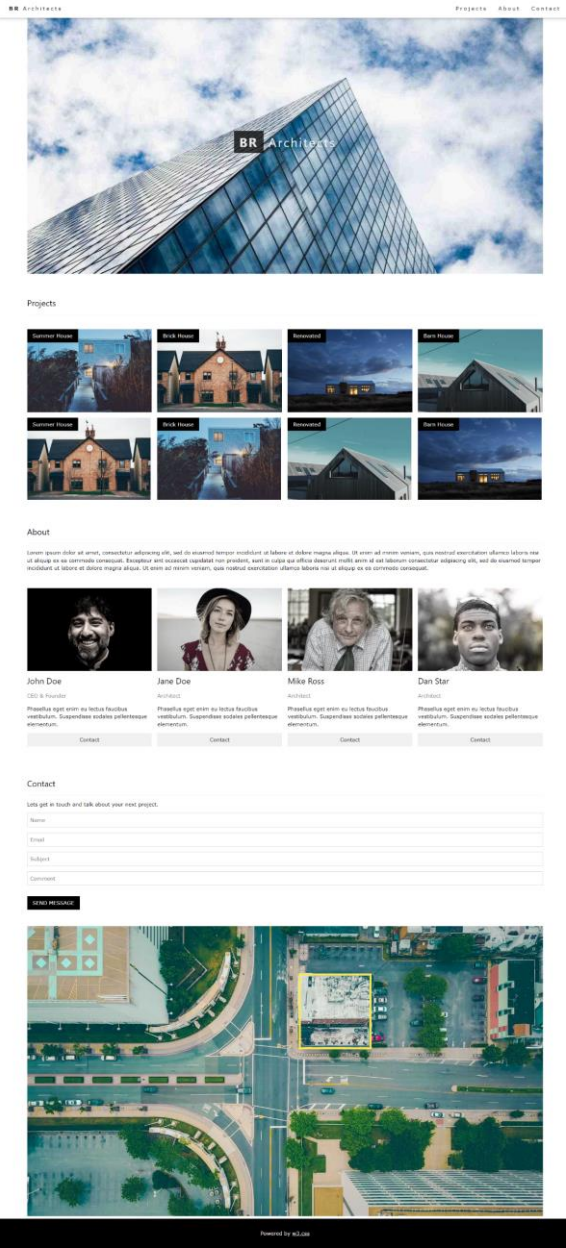
## 메인

- 크게 5개의 section으로 구분

## 푸터

- 하단 출처 영역

# 1. 레이아웃 정의



## 미디어 쿼리

0~600px

- › 상단 우측 메뉴 없음
- › 모든 요소 block 상태
- › 메인 로고 아래 타이틀 없음

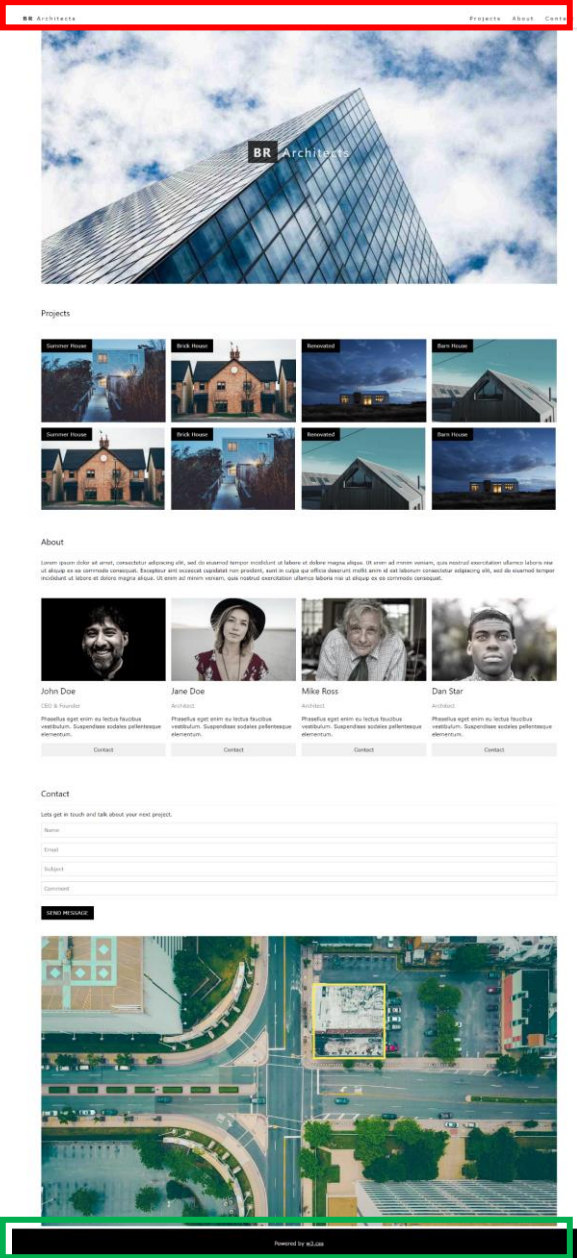
601px~ 992px

- › 상단 우측 메뉴 보임
- › 메인 로고 타이틀 (architects)
- › project 2열로 보임
