

Contact

Lets get touch and talk about your next project.

Name

Email

Subject

Comment

SEND MESSAGE

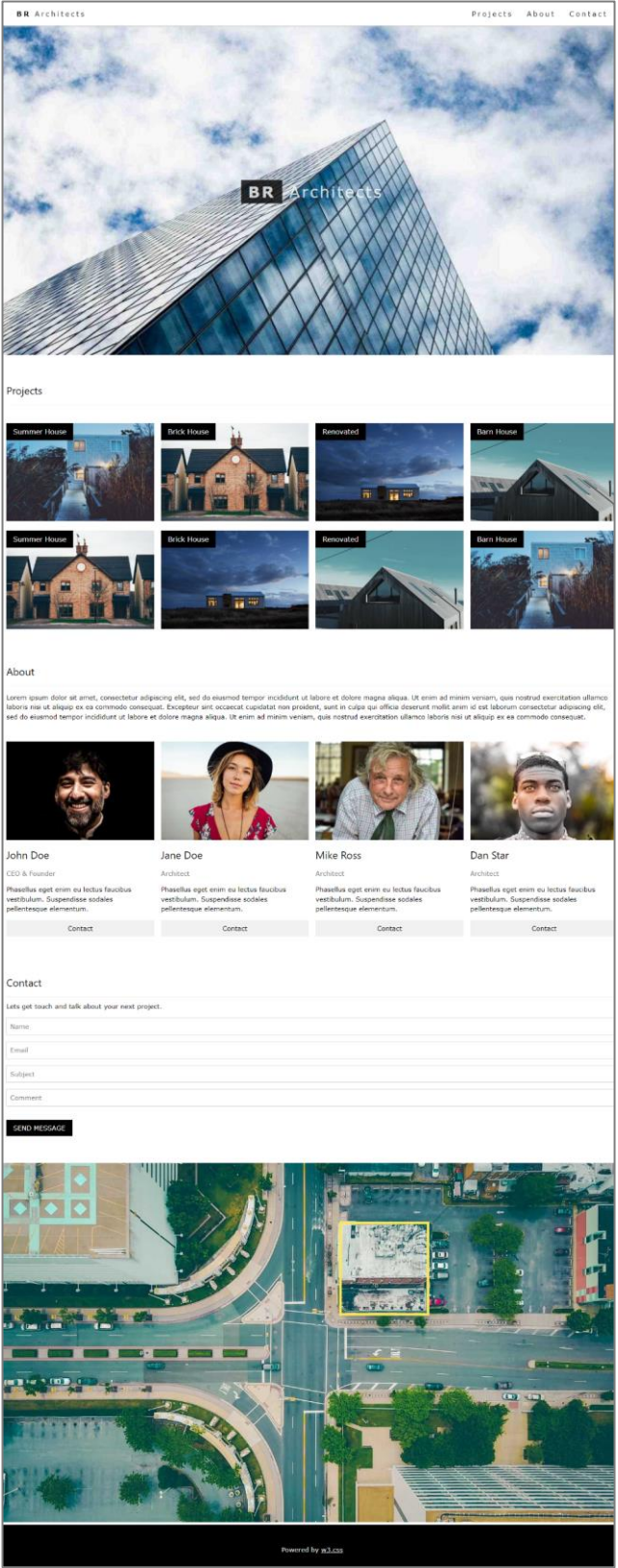


클론코딩 - 02

Architects hompage

서보선

1. 최종 결과물



네비게이션 메뉴
배너

메인 콘텐츠 1

메인 콘텐츠 2

Contact Form

Map

Footer

2. DOM Tree

HTML Console 창에 DOM Tree를 출력하기 위한 코드



```
1 <script type="module">
2     import exploreDom from './js/Dom.js';
3
4     const body = document.querySelector('body');
5     const i = exploreDom(body);
6     console.log(i);
7 </script>
```

