

WEB3(3) (JAVA,React) 웹 전자지갑 풀스택 개발자

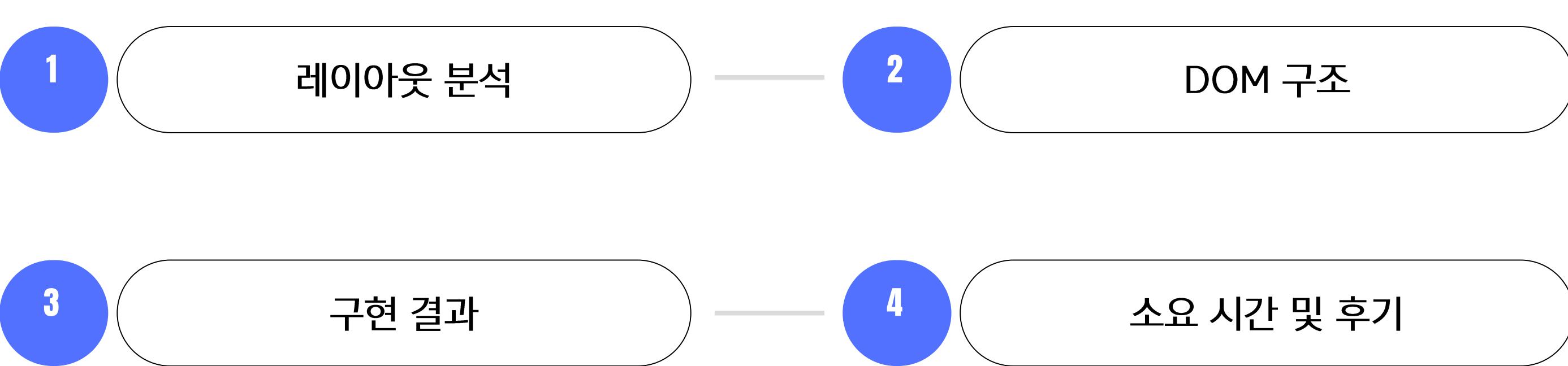
화면 UI 구현하기

HTML, CSS, JavaScript

박재한 박진수 이승현

목차

CONTENT



PART 1

레이아웃

분석

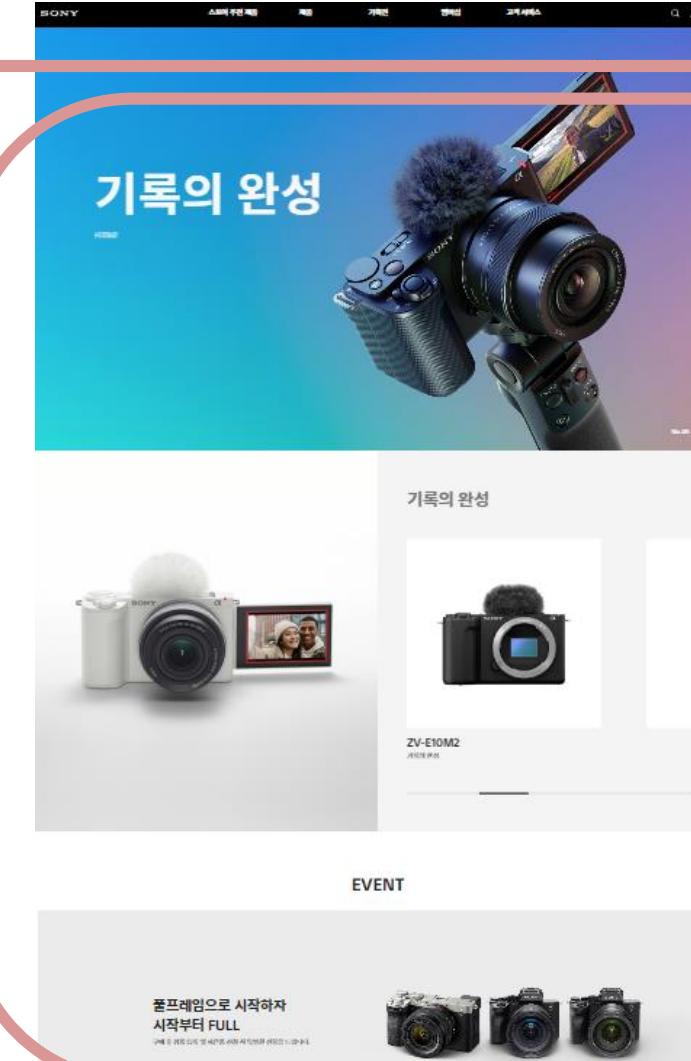
1

레이아웃 분석 (전체)

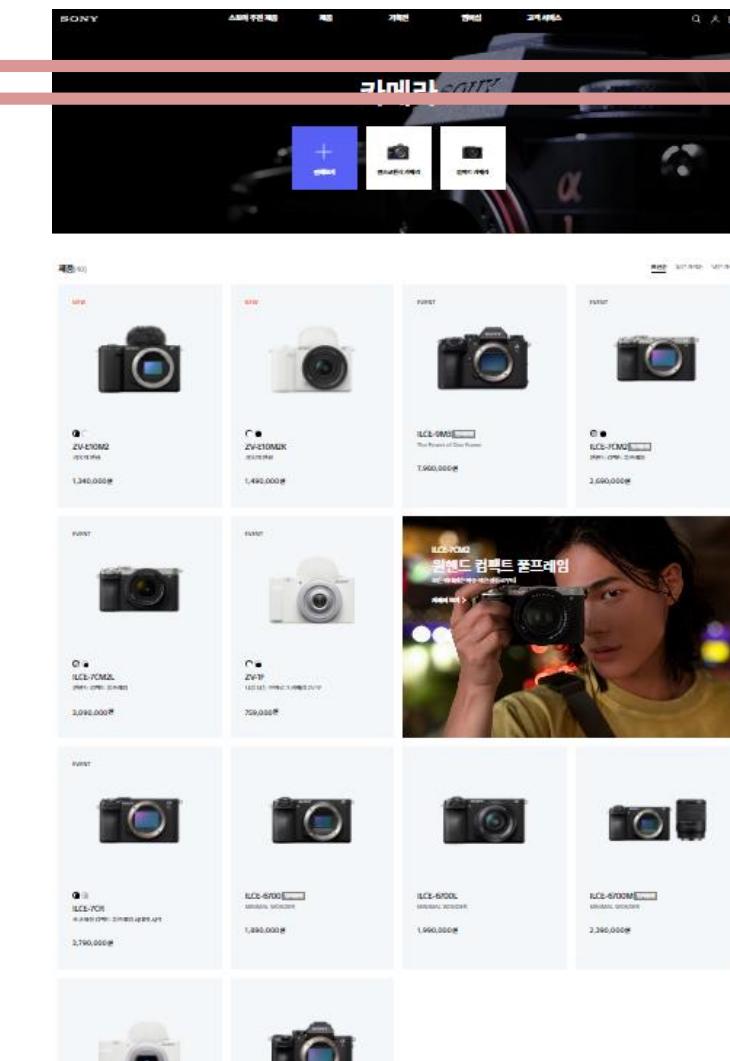
header

- 메뉴바

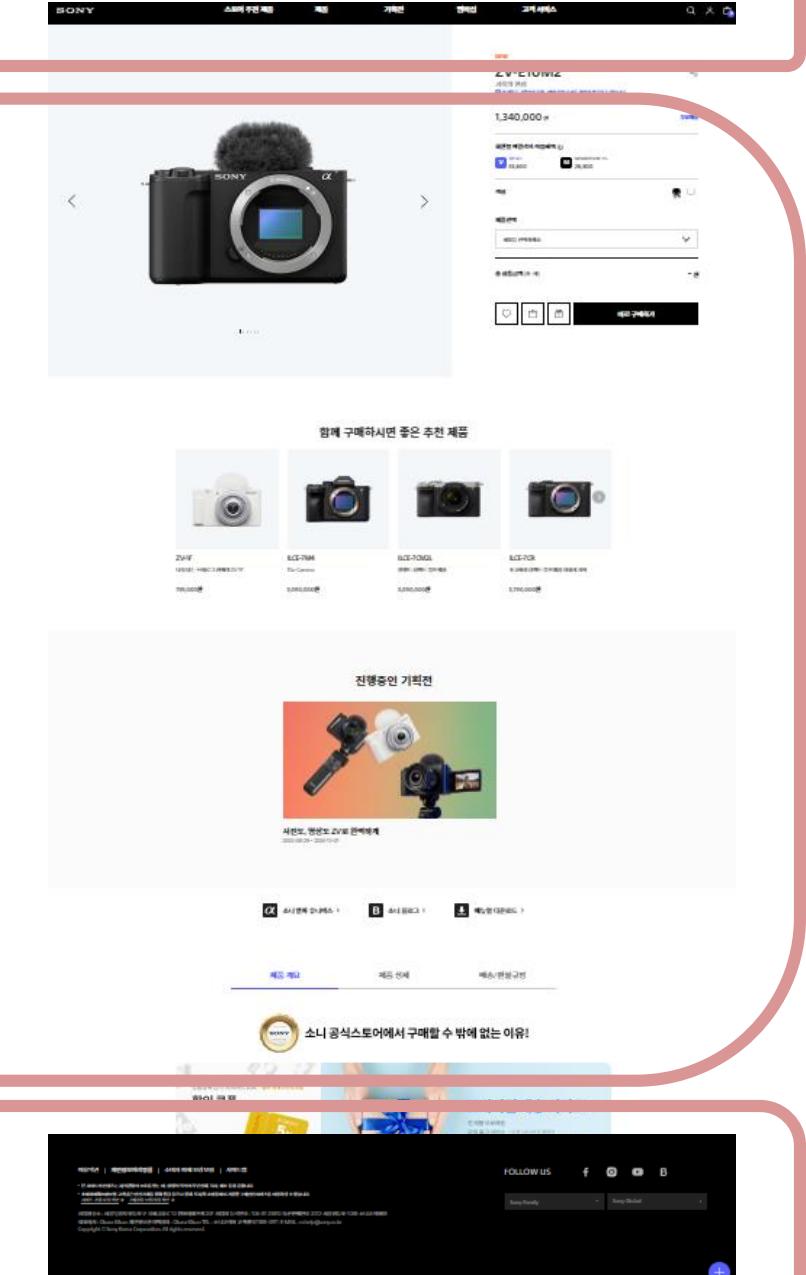
메인 페이지



상품 목록 페이지



상품 상세 페이지



main

- 주요 컨텐츠

footer

- 기타 링크, 카피라이트

1-1

레이아웃 분석 (header)



- nav는 전체화면을 채우도록 (gnb의 배경 역할), ul은 고정된 넓이로 더 크거나 작은 해상도에서도 일관성있게 보이도록 구성
- 좌측 로고와 우측 아이콘메뉴는 nav의 양 끝에 위치

- li에 마우스 호버 시 color, border-bottom 강조 효과, 해당 depth2의 ul요소에 배경색 변경
- 로고 클릭 시 index(root)페이지로 이동
- 아래로 스크롤 시 div1 숨기기, 위로 스크롤 시 div1 표시