- › about 2열로 보임

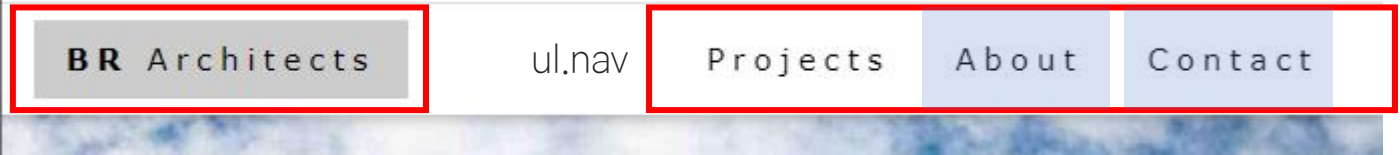
993px~1580px

- › 메인 요소들 4열 배치
- › 가로 폭만 넓어짐

## 2. 레이아웃 상세 (헤더, 푸터)

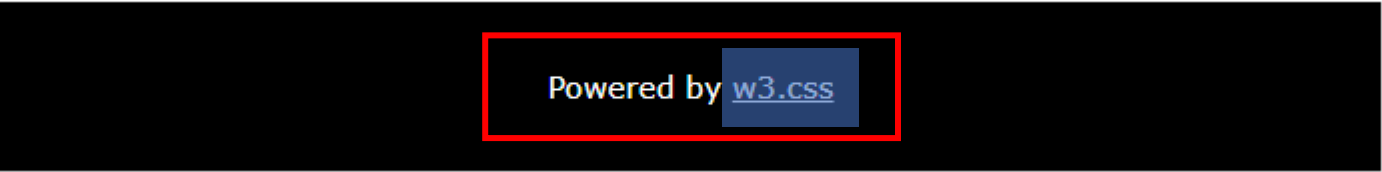


logo



### 헤더

- 왼쪽 로고, 오른쪽 메뉴 (flex)
- 브라우저를 스크롤해도 상단에 고정 (fixed)
- 마우스 위치 시 회색 박스 효과 (hover)
- 오른쪽 메뉴 클릭 시 각 article 영역으로 이동 (하이퍼링크)