// HTML 본문의 body(최상위) element를 참조한 모듈 함수에 넣어준다

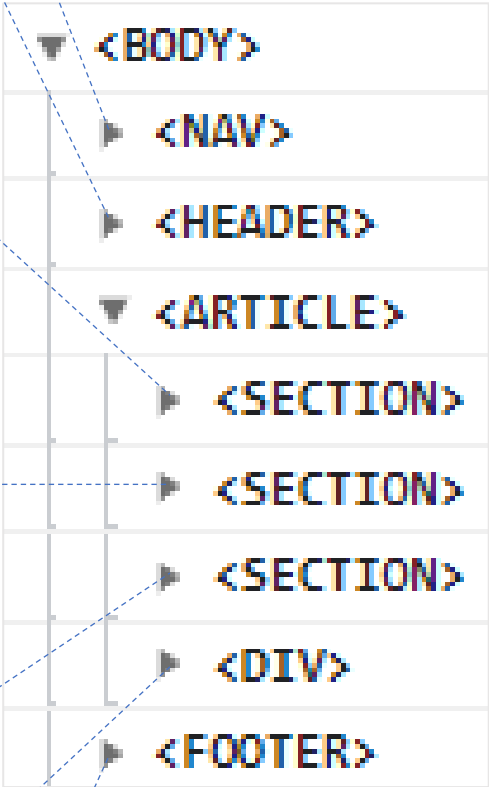
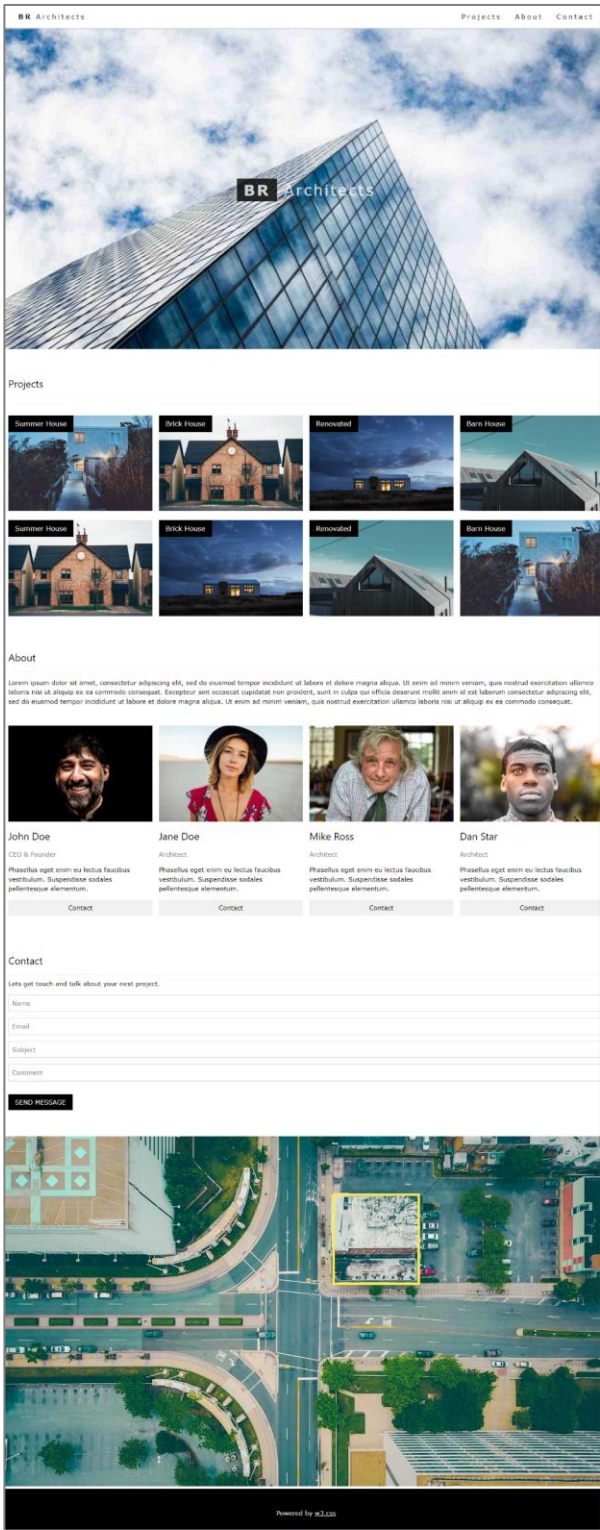


```
1 let lineNumber = 0;
2
3 function exploreDom(ancestor) {
4     lineNumber++;
5     if (ancestor.children) {
6         console.group(`<${ancestor.tagName}>`);
7         Array.from(ancestor.children).forEach(v => {
8             exploreDom(v);
9         });
10        console.groupEnd();
11    } else {
12        console.log(ancestor.tagName);
13        return lineNumber;
14    }
15
16    return lineNumber;
17 }
18
19 export default exploreDom;
```

// children -> 자식 요소가 있다면 console.group() 후 재귀(하단 깊이 재탐색)

