

클론코딩 05

Portfolio homepage

서보선

1. 최종 결과물 데스크톱

사이드 메뉴

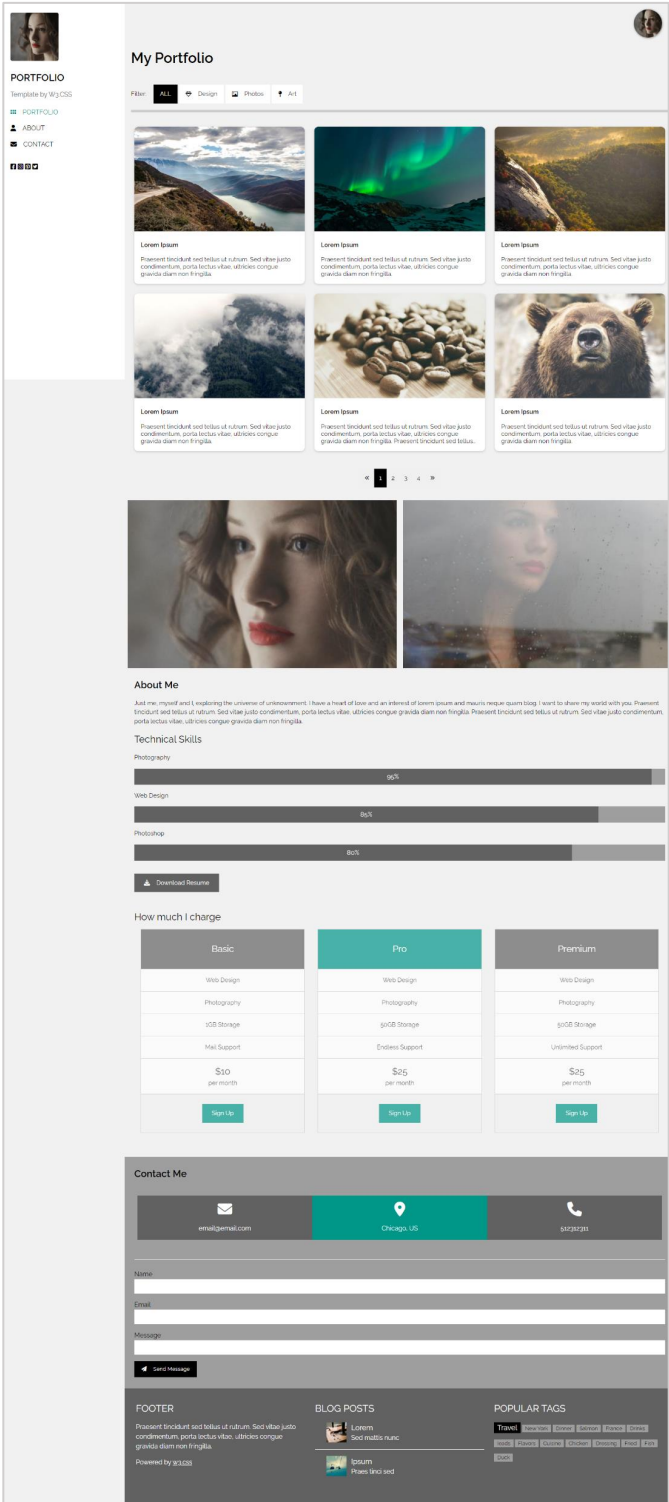
탑 메뉴, 헤더

메인 콘텐츠 1

메인 콘텐츠 2

메인 콘텐츠 3

Footer



1. 최종 결과물 태블릿, 모바일



태블릿 사이즈

아이템 배치
2열 변경,

사이드 메뉴
숨김 처리



모바일 사이즈

아이템 배치
1열 변경,

사이드 메뉴
숨김 처리

2. DOM Tree

HTML Console 창에 DOM Tree를 출력하기 위한 코드(수정)

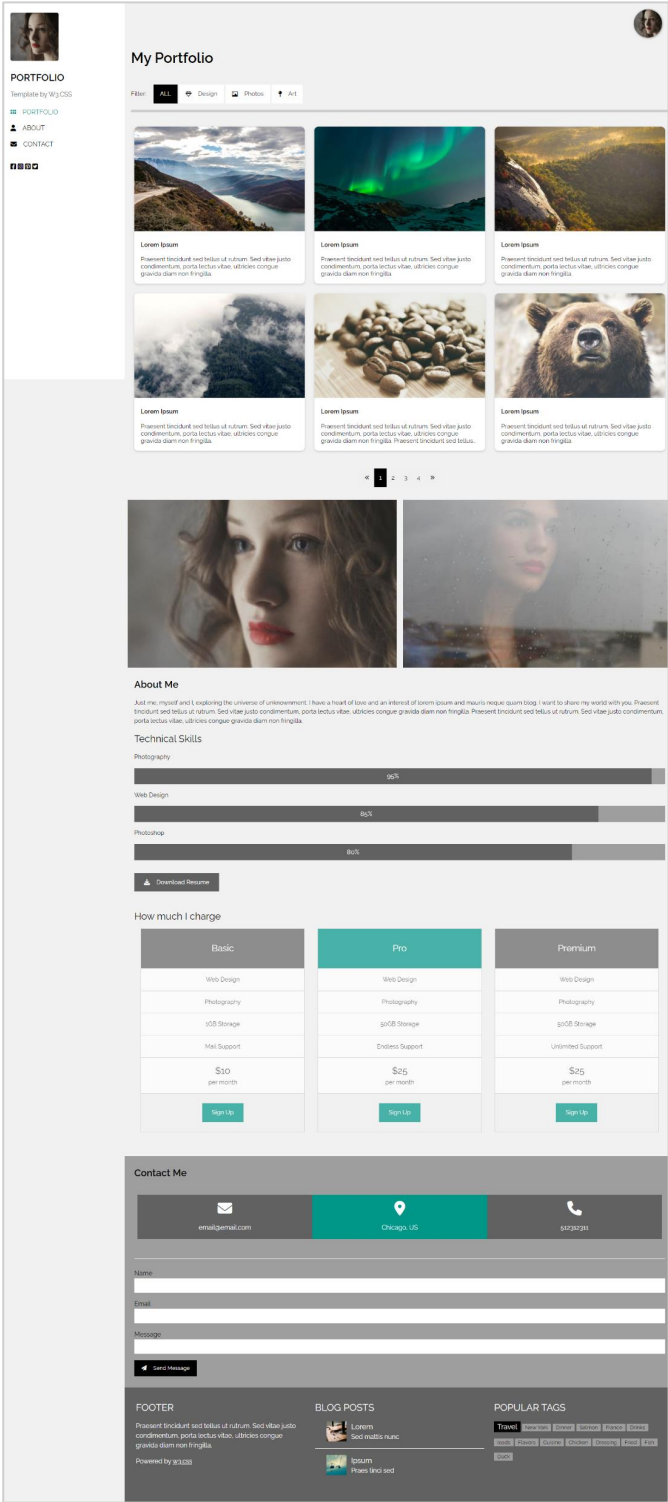
```
1 <script type="module">
2     import exploreDom from './js/Dom.js';
3
4     const body = document.querySelector('body');
5     const i = exploreDom(body);
6     console.log(i);
7 </script>
```