1-1

레이아웃 분석 (header)

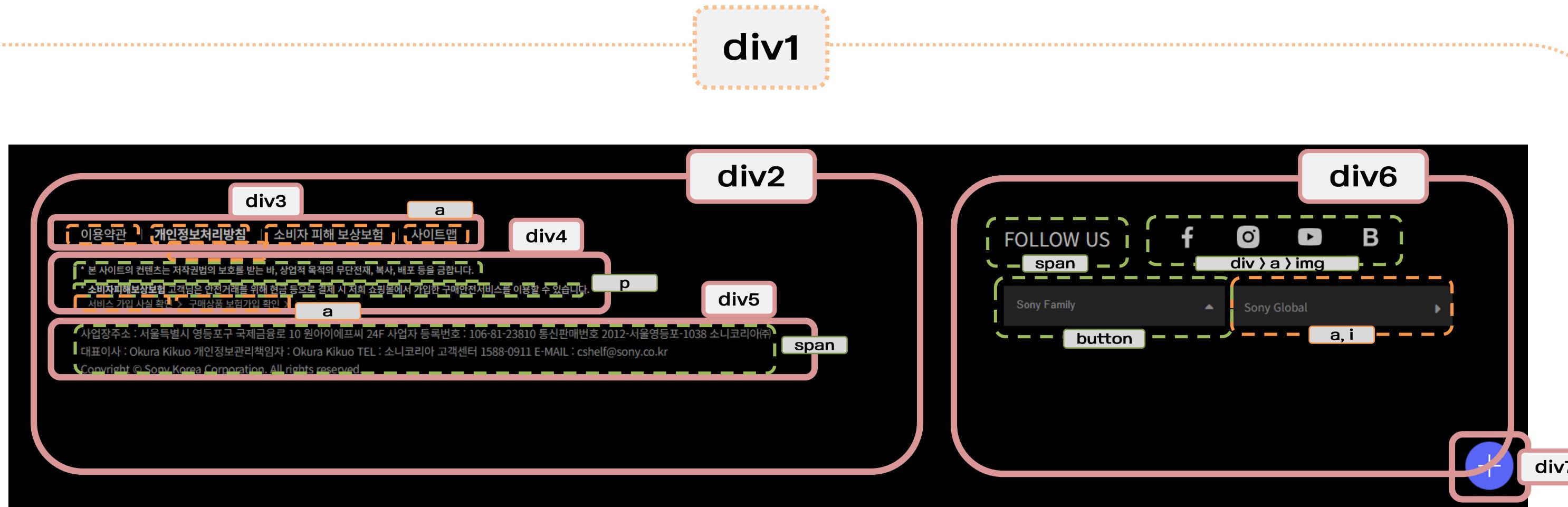


- div2는 전체화면을 채우도록 (depth2의 배경 역할), div3은 고정된 넓이로 더 크거나
작은 해상도에서도 일관성있게 보이도록 구성

- div2는 depth1의 ul에 마우스 호버 시 표시
- 각 ul에 마우스 호버 시 배경색 변경, 해당 depth1의 li요소에 color, border-
bottom 강조 효과

1-2

레이아웃 분석 (footer)

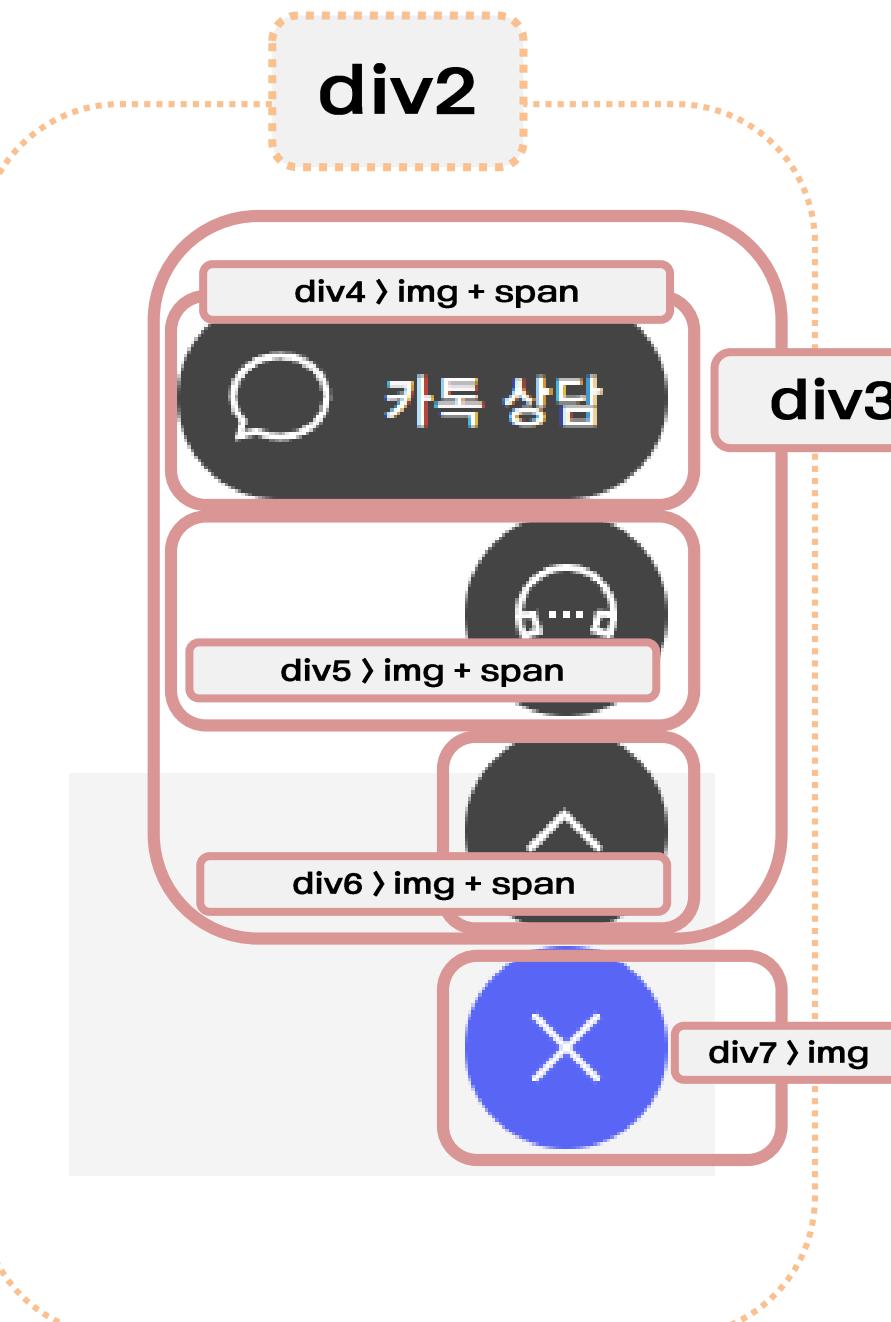
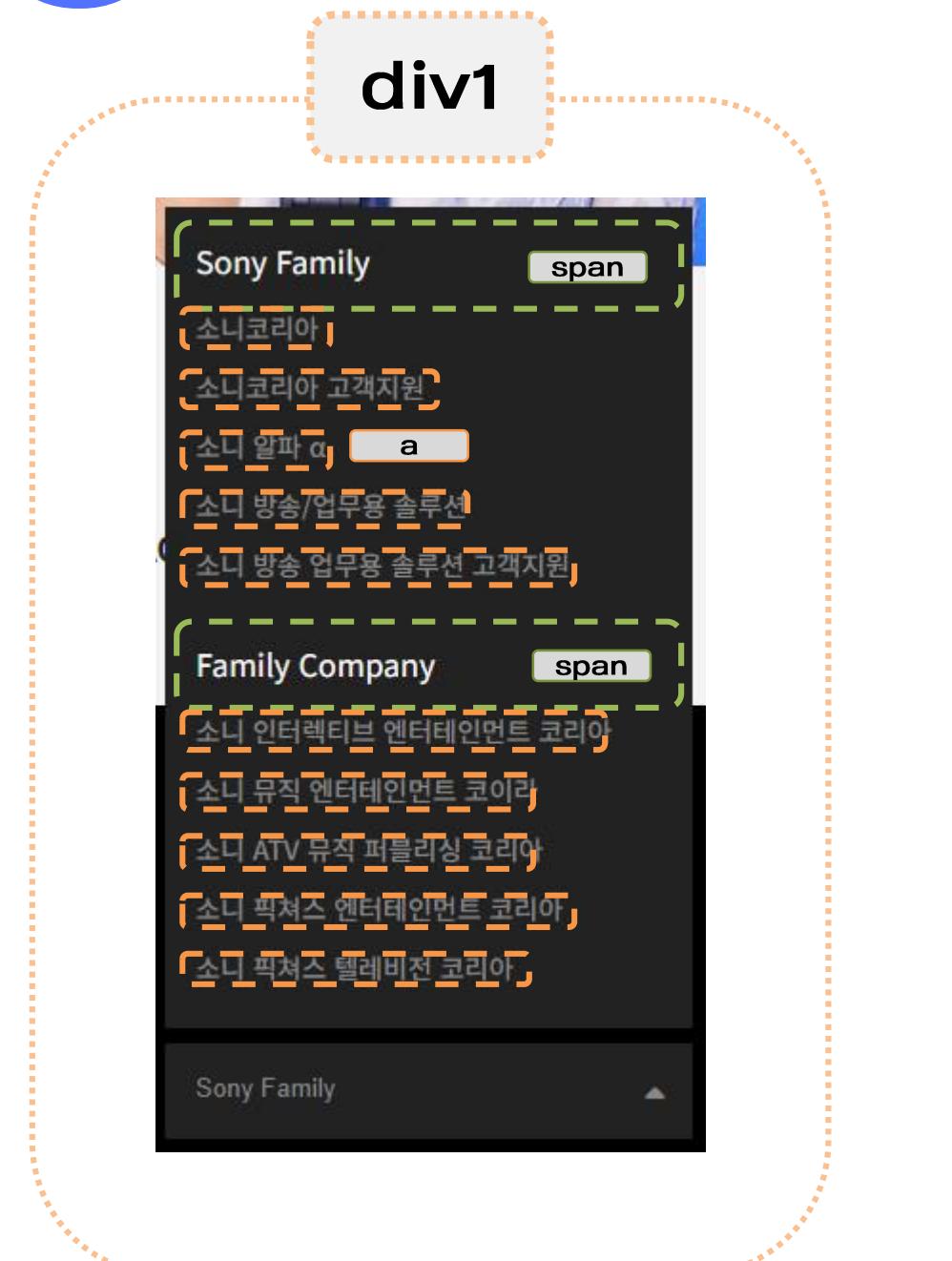