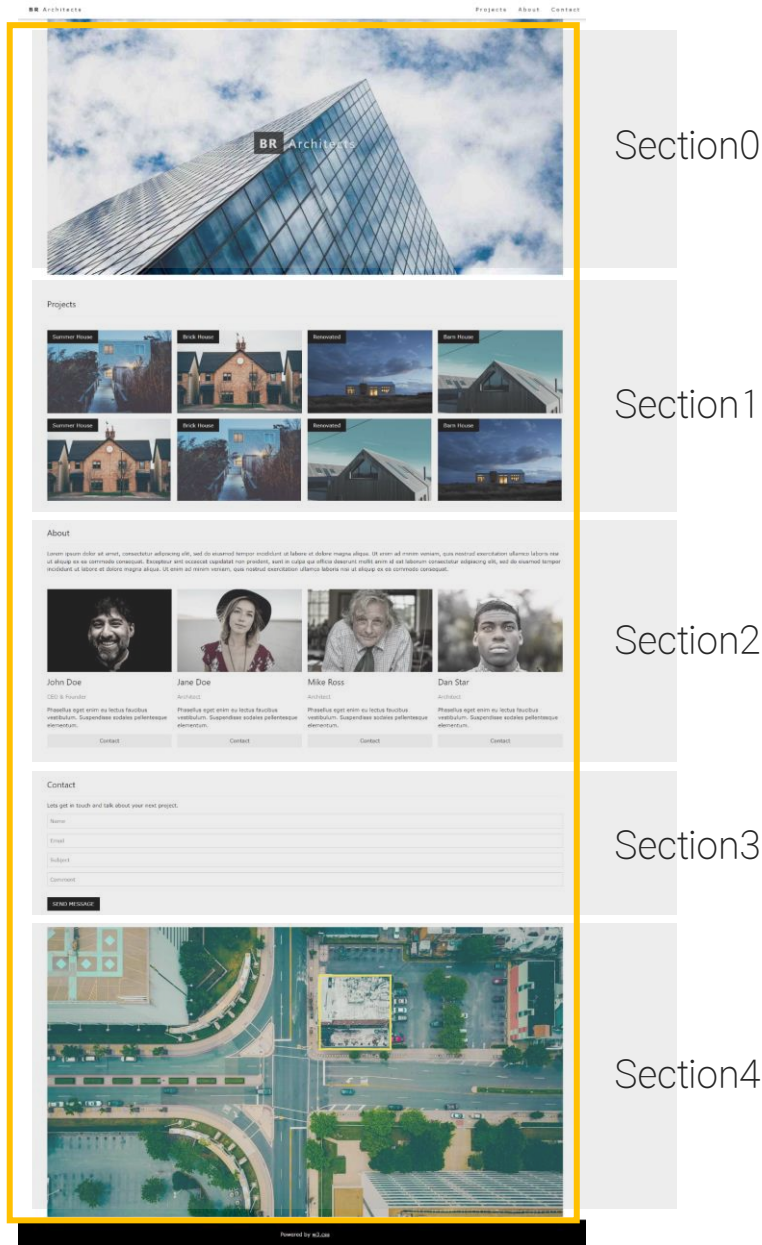


### 푸터

- 일부 글자에 하이퍼링크



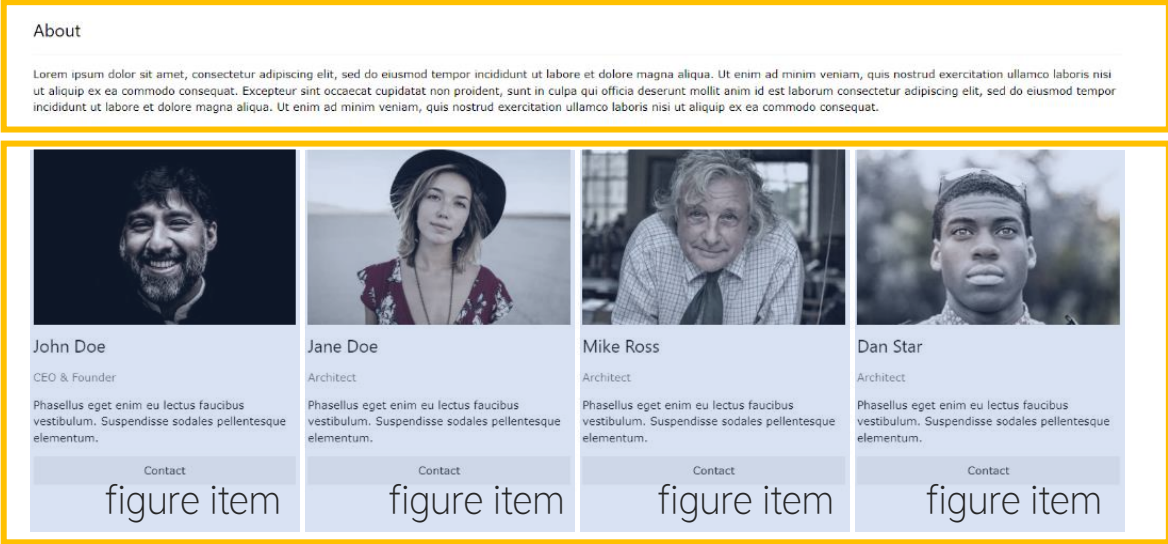
## 2. 레이아웃 상세 (메인)