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

```
1 let lineNumber = 0; // 줄 콘솔 출력 라인 수
2 let dept = 0; // 각 노드 계층 단계의 깊이
3
4 function exploreDom(ancestor) {
5     lineNumber++; // 함수 호출: 무조건 한 줄 출력 -> 라인 + 1
6     dept++; // 깊이도 + 1 해준 뒤 함수 마지막에 다시 빼줌(재귀 거듭될수록 높아짐 -> 깊이)
7
8     // 자식 노드가 있다면
9     if (ancestor.children[0]) {
10         // console.group을 통해 하위 노드 처리
11         if (ancestor.classList[0]) { // 클래스가 있다면
12             console.group(`<${ancestor.tagName} class="${ancestor.classList[0]}">`); // 클래스 이름도 붙여서 태그 출력
13         } else {
14             console.group(`<${ancestor.tagName}>`); // 클래스 없으면 그냥 태그만 출력
15         }
16         console.log(`///// dept: ${dept}`);
17         // 재귀 호출
18         Array.from(ancestor.children).forEach(v => {
19             exploreDom(v);
20         });
21         console.groupEnd();
22     } else {
23         // 리프 태그 출력(클래스 포함)
24         if (ancestor.classList[0]) {
25             console.log(`<${ancestor.tagName} class="${ancestor.classList[0]}">`);
26         } else {
27             console.log(`<${ancestor.tagName}>`);
28         }
29     }
30     dept--;
31     return lineNumber; // DOM 중 깊이 반환
32 }
33
34 export default exploreDom;
```

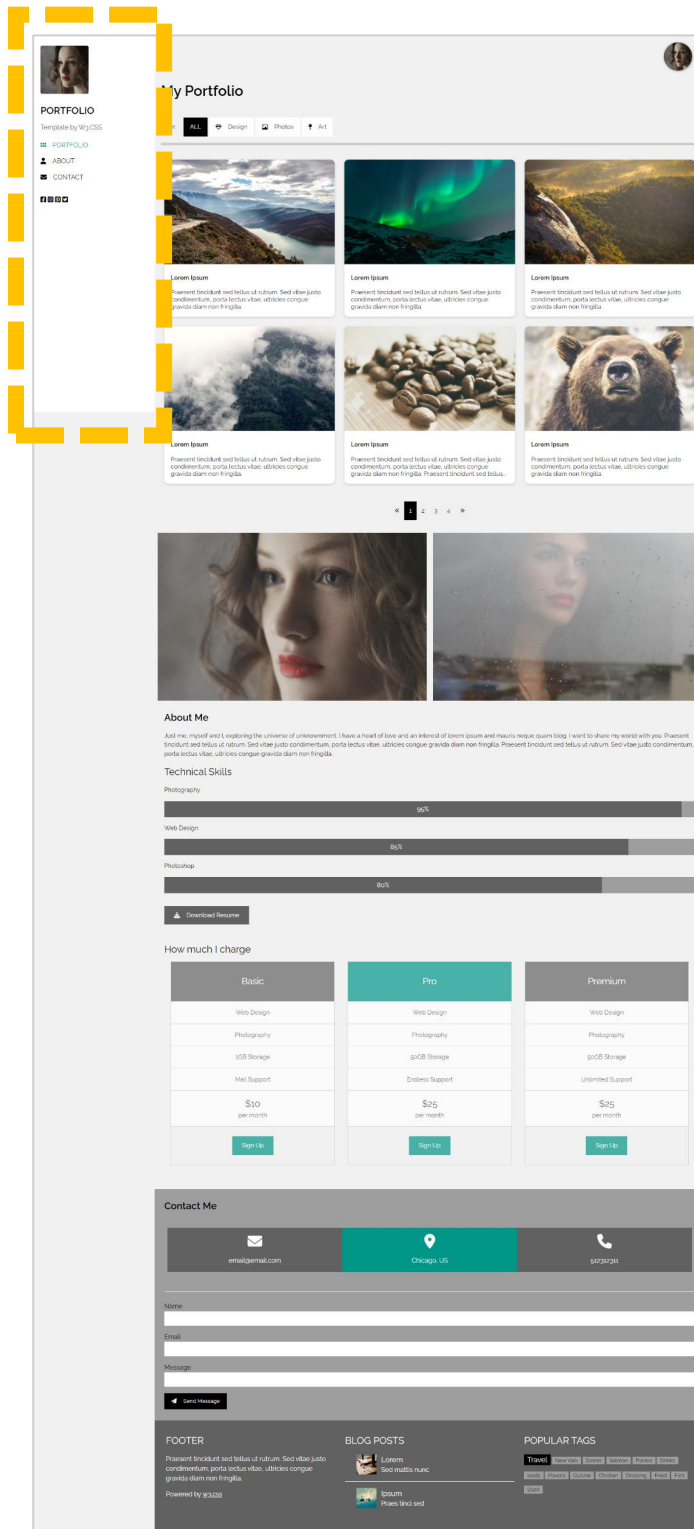
// children -> 자식 요소가 있다면 console.group() 후 재귀(하단 깊이 재탐색)