- flex로 div2는 왼쪽 영역, div6은 오른쪽 영역으로 분리

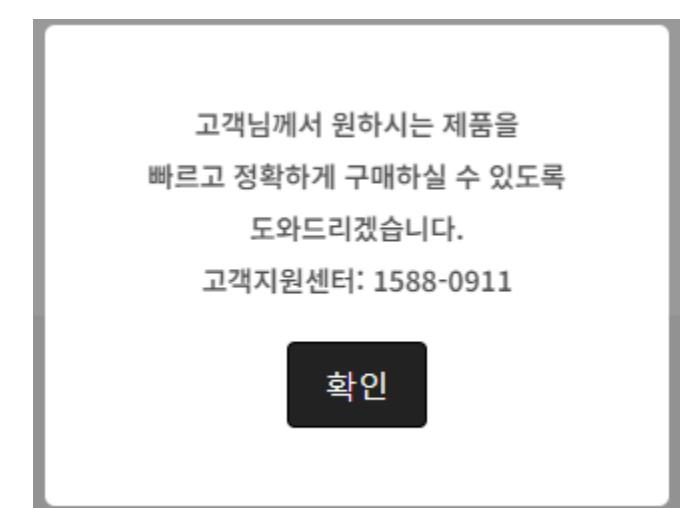
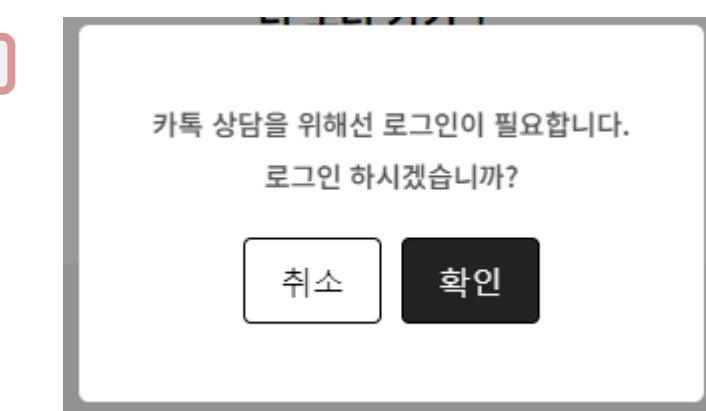
- 소셜아이콘(img)은 각 호버 시 이미지 색상 변경
- sony Family(button) 클릭 시 drop up 메뉴 표시
- div7은 fixed로 우측하단 고정이며 아래로 스크롤 시 표시, 위로 스크롤 시 숨김처리, 클릭 시 drop up 메뉴 표시

1-2

레이아웃 분석 (footer)

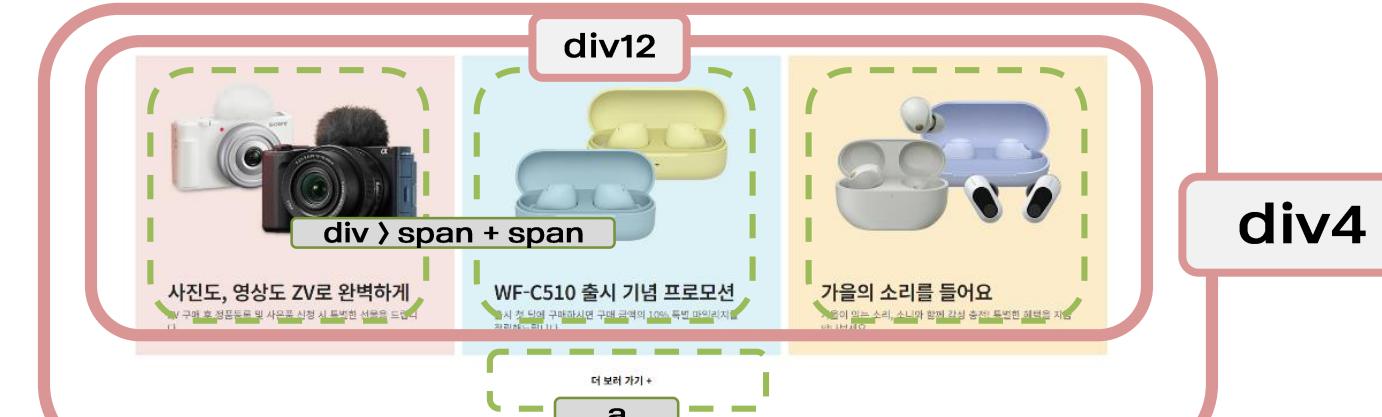


- 카톡 상담(div4), 고객 센터(div5) 호버 시 펼쳐지며 텍스트 표시되고, 클릭 시 각각 sweet alert2 창 표시
- div6 클릭 시 최상단 이동
- div7 클릭 시 + 아이콘으로 돌아가며 메뉴 접힘



1-3

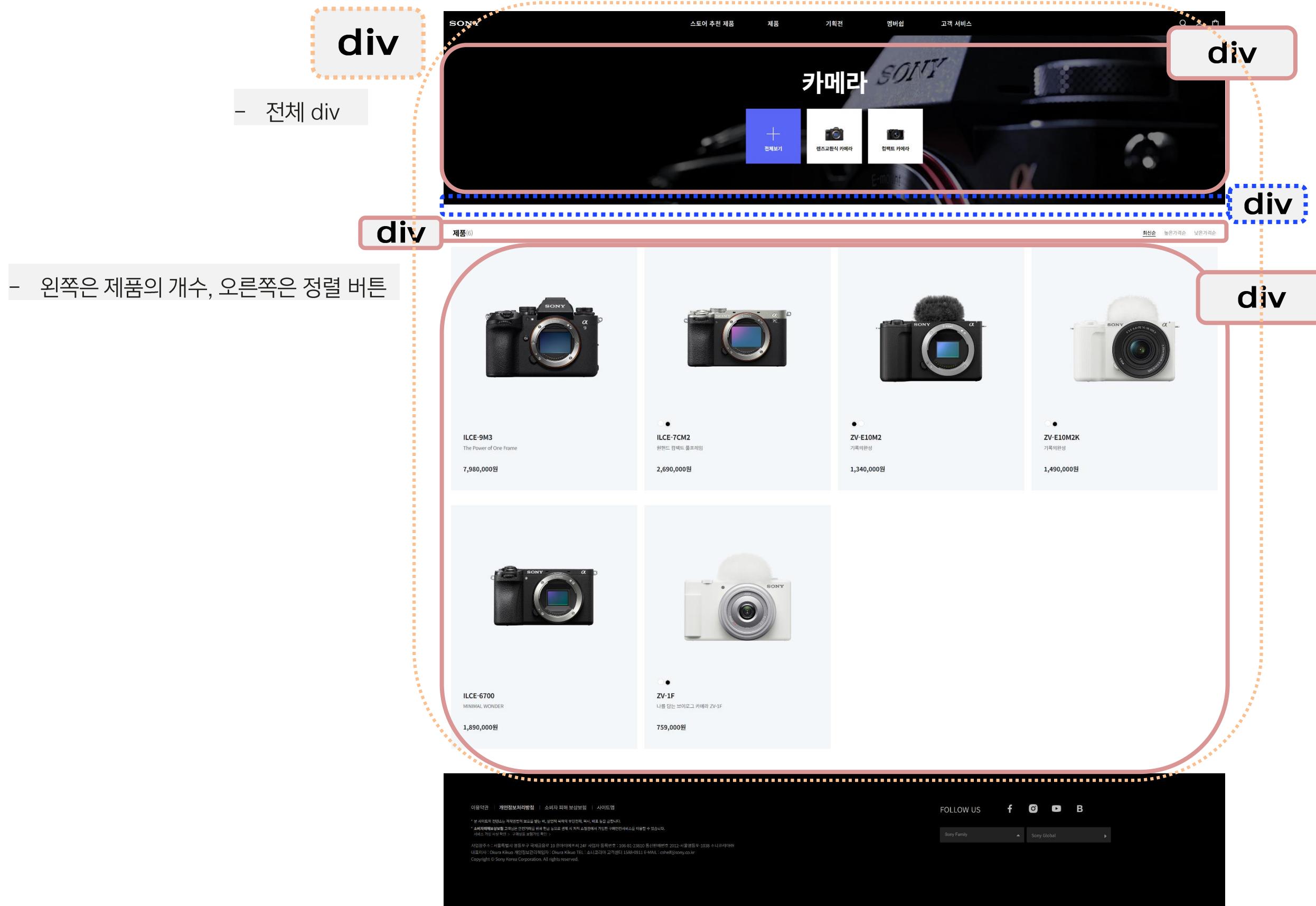
레이아웃 분석 (메인 페이지)



- div4 : 각 div영역 호버 시 이미지 scale 1.1배
- div5 : a태그 영역 클릭 시 상품목록 페이지로 이동, a태그 호버 시 이미지 중앙으로 이동
- div1 : 배너 영역 좌우 슬라이드 가능
- div2 : div8과 div9를 flex로 왼쪽, 오른쪽 영역으로 나눈 후 각각 슬라이드 효과 삽입, 오른쪽 슬라이드만 컨트롤이 가능하며 오른쪽 슬라이드 시 왼쪽 영역이 함께 슬라이드 된다. 오른쪽 a 태그 영역 클릭 시 상세페이지 이동, span 태그는 숨겨져 있다가 슬라이드 active시 표시 된다.
- div10, div12, div5, div6 : img는 백그라운드 이미지

1-4

레이아웃 분석 (상품 목록 페이지)