### 메인

- section 0, 4는 그림으로 구성 → 브라우저 사이즈에 따라 확대
- section1과 section2에는 각각 item들이 정렬
  - 기본적으로 브라우저 사이즈에 따라 확대 & 일정 크기가 되면 열 수 변경
- section3에는 form 삽입

## 2. 레이아웃 상세 (메인)



text box

figure container

### 메인 – 각 section 구성

- text box와 각 item container로 구분
- text box 안에서 title은 <h> 태그 / 내용은 <p> 태그 활용
- item container 안에서 figure item 영역을 각각 분리
- figure item은 image, name 등 세부 영역 구분

figure image

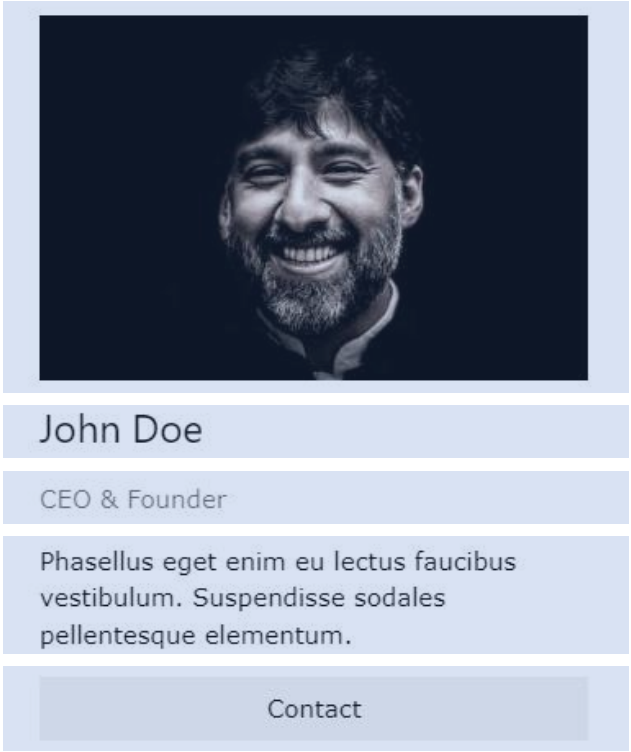


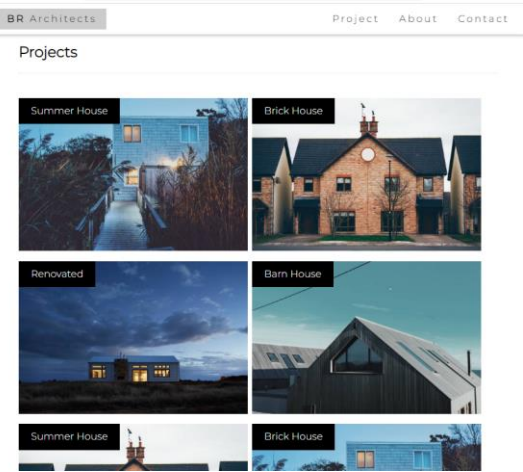
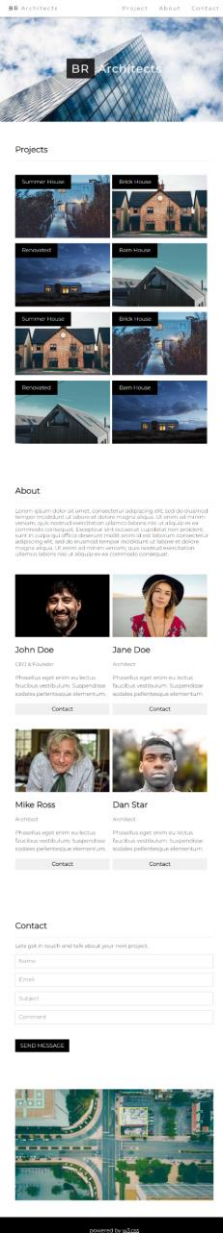
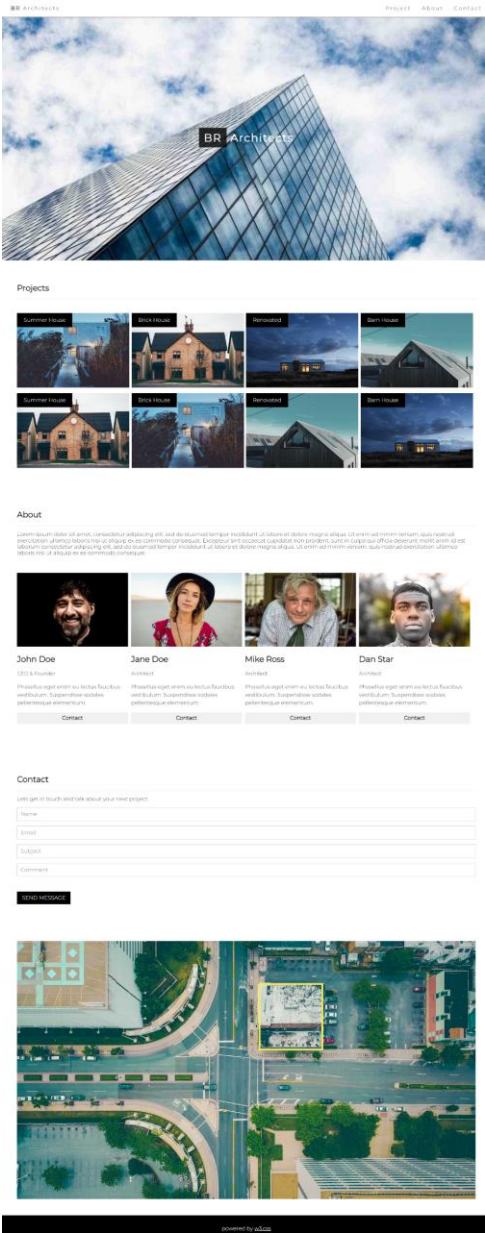
figure name

figure job

figure etc

contact button

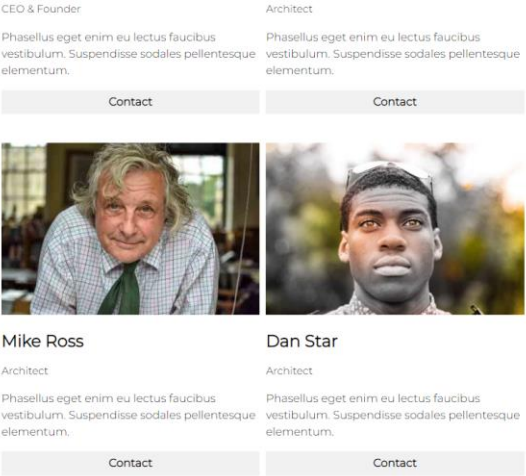
# 3. 구현 결과



## Contact

Lets get in touch and talk about your next project.