2. DOM Tree 전체 상위 구조



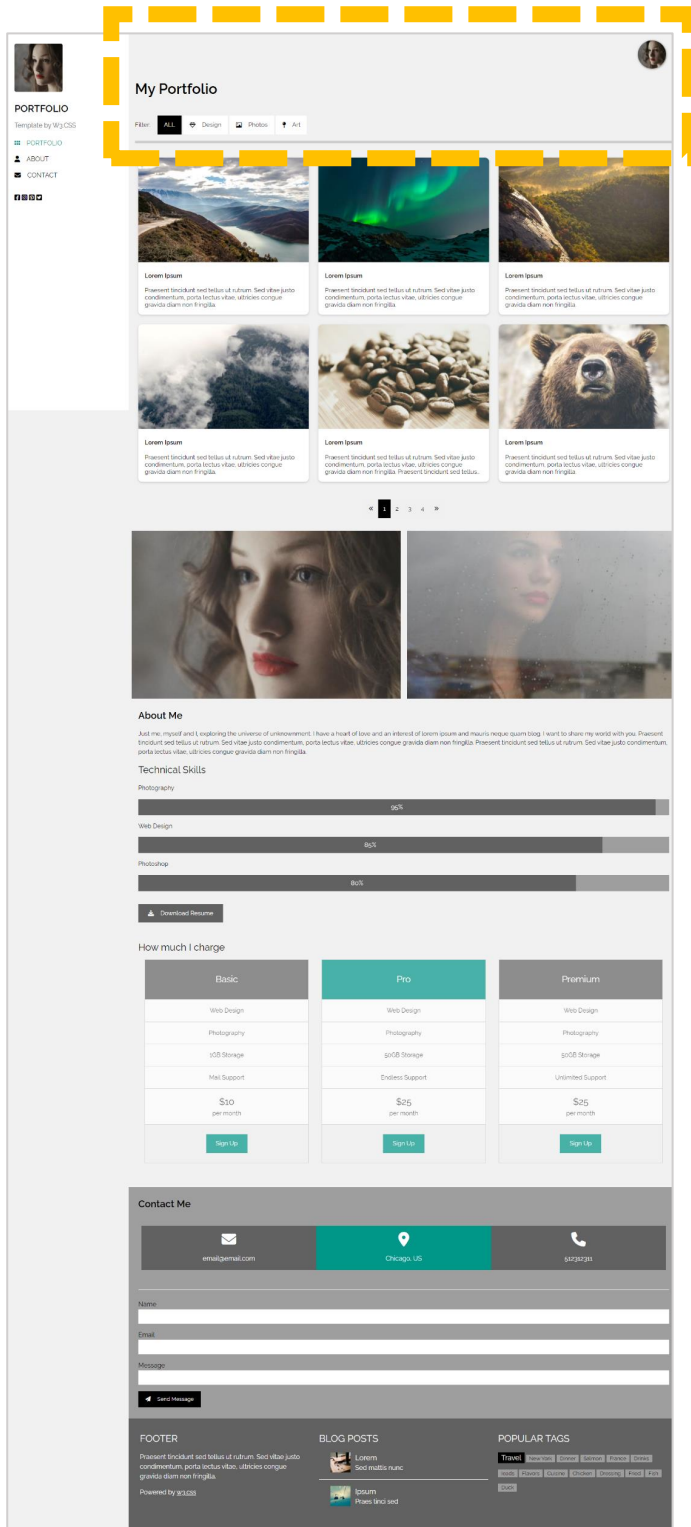
```
<BODY>
  // dept: 1
  <DIV class="side-menu">
    <DIV>
      <DIV class="main_contents">
        // dept: 2
        <HEADER class="header">
          <ARTICLE>
            // dept: 3
            <SECTION class="portfolios">
              <DIV class="portfolios_control-bar">
                <DIV class="about-me_photos">
                  <DIV class="modal">
                    <SECTION class="about-me">
                      <SECTION class="contact">
                        <FOOTER class="footer">
                          <SCRIPT>
```

2. DOM Tree Side menu



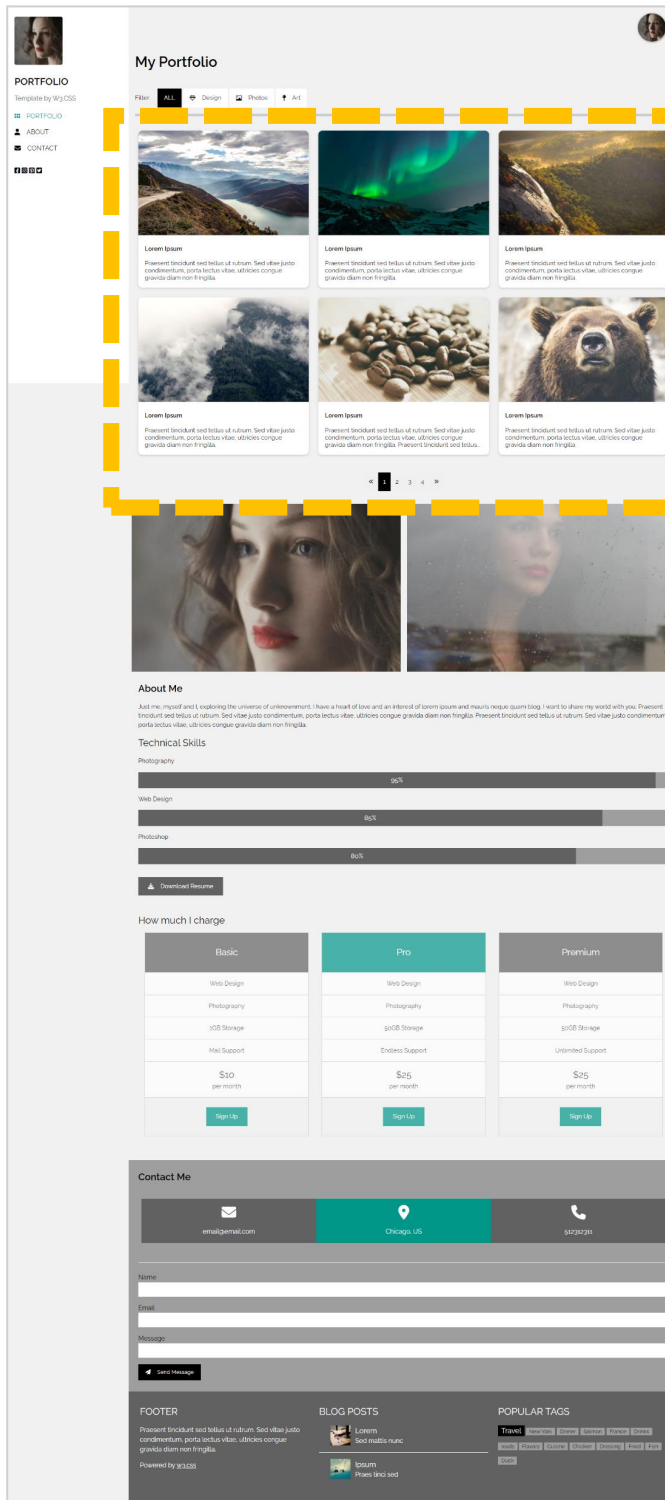
```
////// dept: 1
▼ <DIV class="side-menu">
  ////// dept: 2
  ▼ <DIV class="side-menu__content">
    ////// dept: 3
    ▼ <DIV class="side-menu__content_top">
      ////// dept: 4
      ▼ <DIV class="side-menu__content_top_avatar">
        ////// dept: 5
        <IMG>
        ▼ <DIV class="shadow-box">
          ////// dept: 6
          <I class="fa-solid">
        ▼ <BUTTON>
          ////// dept: 5
          <I class="fa-solid">
        <BR>
        ▼ <H4>
          ////// dept: 4
          <B>
          <P>
        ▼ <A>
          ////// dept: 3
          <I class="fa-solid">
        ▼ <A>
          ////// dept: 3
          <I class="fa-solid">
        ▼ <A>
          ////// dept: 3
          <I class="fa-solid">
        ▼ <DIV class="side-menu__panel">
          ////// dept: 3
          ④ <I class="fa-brands">
        <DIV class="shadow-box">
```