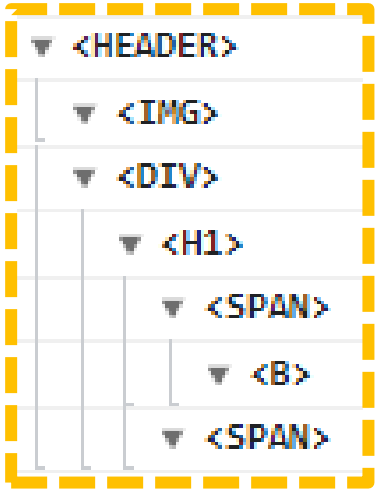
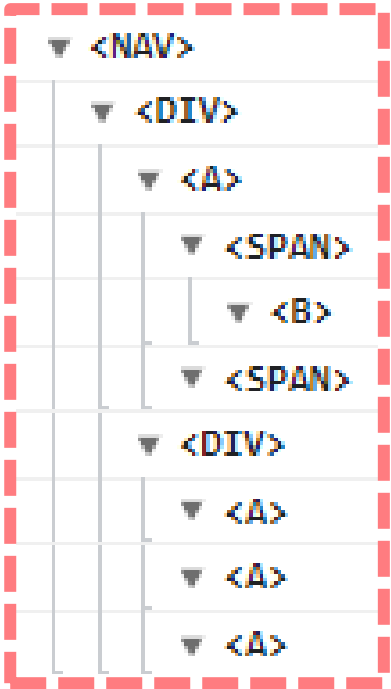
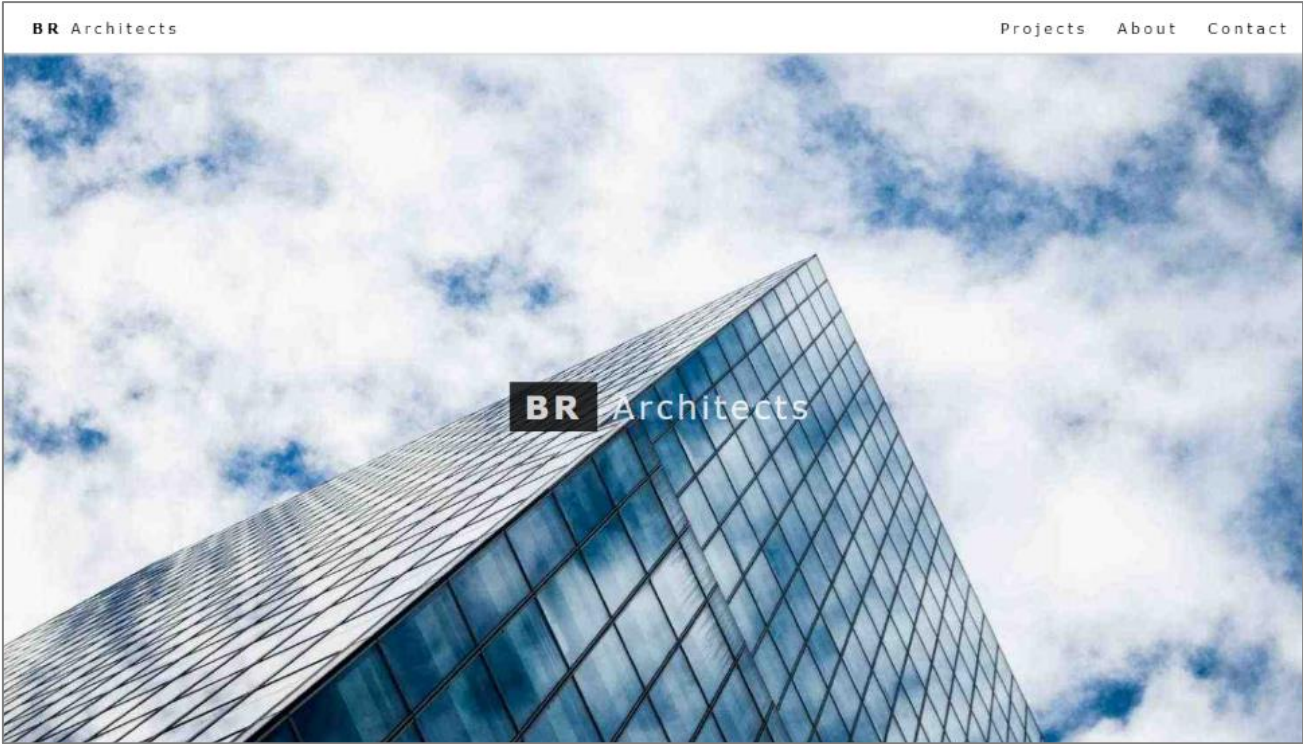
2. DOM Tree