SEND MESSAGE



- 미디어 쿼리 활용한 페이지 변경사항 적용
- 상단바 고정, 메뉴 hover, 섹션 하이퍼링크 등 적용
- 인몰란 적용 (이미지 채도 수정은 구현하지 못함)
- 하단 폼 적용



### 3. 구현 결과 - 전체 영역

```
/* ----- 기본*/
* {
  box-sizing: border-box;
}

html,
body {
  width: 100%;
}

body {
  font-family: 'Montserrat', sans-serif;
}

a {
  all: unset;
  cursor: pointer;
}

ul,
ol,
li {
  list-style: none;
  padding: 0;
  margin: 0;
}

hr {
  border-top: 2px solid #f1f1f1;
}
```

```
/* ----- 바디 */
.inner {
  margin: 0 auto;
  max-width: 1580px;
  padding: 45px 35px;
}

.page-container {
  margin: 0 10px;
}
```

- CSS reset 파일 활용하여 초기화
- 이후 문서 전체에 적용할 CSS 추가 작성
- 전체 문서 width가 일정 수준을 넘지 않도록 inner 클래스에 가로 제한 적용

### 3. 구현 결과 - 헤더

BR Architects

Project

About

Contact



```
<header>
  <div class="page-container">
    <div class="head-menu">
      <div class="logo"><a href="#"><strong>BR</strong> Architects</a></div>
      <ul class="nav">
        <li><a href="#project">Project</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </div>
  </div>
</header>
```

header > logo : 좌측 로고  
> nav : 우측 메뉴

중간 div태그(container 등)는 이후 CSS로  
페이지 전체 margin, padding 적용하는데 활용

```
/* ---헤더 영역 */
header {
  width: 100%;
  background-color: #fff;
  box-shadow: 0 2px 5px 0 #000000;
  letter-spacing: 3px;
  position: fixed;
  top: 0;
  left: 0;
  z-index: 1000;
}

header .head-menu {
  margin: 8px 0;
  display: flex;
  justify-content: space-between;
}

header .logo {
  height: 100%;
  padding: 8px 16px;
}
```

```
header .logo:hover {
  background-color: #ccc;
}

header ul.nav {
  display: flex;
  height: 100%;
  align-items: center;
}

header ul.nav li {
  padding: 8px 16px;
}

header ul.nav li:hover {
  background-color: #ccc;
}
```

- 전체 header 영역 position:fixed 적용 + z-index로 최상단에 띄워줌
- 우측 메뉴는 ul > li 작성한 뒤 flex로 나열
- 로고, 메뉴 hover 시 회색박스 구현

### 3. 구현 결과 - 메인 (section0)

```
<main>
  <section class="main-title">
    <div class="page-container2">
      <div class="main-title-image">
        
      </div>
      <h1>
        <span class="br">BR</span>
        <span class="not-br">Architects</span>
      </h1>
    </div>
  </section>
```

main › section › img

h1 › span.br

h1 › span.not-br

```
/* 상단 이미지 */
.page-container2 {
  margin: 0 auto;
  max-width: 1580px;
  position: relative;
}

.page-container2 h1 {
  color: ■■■ #fff;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  letter-spacing: 3px;
}

.page-container2 h1 .br {
  font-weight: bold;
  padding: 10px 15px;
  background-color: ■■ ■■ rgba(0, 0, 0, .8);
}

section.main-title {
  width: 100%
}

section .main-title-image img {
  min-width: 100%;
  max-width: 100%;
}
```



- 타이틀 이미지는 이후 영역보다 width가 길어 별도 container로 작성
- h1을 이미지 중간에 놓기 위해 position: absolute 활용
- 이미지가 브라우저 크기에 맞도록 min, max-width 적용