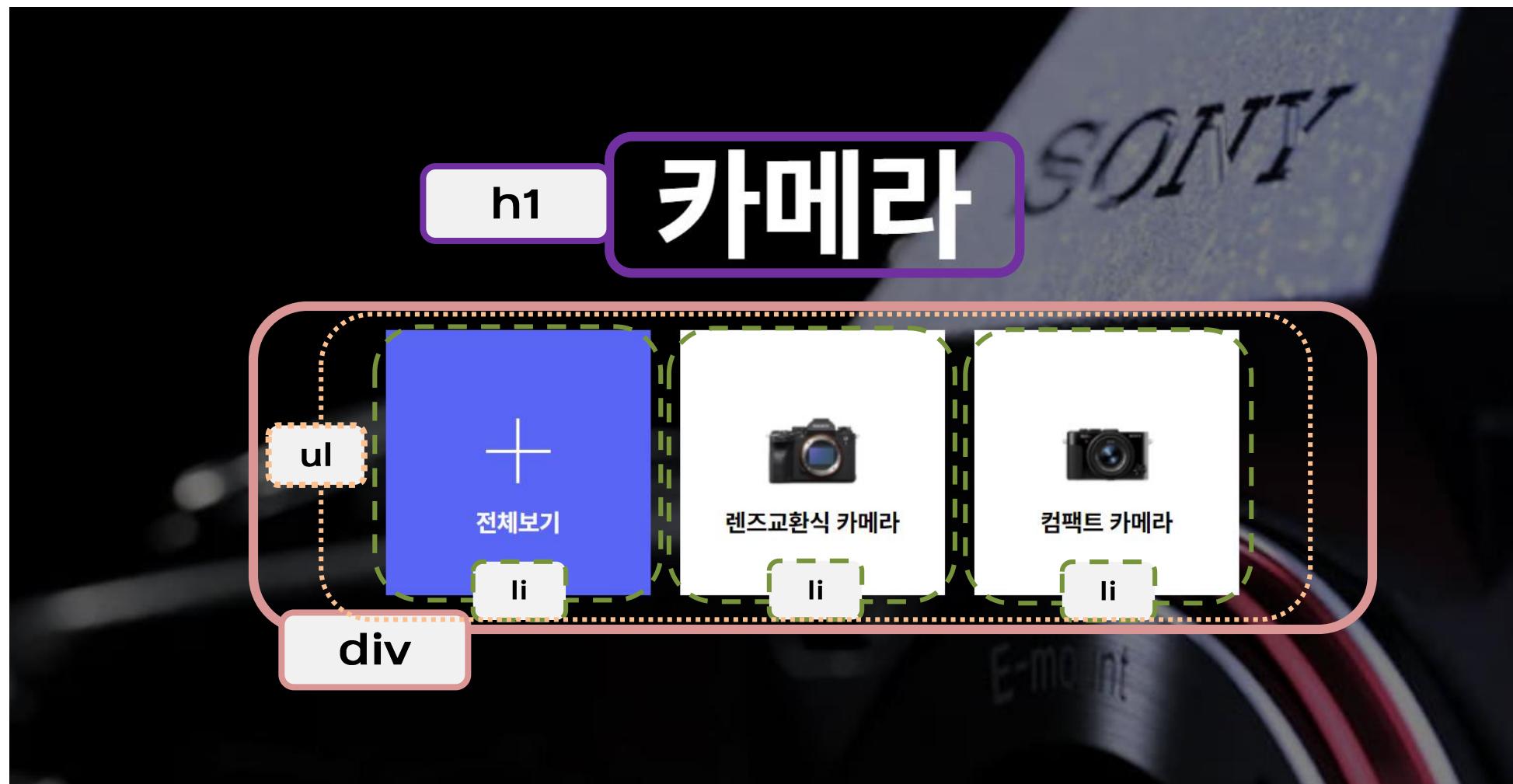


- 배경, 버튼 리스트

- 탭 메뉴 리스트(display:none 상태)

- 상품 목록 리스트

레이아웃 분석 (상품 목록 페이지)



- 카테고리 제목

- 버튼 리스트와 다음 페이지에서 설명할 카테고리 링크 div를 감쌈(현재는 display:none 상태)
- ul로 버튼 리스트 구현

1-4

레이아웃 분석 (상품 목록 페이지)



- 렌즈 교환식 카메라 버튼을 클릭하면 카테고리 링크, 돌아가기 버튼, 탭 메뉴 생성



1-4

레이아웃 분석 (상품 목록 페이지)



- 컴팩트 카메라 버튼을 클릭하면 카테고리 링크, 돌아가기 버튼 생성
- 컴팩트 카메라는 하위 목록이 없어 탭 메뉴가 활성화 되지 않음



- 돌아가기 버튼 생성
- 카테고리 제목을 버튼 리스트 제목으로 변경
- 카테고리 링크 생성

레이아웃 분석 (상품 목록 페이지)



- 제품의 개수 count
- div를 div로 감싸는 형태

- 정렬(최신순, 높은 가격순, 낮은 가격순)
- 최신순이 기본 active 상태

1-4

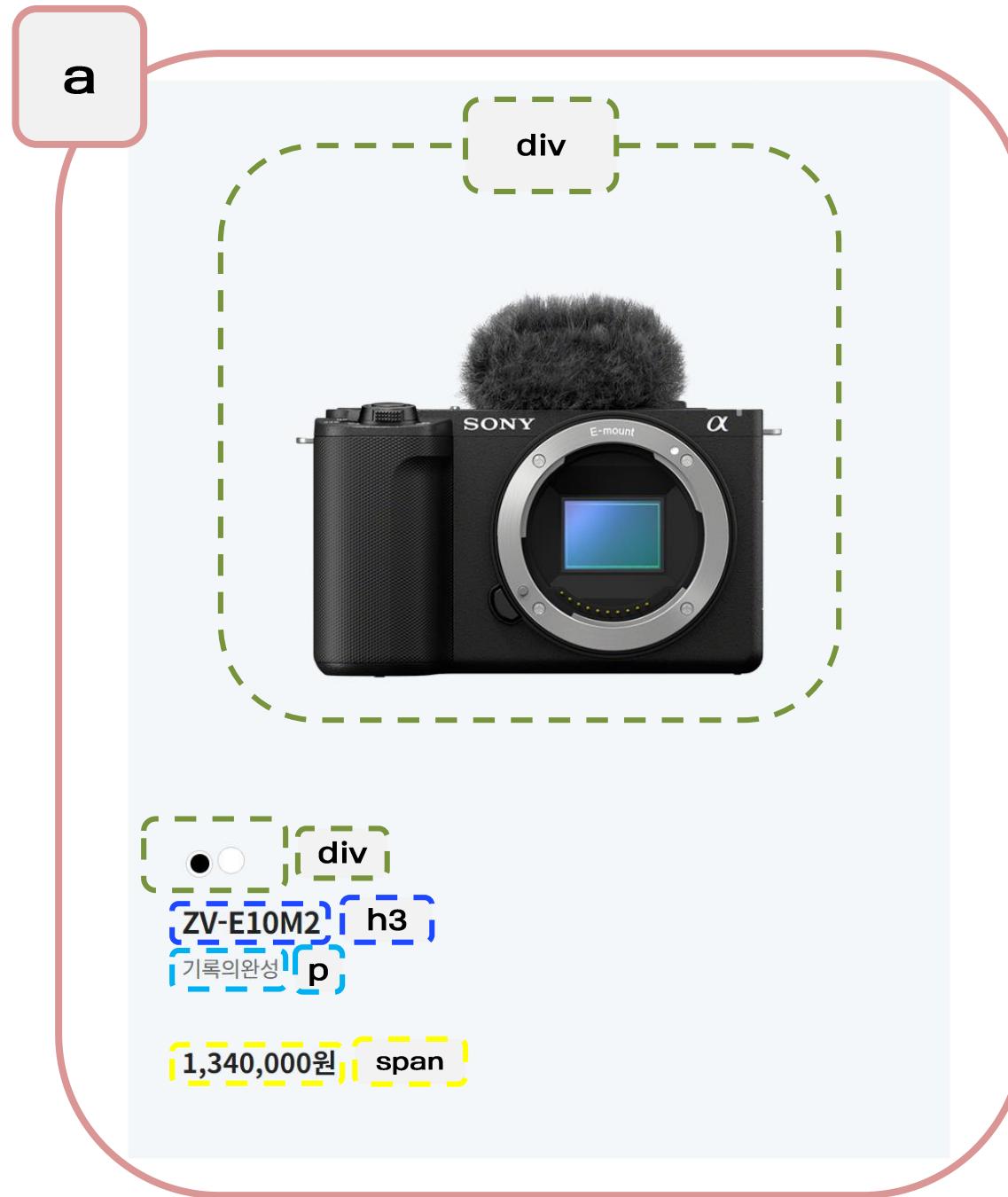
레이아웃 분석 (상품 목록 페이지)