2. DOM Tree Header



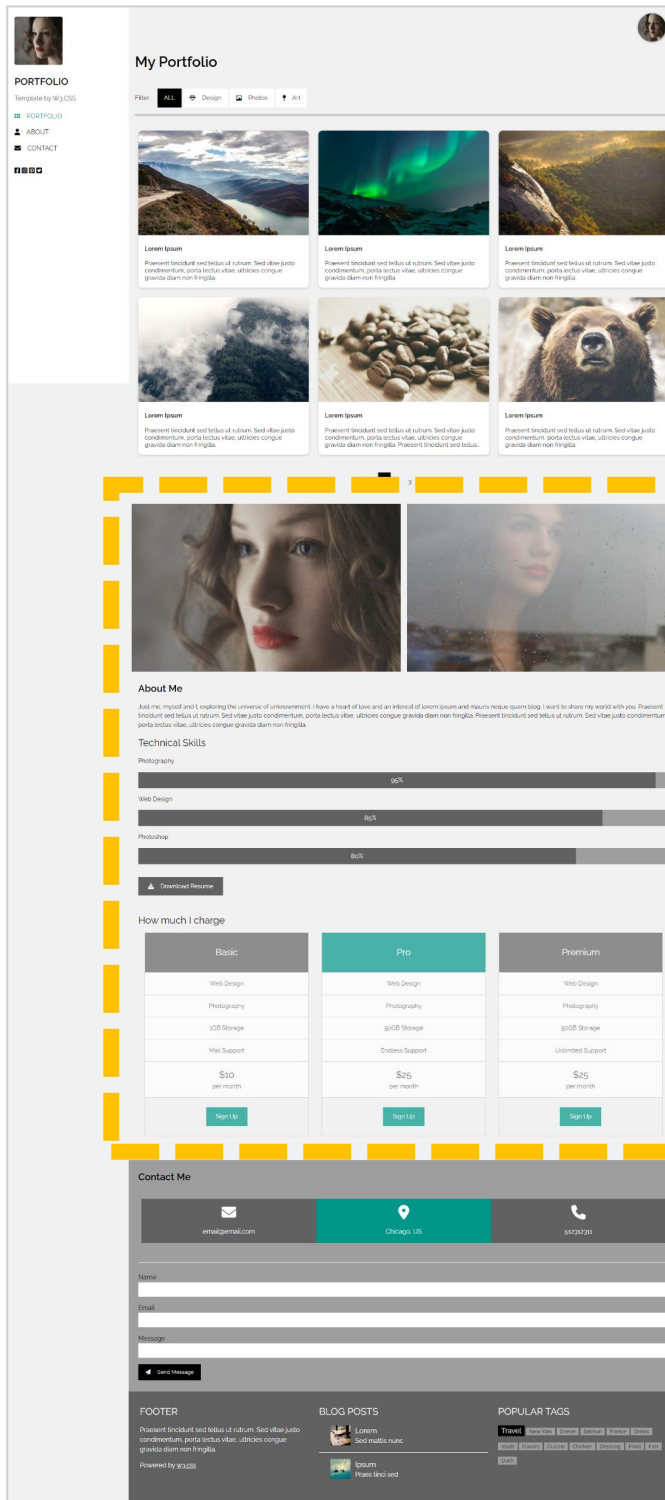
```
//// dept: 2
▼ <HEADER class="header">
  //// dept: 3
  ▼ <DIV class="header_top">
    //// dept: 4
    ▼ <SPAN class="side-menu_open">
      //// dept: 5
      <I class="fa-solid">
    ▼ <A>
      //// dept: 5
      <IMG>
    ▼ <DIV class="shadow-box">
      //// dept: 6
      <I class="fa-solid">
  ▼ <DIV class="header_contents">
    //// dept: 4
    ▼ <H1>
      //// dept: 5
      <B>
    ▼ <DIV class="header_contents_filter">
      //// dept: 5
      <SPAN>
      ▼ <UL>
        //// dept: 6
        ▼ <LI>
          //// dept: 7
          <BUTTON>
        ▼ <LI>
          //// dept: 7
          ▼ <BUTTON>
            //// dept: 8
            <I class="fa-regular">
        ▼ <LI>
          //// dept: 7
          ▼ <BUTTON>
            //// dept: 8
            <I class="fa-regular">
        ▼ <LI>
          //// dept: 7
          ▼ <BUTTON>
            //// dept: 8
            <I class="fa-solid">
```

2. DOM Tree Section 1 – Portfolio



▼	<ARTICLE>
	////// dept: 3
▼	<SECTION class="portfolios">
	////// dept: 4
▼	<DIV class="portfolios_item">
	////// dept: 5
▼	<DIV class="portfolios_item_img">
	////// dept: 6
	
▼	<DIV class="portfolios_item_content">
	////// dept: 6
▼	<P>
	////// dept: 7
	
	<P>
▶	<DIV class="portfolios_item">
▶	<DIV class="portfolios_item">
▶	<DIV class="portfolios_item">
▶	<DIV class="portfolios_item">
▶	<DIV class="portfolios_item">
▼	<DIV class="portfolios_control-bar">
	////// dept: 4
▼	<BUTTON>
	////// dept: 5
	<I class="fa-solid">
4	<BUTTON>
▼	<BUTTON>
	////// dept: 5
	<I class="fa-solid">