전체 상위 구조



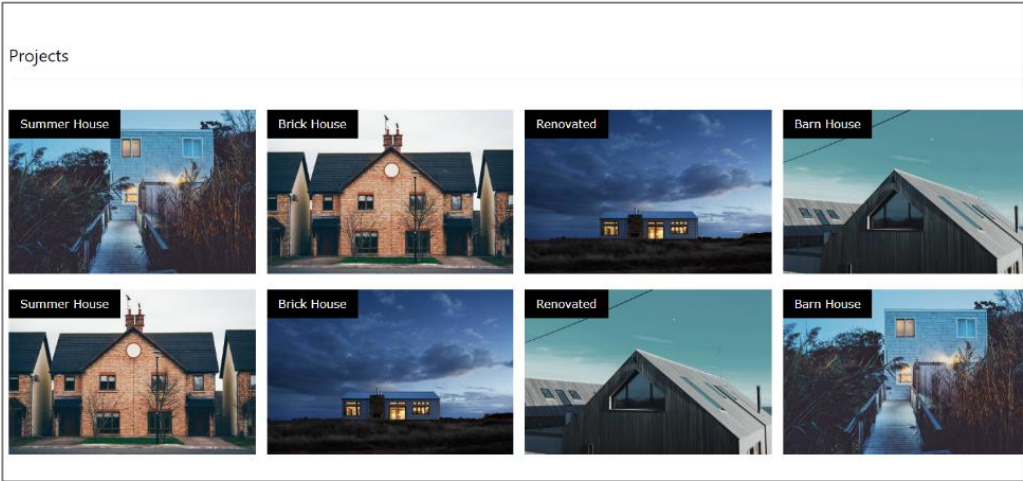
2. DOM Tree

하위 구조 - <nav>, <header>



2. DOM Tree

하위 구조 - <article> - <section> - 1




	▼	<ARTICLE>
	▼	<SECTION>
	▼	<DIV>
	▼	<H3>
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	
	▼	<DIV>
	▼	
	▼	

2. DOM Tree

하위 구조 - <article> - <section> - 2

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




Mike Ross

Architect

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

Contact



Dan Star

Architect

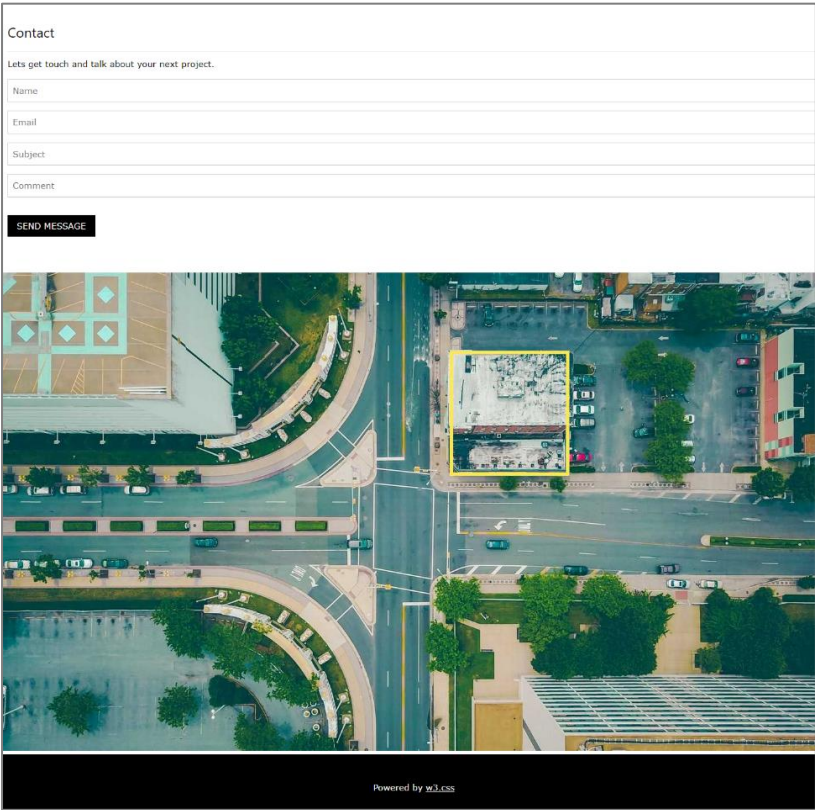
Phasellus eget enim eu lectus faucibus vestibulum. Suspendisse sodales pellentesque elementum.