- 이미지, 글자 어디를 클릭해도 상세 페이지로 가는 링크가 적용 되도록 div 대신 a태그를 사용

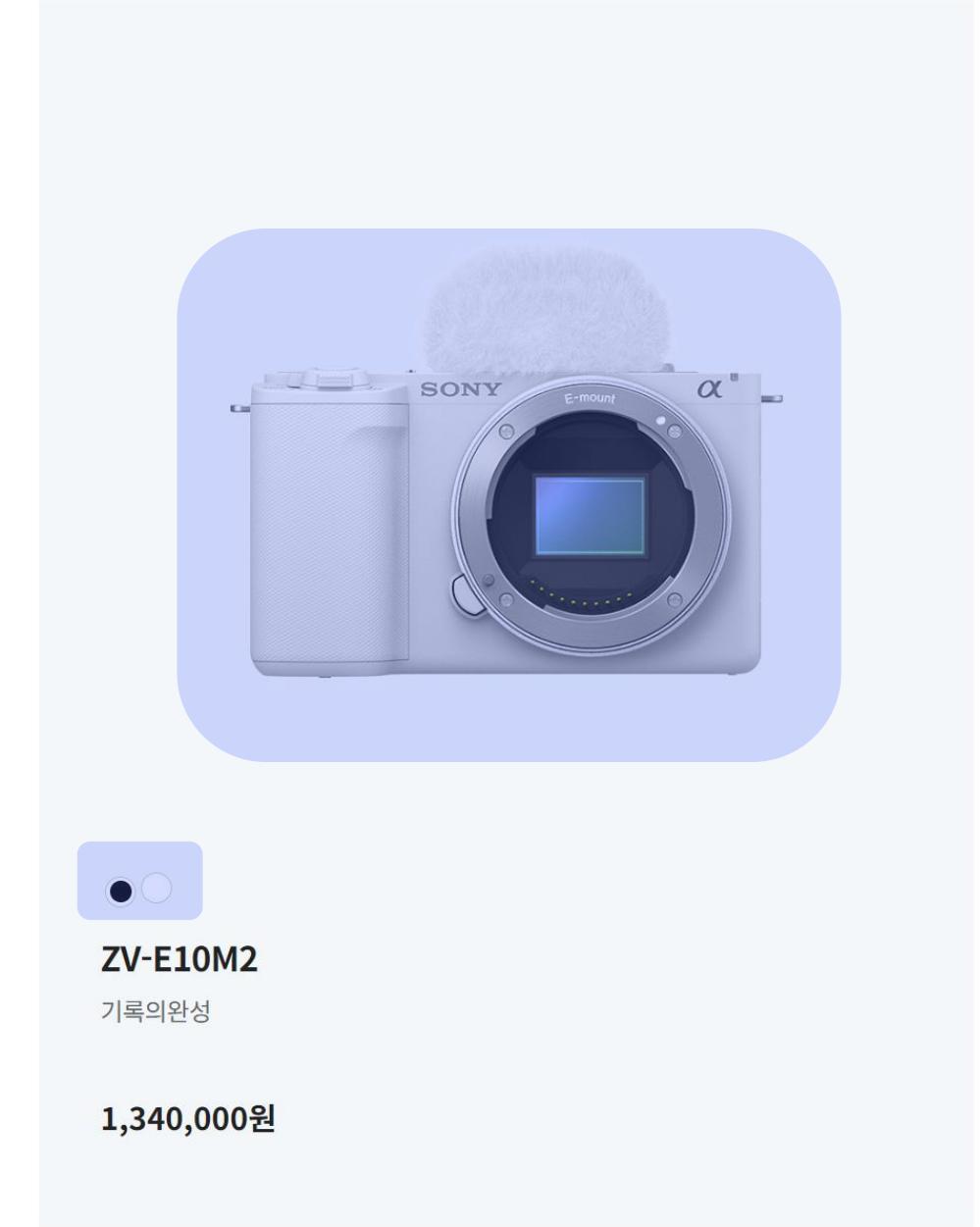
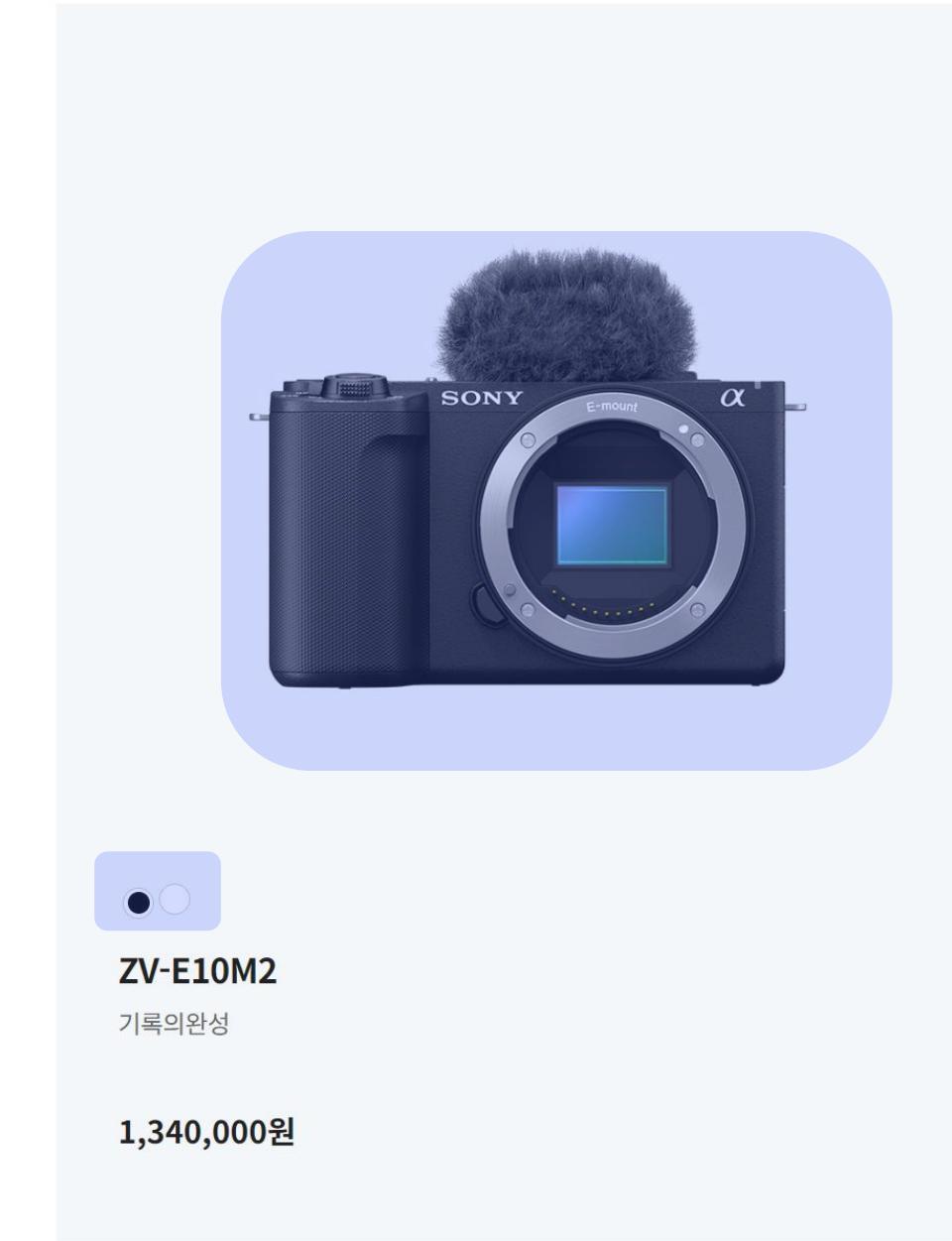


1-4

레이아웃 분석 (상품 목록 페이지)



- 색상 원형에 마우스 오버 시, 색상에 맞는 이미지 변화



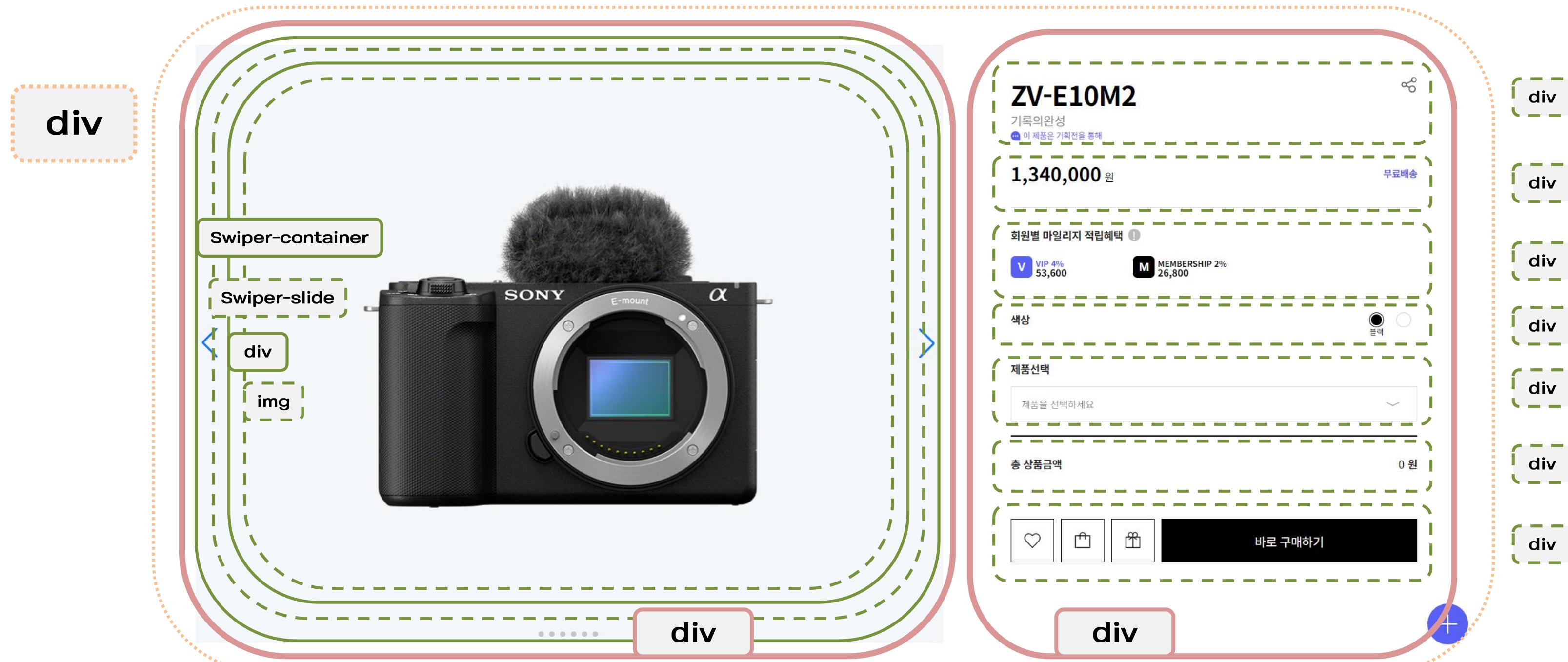
레이아웃 분석 (상품 상세 페이지)



상품 상세 페이지를 div 으로 전체를 감싸고
그 안의 콘텐츠들을 div 으로
1) 상품이미지, 구매관련
2) 추천제품
3) 기획전
4) 기타 소니 링크
5) 상품 상세 정보
로 구역을 나눔

1-5

레이아웃 분석 (상품 상세 페이지)



- 화면을 크게 좌,우 flex로 분리
- 좌측은 이미지 슬라이드 (swiper 라이브러리 사용)
- 우측은 상품의 정보들과 구매관련 내용

- 주요 내용마다 div으로 분리
- div으로 div를 감싸는 형태

1-5

레이아웃 분석 (상품 상세 페이지)

ZV-E10M2

기록의완성

이 제품은 기획전을 통해

1,340,000 원

무료배송

회원별 마일리지 적립혜택 ⓘ

V VIP 4% 53,600 M MEMBERSHIP 2% 26,800

div

색상 **p**

제품선택

제품을 선택하세요

총 상품금액 0 원

바로 구매하기

- 색상 원형 클릭 시, 선택한 색상 강조되며 색상이름 표시, 좌측 슬라이드 이미지 색상 변화

1,340,000 원

무료배송

회원별 마일리지 적립혜택 ⓘ

V VIP 4% 53,600 M MEMBERSHIP 2% 26,800

색상

제품선택

제품을 선택하세요

총 상품금액 0 원

1,340,000 원

무료배송

회원별 마일리지 적립혜택 ⓘ

V VIP 4% 53,600 M MEMBERSHIP 2% 26,800

색상

제품선택

제품을 선택하세요

총 상품금액 0 원

1-5

레이아웃 분석 (상품 상세 페이지)

ZV-E10M2

기록의완성

이 제품은 기획전을 통해

1,340,000 원



무료배송

회원별 마일리지 적립혜택 ⓘ

V VIP 4%
53,600

M MEMBERSHIP 2%
26,800

색상

div



p
div

제품선택

제품을 선택하세요

ZV-E10M2 / 블랙

ZV-E10M2 / 화이트

a, span

li, a, span

li, a, span



바로 구매하기

div, ul

li, a

- “제품을 선택하세요” 클릭 시, 제품 옵션 박스 시각화
- 제품선택시, 서브메뉴 박스 생성

div

제품선택

제품을 선택하세요

div

div

a

X

제품

X

div

ZV-E10M2 / 화이트

div

1,340,000원

총 상품금액

p

4,020,000 원

div

div

div

- x 클릭 시, 제품 옵션 박스 삭제
- “-”, “+” button과 input으로 상품 수량 변경
- 선택한 상품과 수량에 따라 총 상품금액

레이아웃 분석 (상품 상세 페이지)