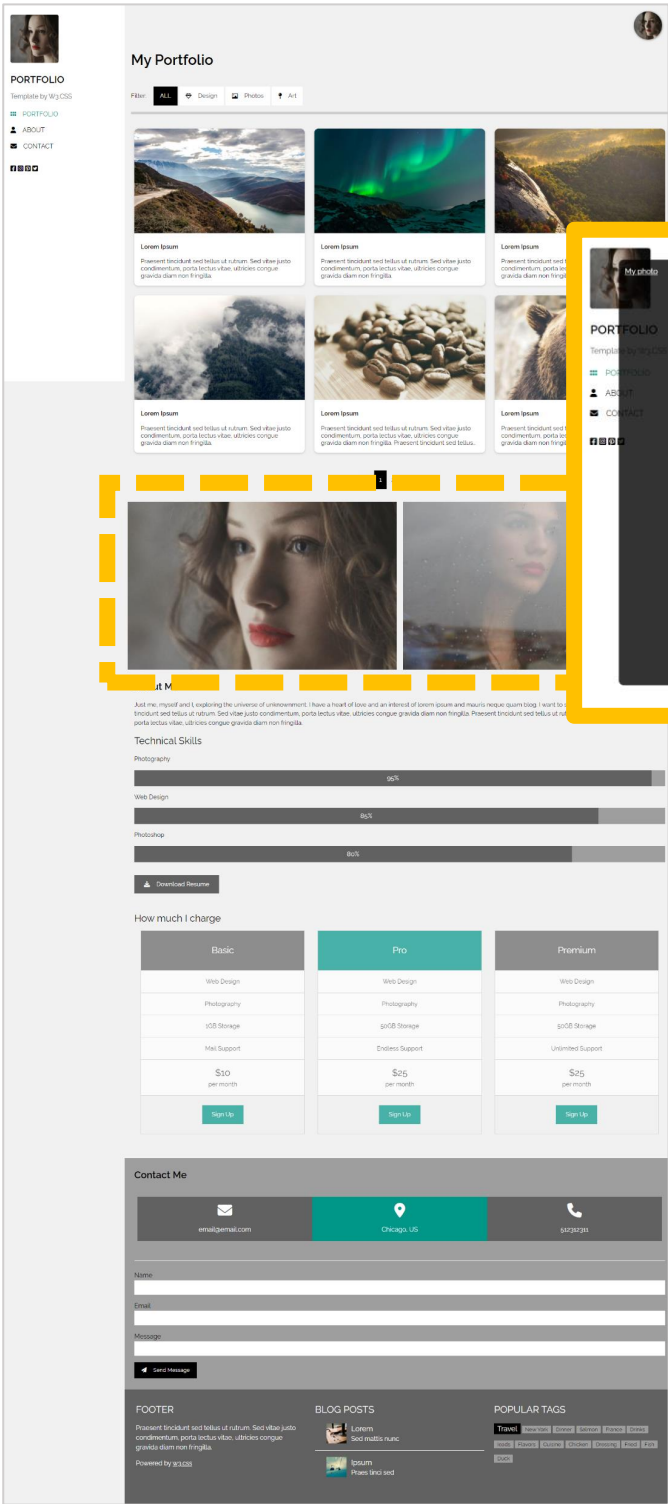
2. DOM Tree Section 2 – About me



▼	<SECTION class="about-me">
	///// dept: 4
▼	<H4>
	///// dept: 5
	
	<P>
	<H4>
	<P>
	<DIV class="proccess-bar">
	<P>
	<DIV class="proccess-bar">
	<P>
	<DIV class="proccess-bar">
	<P>
▼	<BUTTON>
	///// dept: 5
	<I class="fa-solid">
	<H4>
▼	<DIV class="about-me__charge">
	///// dept: 5
▼	<DIV class="about-me__charge__item">
	///// dept: 6
▼	
	///// dept: 7
▼	
	///// dept: 8
	<H2>
4	
▼	
	///// dept: 8
	<H2>
	
▼	
	///// dept: 8
	<BUTTON>
▶	<DIV class="about-me__charge__item">
▶	<DIV class="about-me__charge__item">