Contact

▼	<SECTION>
▼	<DIV>
▼	<H3>
▼	<P>
▼	<DIV>
▼	<DIV>
▼	
▼	<H3>
▼	<P>
▼	<P>
▼	<P>
▼	<BUTTON>
▼	<DIV>
▼	
▼	<H3>
▼	<P>
▼	<P>
▼	<P>
▼	<BUTTON>
▼	<DIV>
▼	
▼	<H3>
▼	<P>
▼	<P>
▼	<P>
▼	<BUTTON>
▼	<DIV>
▼	
▼	<H3>
▼	<P>
▼	<P>
▼	<P>
▼	<BUTTON>

2. DOM Tree

하위 구조 - <article> - <section> - 3, map , <footer>



	▼	<SECTION>
	▼	<DIV>
	▼	<H3>
	▼	<P>
	▼	<FORM>
	▼	<P>
	▼	<INPUT>
	▼	<P>
	▼	<INPUT>
	▼	<P>
	▼	<INPUT>
	▼	<P>
	▼	<INPUT>
	▼	<P>
	▼	<BUTTON>
	▼	<DIV>
	▼	
	▼	<FOOTER>
	▼	<P>
	▼	<A>

3. CSS 공통, Media Query

// 공통 속성 및 초기화

```
1  * {
2    margin: 0;
3    padding: 0;
4    line-height: 1.5;
5    font-family: Verdana,sans-serif;
6  }
7
8  body {
9    width: 100%;
10 }
11
12 body * {
13   overflow: hidden;
14   box-sizing: border-box;
15 }
16
17 h1 {
18   font-size: 36px;
19   font-weight: 400;
20   letter-spacing: 4px;
21   color: rgba(255, 255, 255, .8);
22 }
23
24 h3 {
25   font-size: 24px;
26   font-family: "Segoe UI",Arial,sans-serif;
27   font-weight: 400;
28   margin: 10px 0;
29 }
30
31 p {
32   font-size: 15px;
33 }
34
35 a {
36   text-decoration: none;
37   color: black;
38 }
39
40 button, input {
41   font-size: 15px;
42 }
43
44 a:hover,
45 button:hover {
46   cursor: pointer;
```

// 반응형 웹을 위한 Media Query

```
1  @media screen and (max-width: 1000px){
2    .main-contents {
3      width: 1000px;
4      max-width: 100%;
5    }
6
7    .projects__list,
8    .about__list__item {
9      width: 50%;
10   }
11 }
12
13 @media screen and (max-width: 600px){
14   .main-contents {
15     width: 600px;
16     max-width: 100%;
17   }
18
19   .nav__top__inner-links {
20     display: none;
21   }
22
23   .projects__list,
24   .about__list__item {
25     width: 100%;
26   }
27 }
```