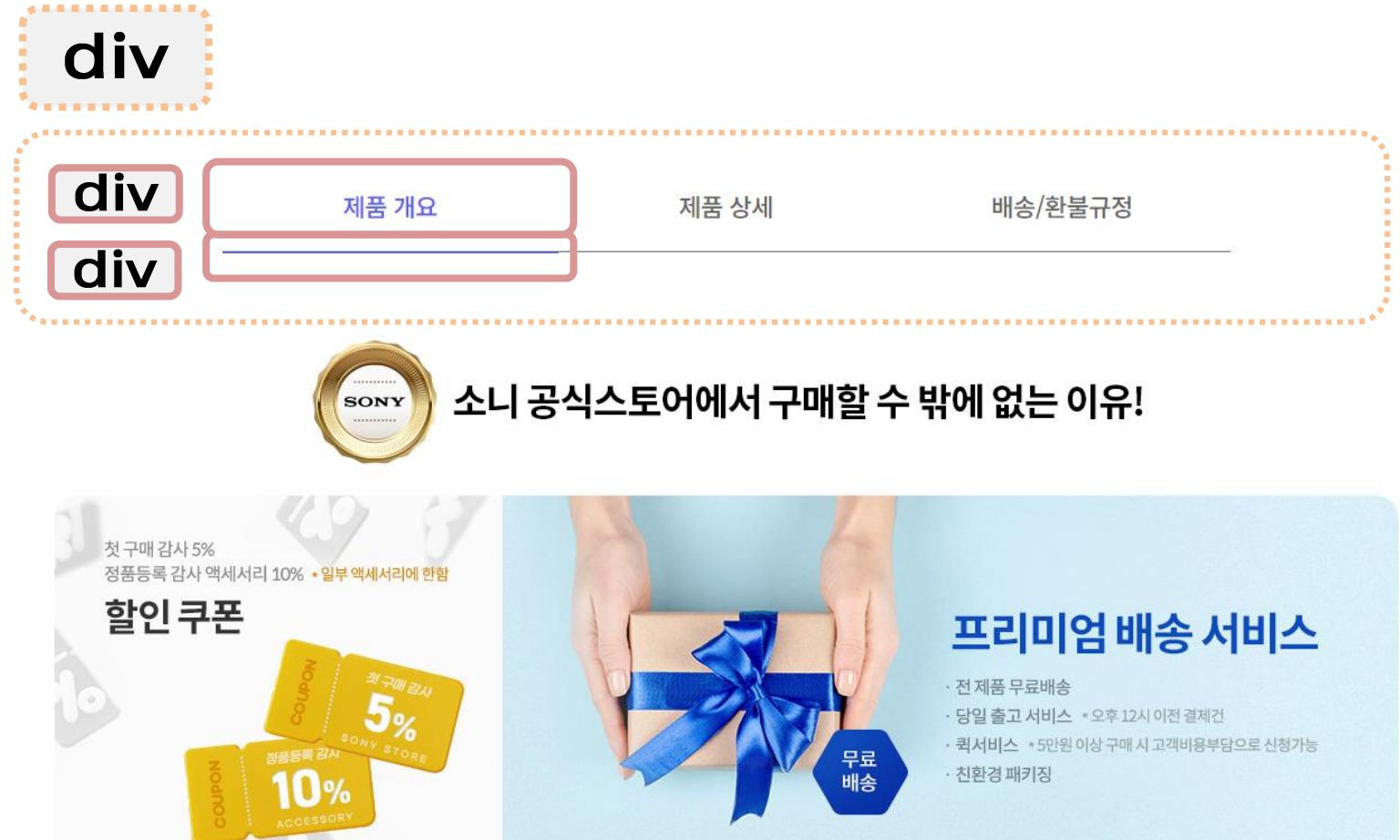


- 현재 상품을 제외한 같은 카테고리의 제품을 랜덤으로 표시
- 가로 드래그 슬라이더 적용

- 상품 div 클릭 시, 클릭한 상품의 상세페이지로 이동

1-5

레이아웃 분석 (상품 상세 페이지)