### 3. 구현 결과 - 메인 (section1)

```
<section class="main-project" id="project">
  <div class="page-container">
    <div class="inner">
      <div class="section-name">
        <h2>Projects</h2>
      </div>
      <hr>
      <div class="section-content">
        <ul class="project-list">
          <li class="list-item">
            <div class="item-name">
              Renovated
            </div>
            <div class="item-image">
              
            </div>
          </li>
          <li class="list-item">
            <div class="item-name">
              Barn House
            </div>
            <div class="item-image">
              
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>
</section>
```

main > section > h2

ul > li ...

```
/* Project 영역 */
.main-project hr {
  margin-bottom: 40px;
}

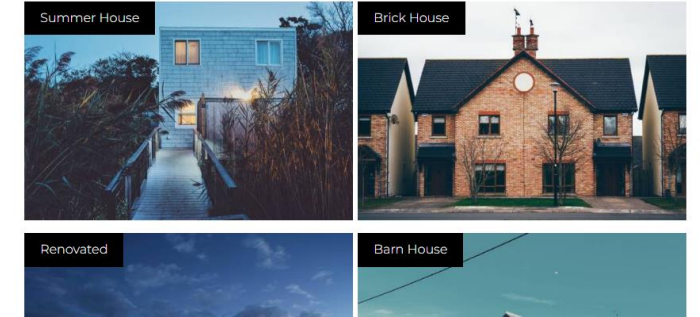
.main-project li {
  margin-bottom: 15px;
}

.main-project .project-list img {
  max-width: 100%;
  min-width: 100%;
}

.main-project .project-list .list-item {
  position: relative;
}

.main-project .project-list .item-name {
  padding: 12px 20px;
  background-color: #000;
  color: #fff;
  text-align: center;
  position: absolute;
  top: 0;
  left: 0;
}
```

#### Projects



- 이 부분의 list item에는 최소한의 CSS만 적용하고 이후 미디어쿼리에 flex 등 추가 서식 작성함

### 3. 구현 결과 - 메인 (section2)

```
<section class="main-about" id="about">
  <div class="page-container">
    <div class="inner">
      <div class="section-name">
        <h2>About</h2>
      </div>
      <hr>
      <div class="section-text">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do e
      </div>
      <div class="section-content">
        <div class="figure-item">
          <div class="figure-image">
            
          </div>
          <div class="figure-text">
            <div class="figure-name">John Doe</div>
            <div class="figure-department">CEO & Founder</div>
            <div class="figure-tmi">
              <p>Phasellus eget enim eu lectus faucibus vestibulum. Suspendisse sodales pellentesque elementum.
            </div>
            <div class="contact-button"><a href="#">Contact</a></div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

main > section > div(container) > div(items)

```
/* About 영역 */
.main-about .section-content {
  width: 100%;
}

.main-about .section-text {
  margin-bottom: 50px;
}

.main-about .figure-item {
  padding-bottom: 30px;
}

.main-about .figure-item img {
  max-width: 100%;
  min-width: 100%;
}

.main-about .figure-item .figure-name {
  font-size: 1.5rem;
  font-weight: bold;
  margin: 20px 0;
}

.main-about .figure-item .figure-department {
  font-size: 15px;
  margin: 20px 0;
}

.main-about .figure-item .figure-tmi {
  line-height: 1.5rem;
}

.main-about .figure-item .contact-button {
  text-align: center;
  font-weight: bold;
  padding: 8px 16px;
  background-color: #f1f1f1;
}
```

#### About

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



John Doe

CEO & Founder

Phasellus eget enim eu lectus faucibus vestibulum. Suspendisse sodales pellentesque elementum.

Contact



Jane Doe

Architect

Phasellus eget enim eu lectus faucibus vestibulum. Suspendisse sodales pellentesque elementum.

Contact

- ul, li로 구현한 section1 부분과 다르게 section2는 일반 div로 구현 (실험차)



### 3. 구현 결과 - 메인 (section3)

```
<section class="main-contact" id="contact">
  <div class="page-container">
    <div class="inner">
      <div class="section-name">
        <h2>Contact</h2>
      </div>
      <hr>
      <div class="section-content">
        <p>Lets get in touch and talk about your next project.</p>
        <form action="/test.html">
          <input type="text" placeholder="Name" class="input-item" required>
          <input type="email" placeholder="Email" class="input-item" required>
          <input type="text" placeholder="Subject" class="input-item" required>
          <input type="text" placeholder="Comment" class="input-item" required>
          <button class="project-contact">SEND MESSAGE</button>
        </form>
      </div>
    </div>
  </div>
</section>
```

main > section > h2

form > input

#### Contact

Lets get in touch and talk about your next project.