/ 미디어 쿼리 */**

브라우저의 크기 별로

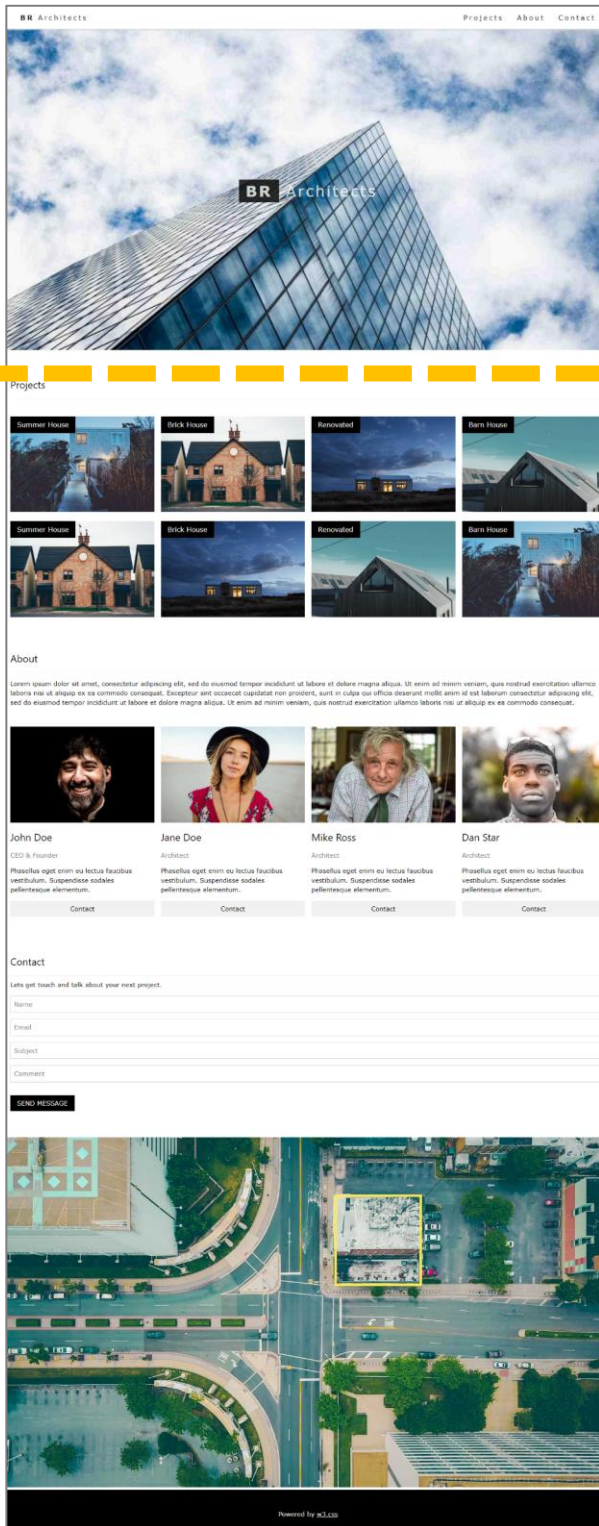
본문의 정렬 형태를 변경해주어야 하므로

float 된 요소들의 너비를

브라우저 크기 별 행에 맞도록

조정해준다.

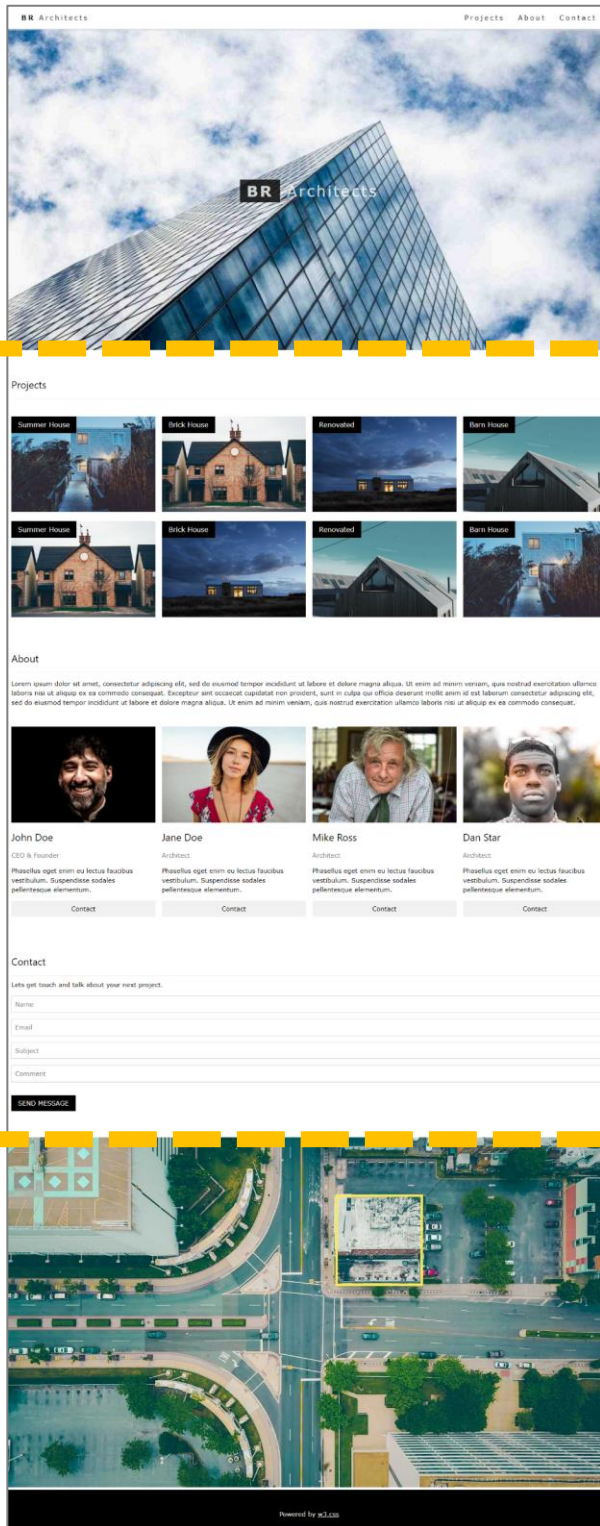
3. CSS <nav>, <header>



```
1  /** Navigation */
2  /* position과 z-index를 통한 고정 플로팅 처리 */
3  .nav {
4      width: 100%;
5      height: 55px;
6      position: fixed;
7      top: 0;
8      left: 0;
9      z-index: 9999;
10 }
11
12 /* <a> 태그 마우스 호버 배경색 변경 */
13 .nav_top a:hover {
14     background-color: #d5d5d5;
15 }
16
17 /* 정렬에 사용된 float 해제 ::after */
18 .nav_top_inner-links:last-child::after {
19     content: '';
20     display: block;
21     clear: both;
22     float: none;
23 }
24
25 /* 정렬 float */
26 .nav_top_inner-links a {
27     display: block;
28     float: left;
29 }
```

```
1  /** Header */
2  /* 로고 검치기 용 position: absolute */
3  .header {
4      position: relative;
5  }
6
7  .header_title {
8      position: absolute;
9      top: 50%;
10     left: 50%;
11     transform: translate(-50%, -50%);
12 }
13
```