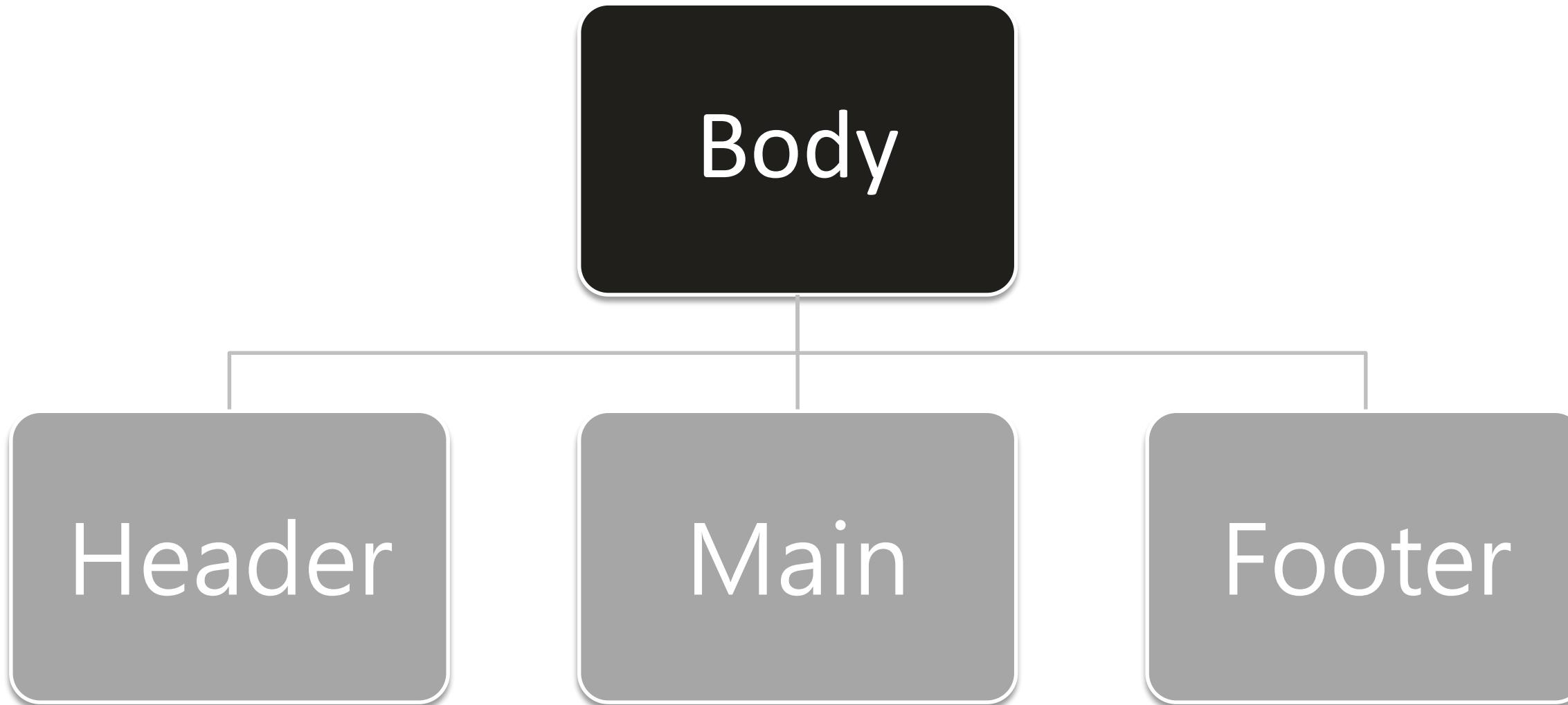


- “제품개요”, “제품상세”, “배송/환불규정” 호버 or 클릭 시, 색상으로 강조, 각 항목 시각화



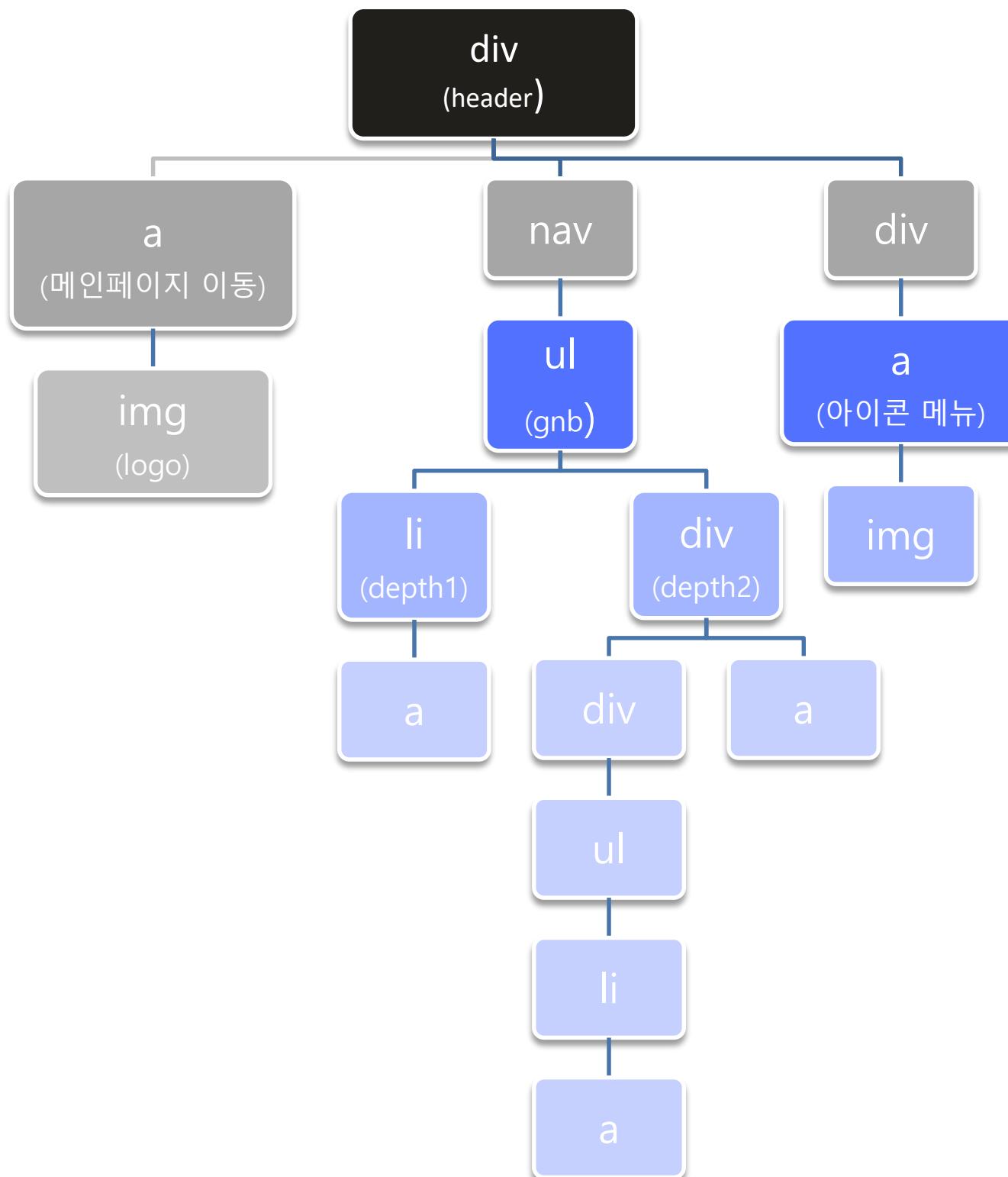
PART 2

DOM 구조



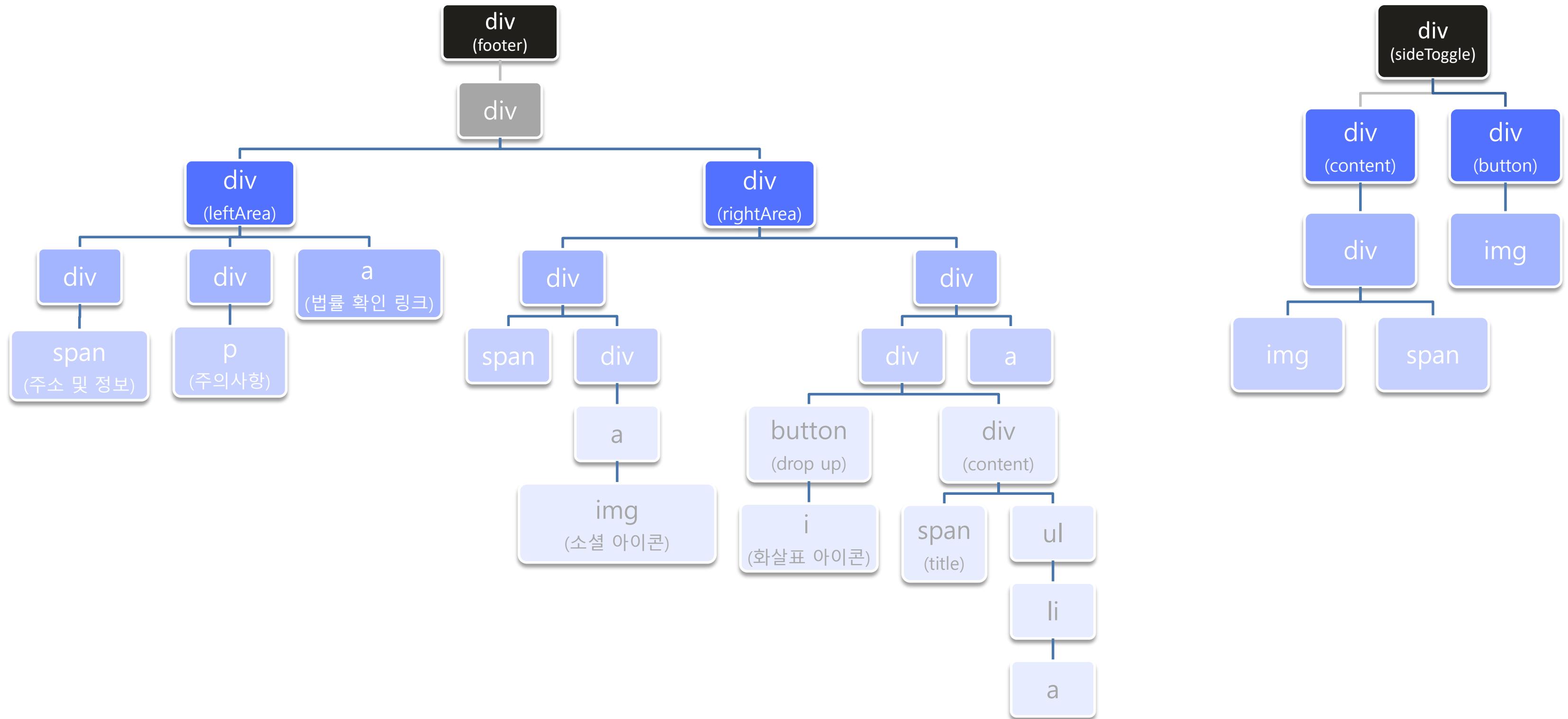
2-1

DOM 구조 - header



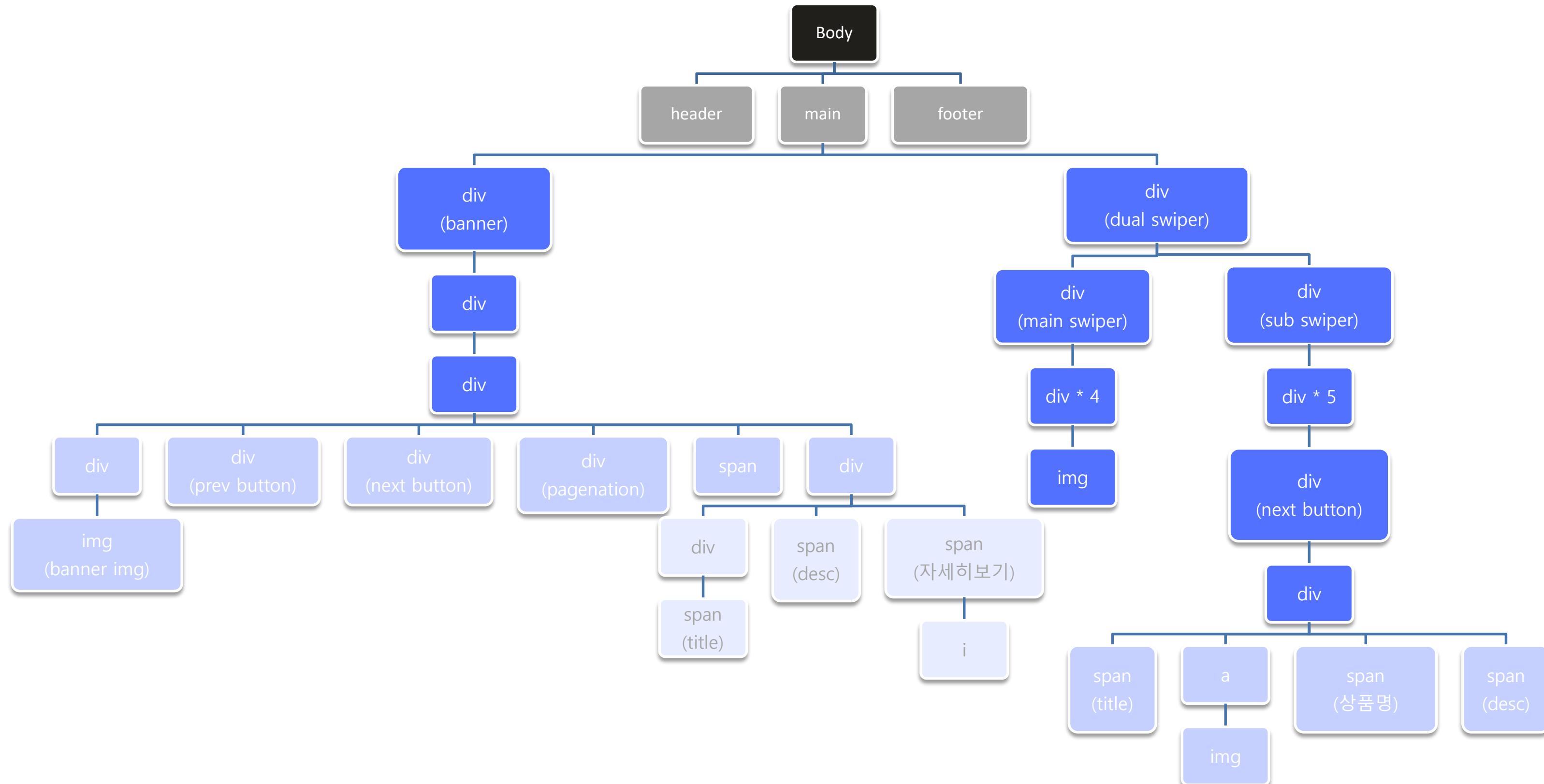
2-2

DOM 구조 - footer