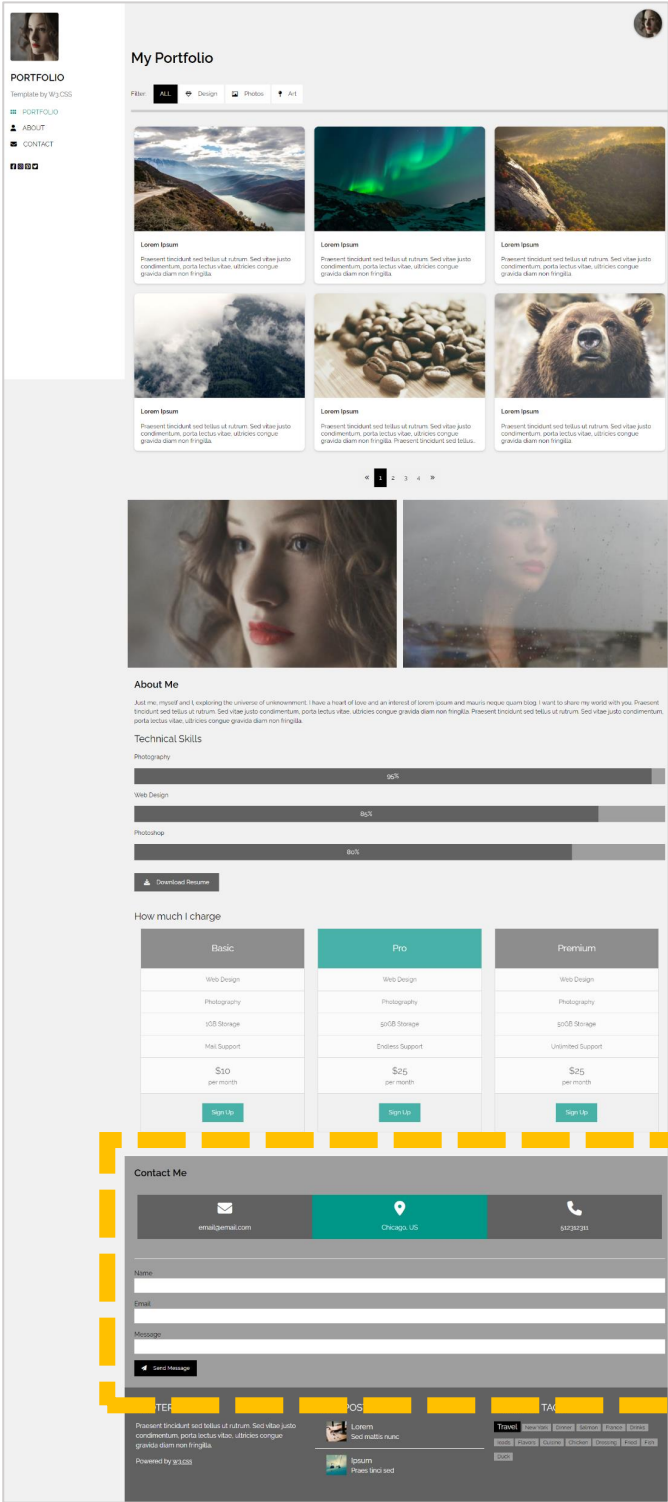
2. DOM Tree

Section 2 – About me – Modal window



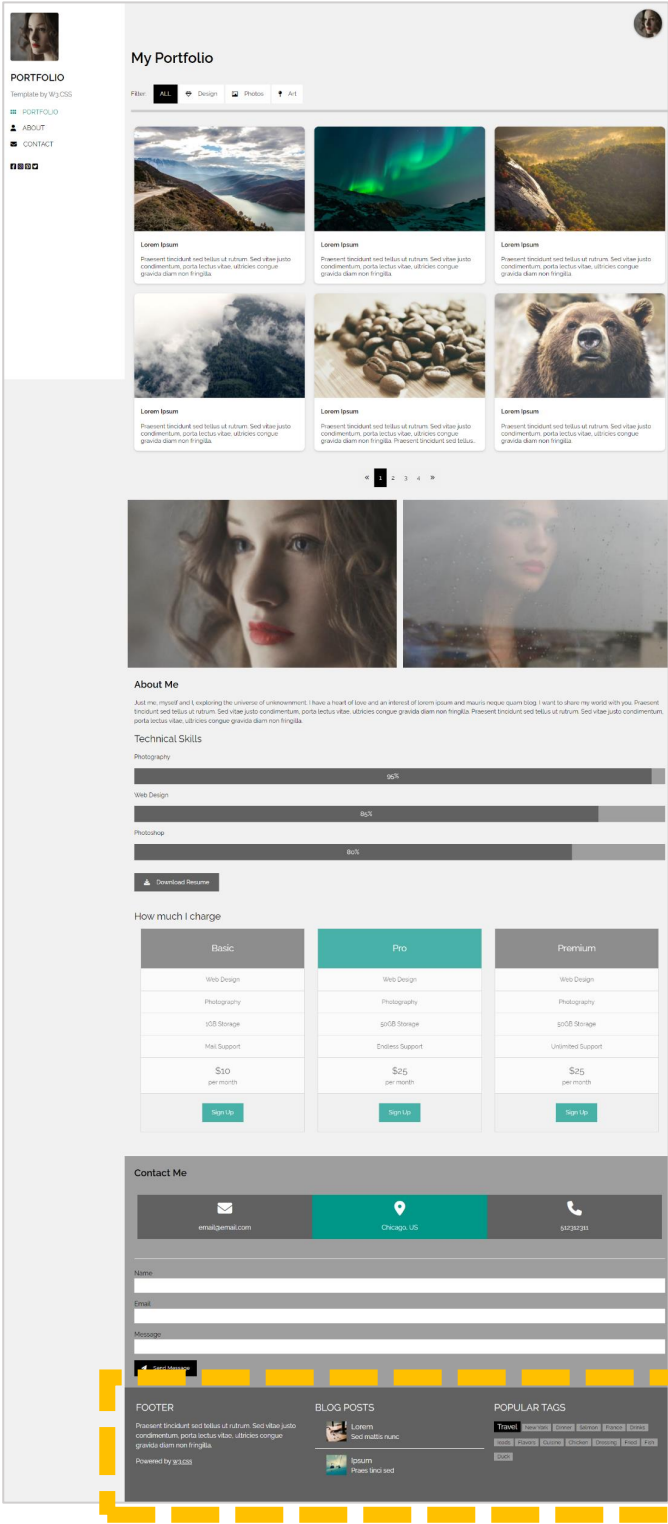
```
<div class="about-me_photos">
  // dept: 4
  <img>
  <div class="modal">
    // dept: 4
    <div class="modal_top">
      // dept: 5
      <p>
        <button>
          // dept: 6
          <i class="fa-solid">
        <div class="modal_body">
          // dept: 5
          <img>
```

2. DOM Tree Section 3 - Contact



```
<SECTION class="contact">
  // dept: 4
  <H4>
    // dept: 5
    <B>
  <DIV class="contact_ways">
    // dept: 5
    <DIV class="contact_ways_item">
      // dept: 6
      <P>
        // dept: 7
        <I class="fa-solid">
      <P>
    <DIV class="contact_ways_item">
      // dept: 6
      <P>
        // dept: 7
        <I class="fa-sharp">
      <P>
    <DIV class="contact_ways_item">
      // dept: 6
      <P>
        // dept: 7
        <I class="fa-solid">
      <P>
    <HR>
  <FORM class="contact_form">
    // dept: 5
    <LABEL>
    <INPUT>
    <LABEL>
    <INPUT>
    <LABEL>
    <INPUT>
    <P>
      // dept: 6
      <BUTTON>
        // dept: 7
        <I class="fa-solid">
```

2. DOM Tree Footer



```
▼ <FOOTER class="footer">
  // dept: 3
  ▼ <DIV class="footer_container">
    // dept: 4
    <H3>
    <P>
    ▼ <P>
      // dept: 5
      <A>
    </P>
  </DIV>
  ▼ <DIV class="footer_container">
    // dept: 4
    <H3>
    ▼ <UL>
      // dept: 5
      ▼ <LI>
        // dept: 6
        <IMG>
        ▼ <DIV>
          // dept: 7
          <SPAN>
          <BR>
          <SPAN>
        </DIV>
      </LI>
      // dept: 6
      <IMG>
      ▼ <DIV>
        // dept: 7
        <SPAN>
        <BR>
        <SPAN>
      </DIV>
    </UL>
  </DIV>
  ▼ <DIV class="footer_container_tags">
    // dept: 4
    <H3>
    ▼ <P class="footer_container_tags">
      // dept: 5
      14 <SPAN>
```

3. CSS 공통, 화면 별 media query

```
1 @import './reset.scss';
2 @import './responsive.scss';
3 @import './variables.scss';
4 @import './custom.scss';
5
6 @import url('https://fonts.googleapis.com/css2?family=Raleway:wght@300;400;600&display=swap');
7
8 * {
9     font-family: 'Raleway', sans-serif;
10    line-height: 1.5;
11 }
12
13 body {
14     min-width: 350px;
15     max-width: 2000px;
16     background-color: $lightGray;
17 }
18
19 h1, h2, h3, h4, h5, h6 {
20     font-weight: 400;
21 }
22
23 button {
24     &:hover {
25         cursor: pointer;
26     }
27 }
```