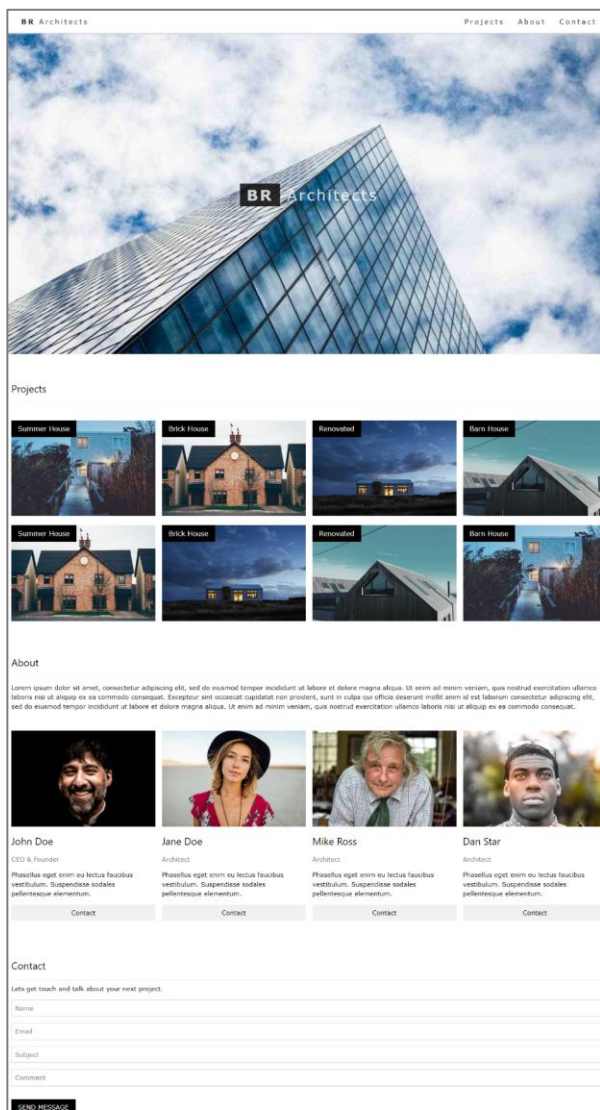
3. CSS <article>



```
1  /* Projects */
2  /* float 정렬 처리 */
3  .projects::after {
4      content: '';
5      display: block;
6      clear: both;
7      float: none;
8  }
9
10 /* span 겹치기 처리 */
11 .projects_list {
12     position: relative;
13     float: left;
14 }
15
16 .projects_list> span {
17     display: block;
18     position: absolute;
19     top: 0;
20     left: 8;
21 }
```

```
1  /* About */
2  /* float 해제 */
3  .about_list::after {
4      content: '';
5      display: block;
6      clear: both;
7      float: none;
8  }
9
10 /* 정렬 및 크기 조정(Media query) */
11 .about_list_item {
12     width: 25%;
13     float: left;
14 }
15
16 .about_list_item > p {
17     display: block;
18 }
19
20 .about_list_item > p:first-of-type {
21     color: gray;
22 }
```


3. CSS <footer>



```

1  /* map 이미지 크기 */
2  map img:last-of-type {
3      width: 100%;
4      padding: 0 8px;
5  }
6
7  /** Footer */
8  .footer {
9      width: 100%;
10 }
11
12 .footer p {
13     display: block;
14     width: 100%;
15     text-align: center;
16 }
17
18 .footer p a {
19     text-decoration: underline;
20     color: #fff;
21 }
    
```



4. JS

```
1 <script type="module">
2   import RegexHelper from './js/helper/RegexHelper.js';
3
4   const name = document.querySelector('input[placeholder="Name"]');
5   const email = document.querySelector('input[placeholder="Email"]');
6
7   document.querySelector('#contact__submit').addEventListener('click', e => {
8     e.preventDefault();
9
10    const regexHelper = RegexHelper.getInstance();
11
12    try {
13      regexHelper.value(name, "Please input your name.");
14      regexHelper.engKor(name, "Please check your name is correct format(can only use English or Korean)");
15      regexHelper.minLength(name, 2, "The name is too short(needs more than two letters).");
16
17      regexHelper.value(email, "Please input your E-mail.");
18      regexHelper.email(email, "Please check E-mail address is correct format.");
19    } catch(err) {
20      alert(err.message);
21      console.error(err);
22      err.selector.focus();
23      return;
24    }
25  });
26 </script>
27
28 <!-- focus out 시키기 실패 -->
29 <!-- <script type="module">
30   document.querySelectorAll('.link__target').addEventListener('focus', e => {
31     e.currentTarget.blur();
32   });
33 </script> -->
34
35 <script type="module">
36   import exploreDom from './js/Dom.js';
37
38   const body = document.querySelector('body');
39   console.log(exploreDom(body));
40 </script>
```