```
/* Contact 영역 */
.main-contact form {
  margin-bottom: 20px;
}

.main-contact form input {
  width: 100%;
  border: 1px solid #ccc;
  padding: 10px 10px;
  margin-bottom: 15px;
}

.main-contact form .project-contact {
  margin-top: 30px;
  padding: 10px 15px;
  border: none;
  background-color: #000;
  color: #fff;
}

.main-contact form .project-contact:hover {
  background-color: #ccc;
  color: #000;
}
```

- placeholder로 내부에 글자 작성 + border와 달라붙지 않게 padding 적용
- SEND MESSAGE 버튼에 hover 효과가 있어 적용

### 3. 구현 결과 - 메인 (section4)

```
<section class="main-footer">
  <div class="page-container">
    <div class="inner">
      
    </div>
  </div>
</section>
```

main > section > img

```
/* 메인 하단 영역 */
.main-footer img {
  max-width: 100%;
  min-width: 100%;
}
```



- 타이틀 이미지와 같이 브라우저 크기에 맞추어 이미지 크기가 변화하도록 적용

### 3. 구현 결과 - 푸터

powered by [w3.css](#)

```
<footer>  
  <div>powered by <span><a href="#">w3.css</a></span></div>  
</footer>
```

footer > a

```
/* ----푸터 영역 */  
footer {  
  height: 80px;  
  text-align: center;  
  line-height: 80px;  
  color: #fff;  
  background-color: #000;  
}  
  
footer a {  
  text-decoration: underline;  
}
```

- 타이포 하이퍼링크 적용

### 3. 구현 결과 - 푸터

```
@media only screen and (max-width: 600px) {
  header ul.nav {
    display: none;
  }

  .page-container2 h1 .not-br {
    display: none;
  }
}

@media only screen and (min-width: 601px) and (max-width: 992px) {
  .main-project .project-list {
    display: flex;
    align-content: space-between;
    flex-wrap: wrap;
  }

  .main-project .list-item {
    min-width: 250px;
    max-width: 47.88%;
    margin: 0 0.4em 12px 0;
  }

  .main-about .figure-content {
    display: flex;
  }

  .main-about .figure-item {
    display: inline-block;
    max-width: 47.61%;
    margin: 0 0.3em 12px 0;
  }
}
```

```
@media only screen and (min-width: 993px) {
  .main-project .project-list {
    display: flex;
    align-content: space-between;
    flex-wrap: wrap;
  }

  .main-project .list-item {
    /* min-width: 213px; */
    max-width: 24.5%;
    margin: 0 0.4em 12px 0;
  }

  .main-about .figure-content {
    display: flex;
  }

  .main-about .figure-item {
    display: inline-block;
    max-width: 24.2%;
    margin: 0 0.3em 12px 0;
  }
}
```

- 영역별 미디어쿼리 적용

## 4. 소요 시간 및 후기

### 소요시간

- 코딩 4:00~ 7:30 (3시간 30분)

### 후기

- 이번에는 요소별로 구역을 세세하게 구분하는 데 신경 썼는데 생각보다 간단하지 않다는 생각이 들었습니다.
- 특히 문서 전체에 margin, padding이 들쭉날쭉하지 않고 일정하게 적용되어야 하는데 이렇게 공통되는 영역을 뽑아내기가 까다로운 것 같아요.
- 미디어 쿼리는 최대한 구현하려고 했는데 사실 제대로 작동하지 않습니다. 이미지 가로 2줄, 세로 4줄 배열을 딱딱 지키는 게 아니라 3줄이 되기도 하고, 옆에 여백이 남아 일그러지기도 합니다.
- 이 부분은 도저히 해결이 어려워서 과제제출 후 원본 소스코드를 뜯어보려고 합니다. 지금은 미디어 쿼리 적용이 제일 어려운 듯 하네요!