// 공통 CSS 스타일

```
1 $breakpoint-tablet: 768px;
2 $breakpoint-desktop: 1024px;
3
4 @mixin tablet {
5     @media screen and (min-width: $breakpoint-tablet) and (max-width: ($breakpoint-desktop - 1px)) {
6         @content;
7     }
8 }
9
10 @mixin desktop {
11     @media screen and (min-width: $breakpoint-desktop) {
12         @content;
13     }
14 }
```

// 반응형 웹 용 Media query mixin

3. CSS reset.scss, variables.scss

```
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 html, body {
8   width: 100%;
9   height: 100%;
10  position: relative;
11 }
12
13 a {
14   text-decoration: none;
15   color: black;
16 }
17
18 ol, ul {
19   list-style: none;
20 }
21
22 header, article, section, footer, div {
23   display: block;
24 }
```

// 간단하게 필요한 것만 작성한 reset.css

```
1 $lightGray: #f1f1f1;
2 $gray: #ccc;
3 $darkGray: #616161;
4 $gray2: #9e9e9e;
5 $white: #fff;
6 $black: #000;
7 $coralGreen: #009688;
8
9 $pixelUnit: 16px;
10 $pixelUnit-small: 12px;
11 $pixelUnit-smallest: 8px;
12 $pixelUnit-large: 24px;
13 $pixelUnit-largest: 32px;
14
15 $fontSize: 15px;
16 $fontSize-small: 12px;
17 $fontSize-smallest: 6px;
18 $fontSize-large: 24px;
19 $fontSize-largest: 36px;
20
21 $itemSize: 400px;
22 $itemSize-tab: 350px;
23 $itemSize-desktop: 400px;
24
25 $sideBarWidth: 300px;
```

// 통합 변수 variables.scss

3. CSS 반응형 media query @mixin, @include

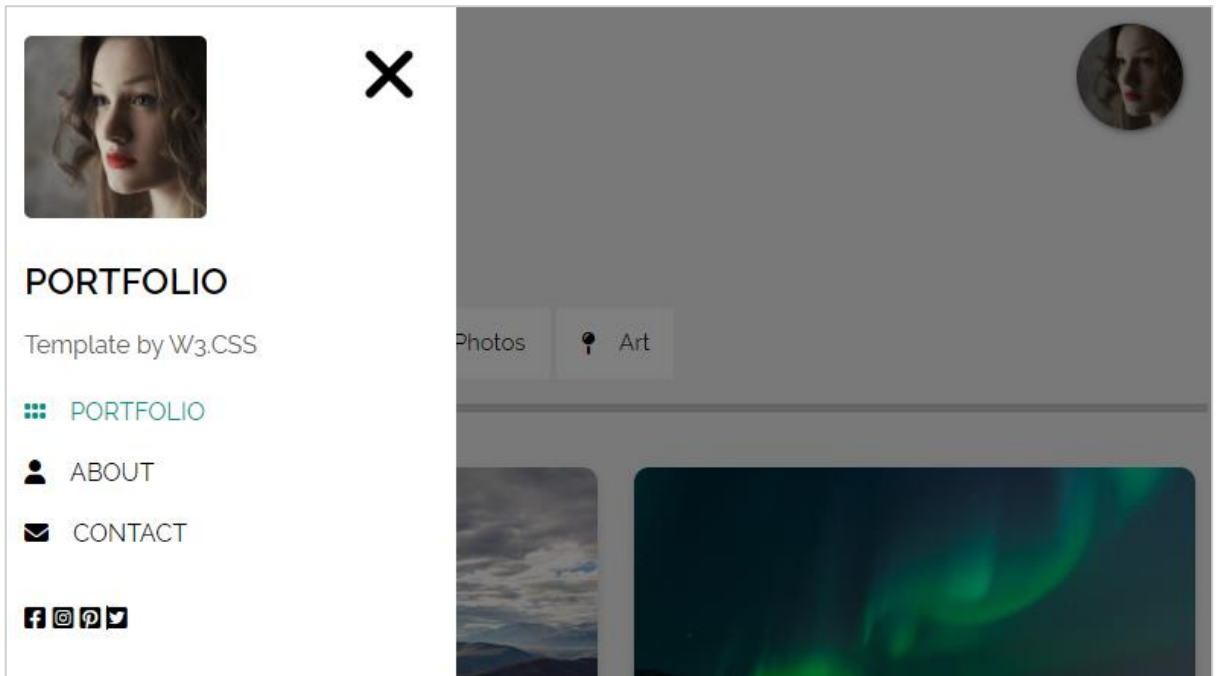
```
1  @mixin multiLineEllipsis($lineHeight, $lineNumber) {
2      display: -webkit-box;
3      -webkit-box-orient: vertical;
4      word-wrap: break-word;
5      overflow: hidden;
6      text-overflow: ellipsis;
7      -webkit-line-clamp: $lineNumber;
8      line-height: $lineHeight;
9      height: calc($lineHeight * $lineNumber);
10 }
11
12 @mixin processBar($width: 100%, $percentage: 50%,
13 $firstBackgroundColor: white, $secondBackgroundColor: skyblue,
14 $textColor: black) {
15     display: block;
16     width: $width;
17     padding: 0;
18     background-color: $firstBackgroundColor;
19     &::after {
20         display: block;
21         width: $percentage;
22         padding: 8px 16px;
23         content: $percentage + '%';
24         text-align: center;
25         background-color: $secondBackgroundColor;
26         color: $textColor;
27     }
28 }
29 }
```