// 사용한 JS 코드, Dom.js는 앞에서 소개, RegexHelper.js를 이용 form 입력값 검사

```
1 class BadRequestException extends Error {
2   #statusCode; // Error Code
3   #selector; // 입력 요소에 대한 selector(element를 찾기 위한 추가요소)
4
5   constructor(msg = "잘못된 요청입니다.", selector = null) {
6     super(msg);
7     super.name = "BadRequestException";
8     this.#statusCode = 400;
9     this.#selector = selector;
10  }
11
12  get statusCode() {
13    return this.#statusCode;
14  }
15
16  get selector() {
17    return this.#selector;
18  }
19 }
20
21 export default BadRequestException;
```

// ExceptionHelper

```
1 class RegexHelper {
2   /** 싱글톤 */
3   /** 싱글톤은 스테틱을 통해 메모리 한계내에 존재하는 객체
4   static #current = null;
5
6   static getInstance() {
7     if (RegexHelper.#current === null) {
8       RegexHelper.#current = new RegexHelper();
9     }
10    return RegexHelper.#current;
11  }
12
13  /**
14   * @param (HTMLElement) field - 검사할 대상에 대한 요소의 DOM Object(Element)
15   * @param (string) msg - 검사 실패 경우 출력할 메시지
16   */
17  value(field, msg) {
18    const content = field.value;
19
20    if (content === undefined || content === null || (typeof content === "string" && content.trim().length === 0)) {
21      throw new BadRequestException(msg, field);
22    }
23    return true;
24  }
25 }
```

// Regex 대표 메소드

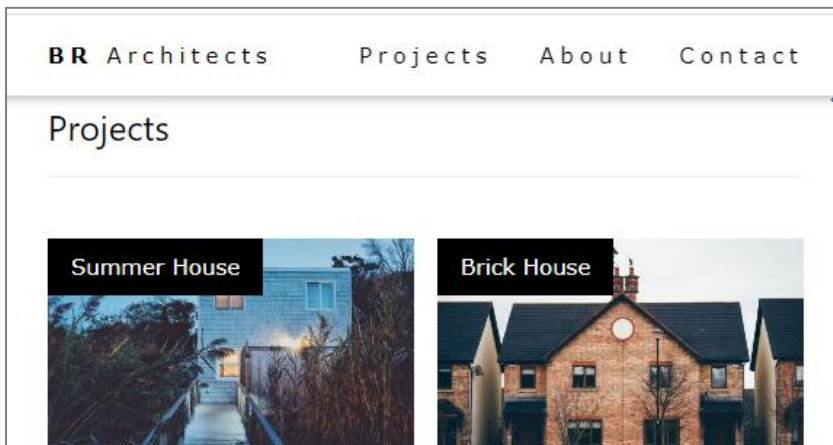
5. 작업 중 에러 처리

1. 네비게이션 내부 링크 클릭 시

이동한 위치의 **콘텐츠 최상단이 <nav>에 가려지는 현상 발생**

-> 콘텐츠 시작 위치의 상단에

<nav>의 height만큼 margin + padding 처리



2. Form 입력 시 **값이 없음에도 전송 가능** 및 submit을 통한 **페이지 이탈**

-> **<input>의 required property와 minLength attribut로**

1차 입력값 검증 수행,

Js 모듈을 활용해 element select 및 검증 수행

Contact

Lets get in touch and talk about your next project.

Name

Email

Subject

Comment

SEND MESSAGE

이 입력란을 작성하세요.

6. 작업 소요 시간 및 소감

총 작업 소요 시간:

4시간

부모와 자식 간의 관계를 잘 파악하여
HTML 및 CSS를 정의하는 연습이 되었습니다.

margin과 padding 등의 기본적인 속성도
겹침 현상이나 출력 위치에 따라
의도와 다른 결과를 출력 가능함을 알았습니다.

반응형 웹을 위한 구조 또한
브라우저 환경이 어떠한 요소를 제어해야 할지
명확히 연습 가능한 시간이었습니다.

감사합니다

↓ 결과물 웹 주소

http://sqhtjs0104.dothome.co.kr/architects_clone/