// custom.scss

multiLuneEllipsis -> 설정한 라인을 넘기는 양의 텍스트에 ... 처리

processBar -> html .about-me 내의 진행도 바를 만들기 위한 mixin

3. CSS 대표 1

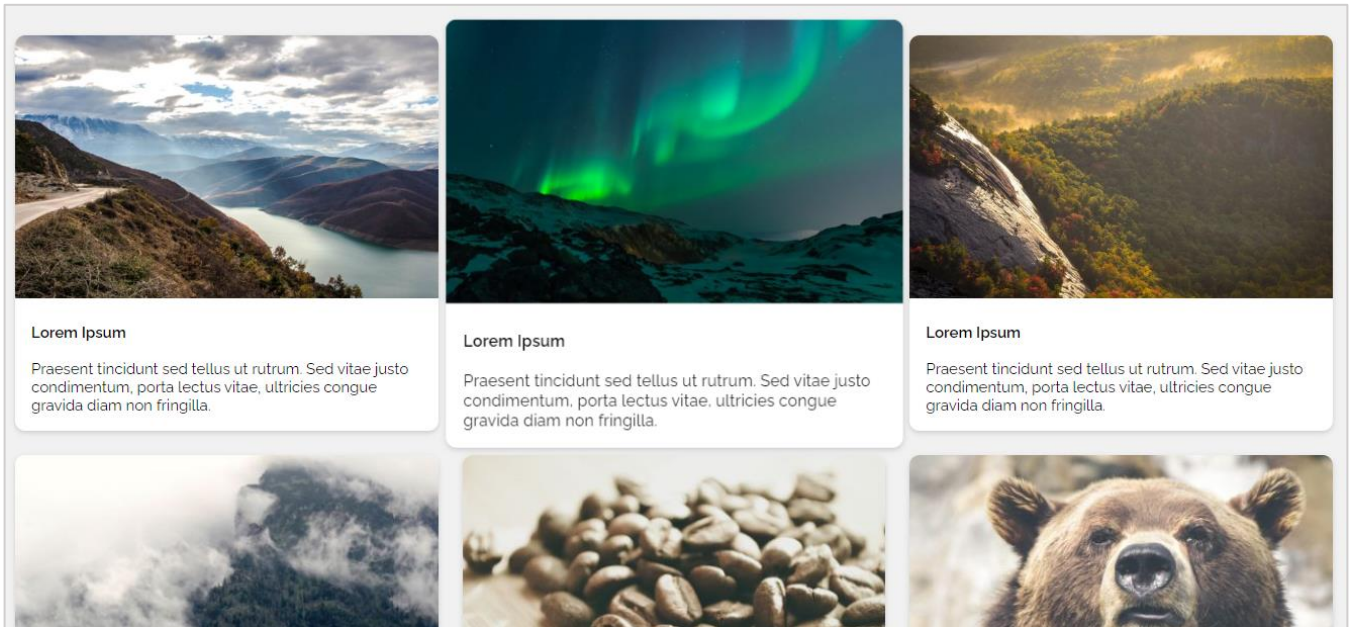


```
1 .side-menu {
2   display: none;
3   flex-flow: column wrap;
4   width: $sideBarWidth;
5   height: 100%;
6   padding: $pixelUnit-large 0;
7   background-color: $white;
8   position: fixed;
9   top: 0;
10  left: 0;
11  z-index: 3;
12
13  animation: slideOpen 0.5s;
14  @keyframes slideOpen {
15    0% {
16      left: -300px;
17      opacity: 0;
18    }
19    100% {
20      left: 0;
21      opacity: 1;
22    }
23  }
24
25  @include desktop {
26    display: flex;
27  }
```

```
1 .shadow-box {
2   display: none;
3   width: 100%;
4   height: 100%;
5   background-color: $black;
6   opacity: 0.5;
7   position: fixed;
8   top: 0;
9   left: 0;
10  z-index: 2;
11 }
```

태블릿 이하 화면에서 사이드 메뉴 숨김 처리,
버튼 클릭 시 나타내며 화면 어둡게.
등장 시 애니메이션 효과 적용

3. CSS 대표 2



```
1 .portfolios__item {
2   flex: 1 1 auto;
3   width: $itemSize;
4   margin: $pixelUnit-small;
5   background-color: $white;
6   border-radius: 10px;
7   box-shadow: 0 2px 5px $gray;
8   transition: 0.2s;
9
10  &:hover {
11    cursor: pointer;
12    scale: 1.1;
13  }
14
15  @include tablet {
16    width: $itemSize-tab;
17  }
18  @include desktop {
19    width: $itemSize-desktop;
20  }
21 }
```

```
1 .portfolios__item__content {
2   padding: 0 $pixelUnit;
3
4   p {
5     margin: $pixelUnit 0;
6     font-size: $fontSize;
7
8     &:last-child {
9       @include multiLineEllipsis(1.2em, 3);
10    }
11  }
12 }
```

화면 사이즈 별 정렬,
Hover 시 확대 효과,
마지막 p ellipsis 처리

4. JS

```
1 <!-- Side menu control -->
2 <script type="module">
3   const sideMenu = document.querySelector('.side-menu');
4   const shadowBox = document.querySelector('body > .shadow-box');
5
6   document.querySelector('#side-menu__open').addEventListener('click', e => {
7     sideMenu.style.display = 'flex';
8     shadowBox.style.display = 'block';
9   });
10
11   document.querySelector('#side-menu__quit').addEventListener('click', e => {
12     sideMenu.style.display = 'none';
13     shadowBox.style.display = 'none';
14   });
15
16   window.onresize = e => {
17     if (document.documentElement.clientWidth >= 1024) {
18       sideMenu.style.display = 'flex';
19       shadowBox.style.display = 'none';
20     } else {
21       if (shadowBox.style.display == 'block') {
22         sideMenu.style.display = 'flex';
23         shadowBox.style.display = 'block';
24       } else {
25         sideMenu.style.display = 'none';
26         shadowBox.style.display = 'none';
27       }
28     }
29   };
30 </script>
```

// 디스플레이 별 Side menu 제어

```
1 <!-- preventDefault -->
2 <script type="module">
3   document.querySelector('#form__submit').addEventListener('click', e => {
4     e.preventDefault();
5   });
6 </script>
```

// Form submit button의 기본 기능 해제

4. JS

```
1 <!-- Modal window control -->
2 <script type="module">
3   document.querySelectorAll('.about-me__photos img').forEach(v => {
4     v.addEventListener('click', e => {
5       document.querySelector('.modal__body img').setAttribute('src', e.currentTarget.getAttribute('src'));
6       document.querySelector('.modal__top p').innerHTML = e.currentTarget.getAttribute('alt');
7       document.querySelector('.modal').style.display = "block";
8       document.querySelector('body').style.overflow = "hidden";
9     });
10  });
11
12  document.querySelector('#modal__close').addEventListener('click', e => {
13    document.querySelector('.modal').style.display = "none";
14    document.querySelector('body').style.overflow = "auto";
15  });
16 </script>
```

// Modal 창 제어

```
1 <!-- Dom tree -->
2 <script type="module">
3   import dom from './assets/helper/Dom.js';
4   console.log(dom(document.querySelector('body')));
5 </script>
```

// Dom 출력

5. 작업 소요 시간 및 소감

총 작업 소요 시간:

12시간

W3C의 코드를 아예 참조하지 않고 구현,
더불어 Flex 등의 실제 자주 쓰이는
반응형 웹 방식을 실습해 봄으로써
다양한 디바이스 환경에서의
소비자 접근성, 활용성에 대해 깊이 연구하였습니다.

더불어 반응형 웹에 사용 가능함직한
여러 스크립트 코드를 살펴볼 수 있어
매우 유익한 실습이었습니다.

감사합니다

↓ 결과물 웹 주소

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