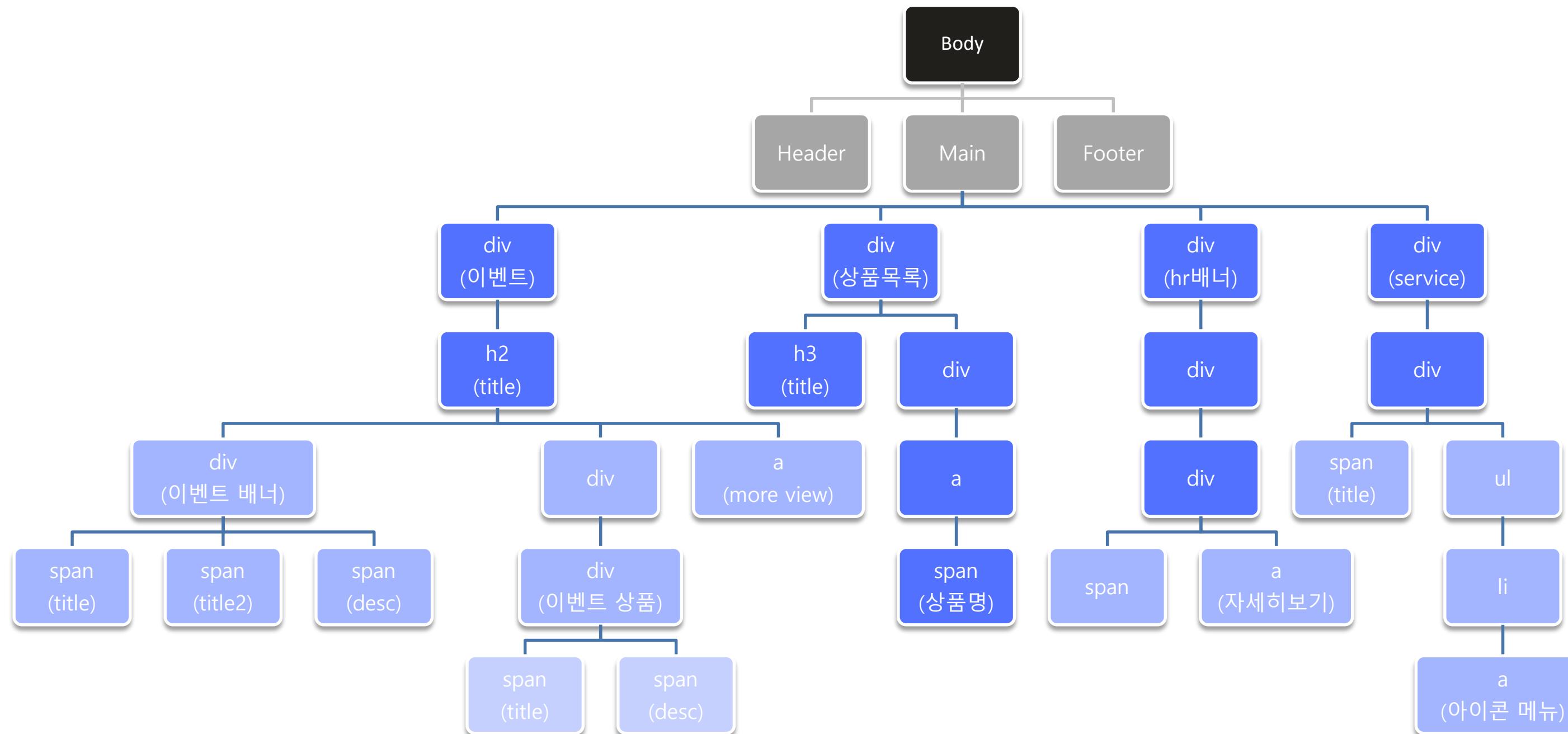


2-3

DOM 구조 - 메인 페이지

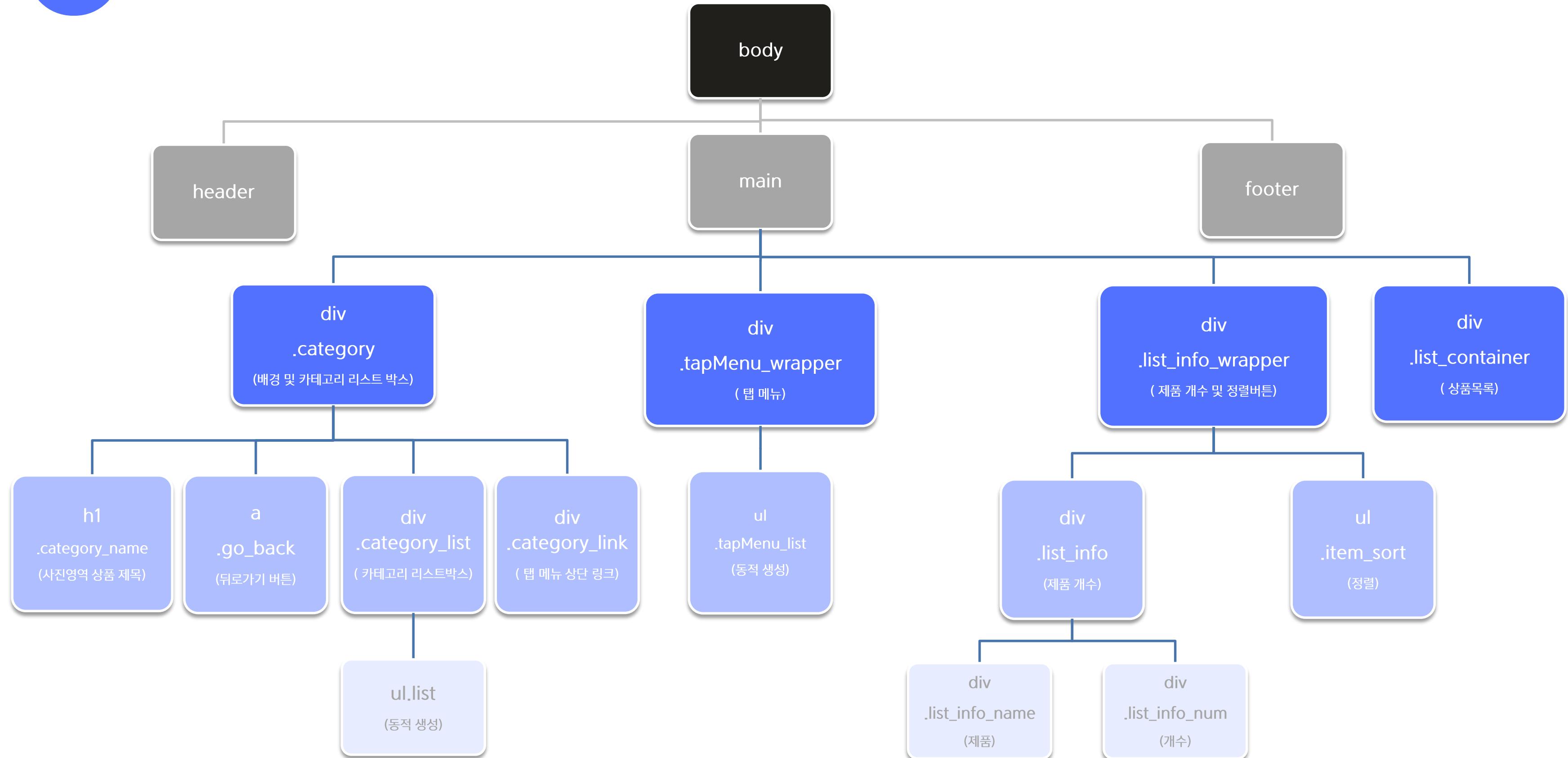


DOM 구조 - 메인 페이지

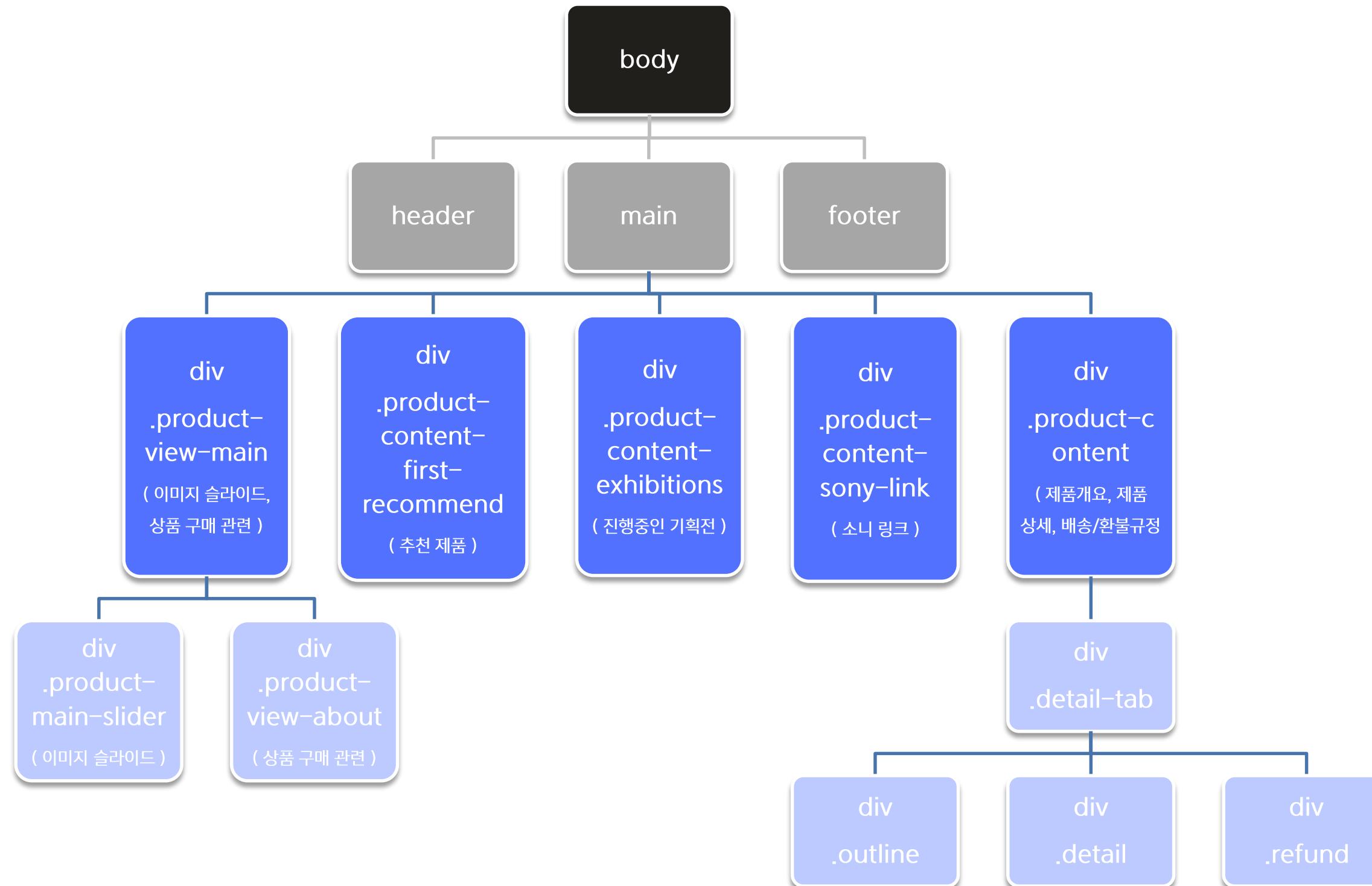


2-4

DOM 구조 - 상품 목록 페이지



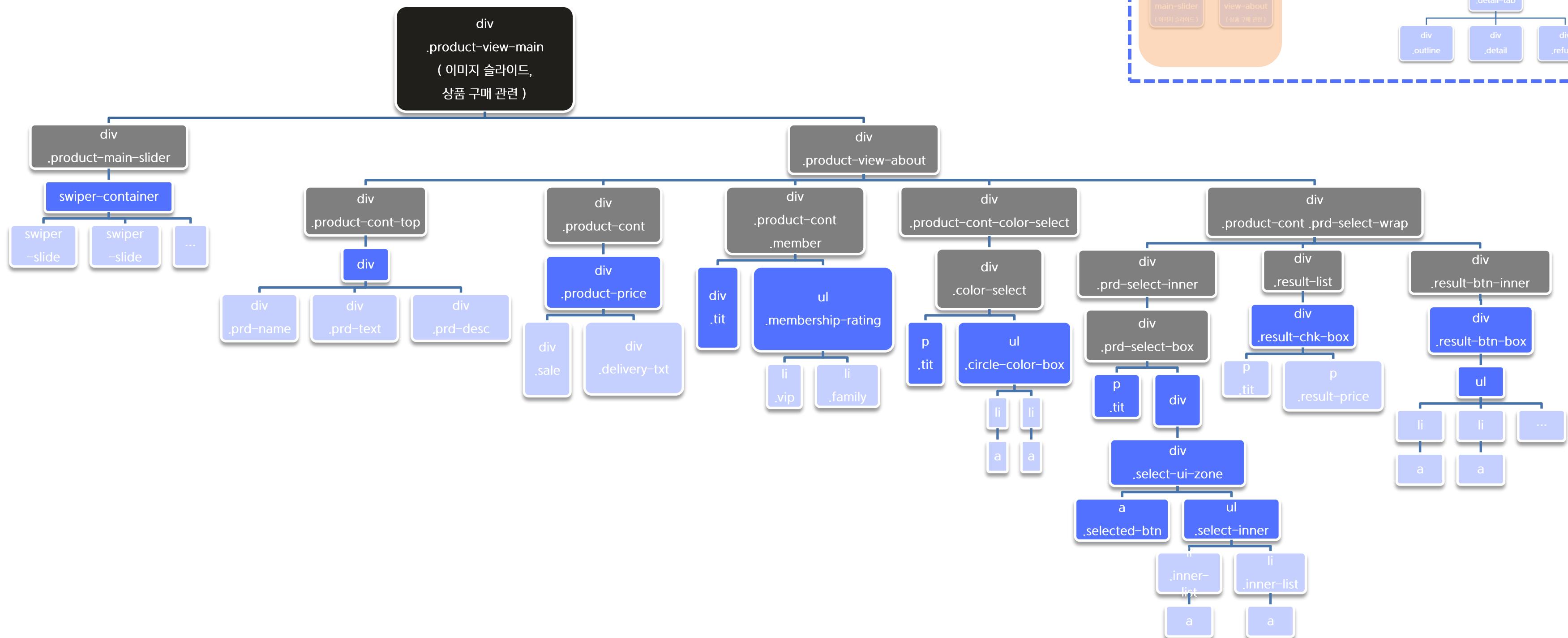
DOM 구조 - 상품 상세 페이지



2-5

DOM 구조 - 상품 상세 페이지

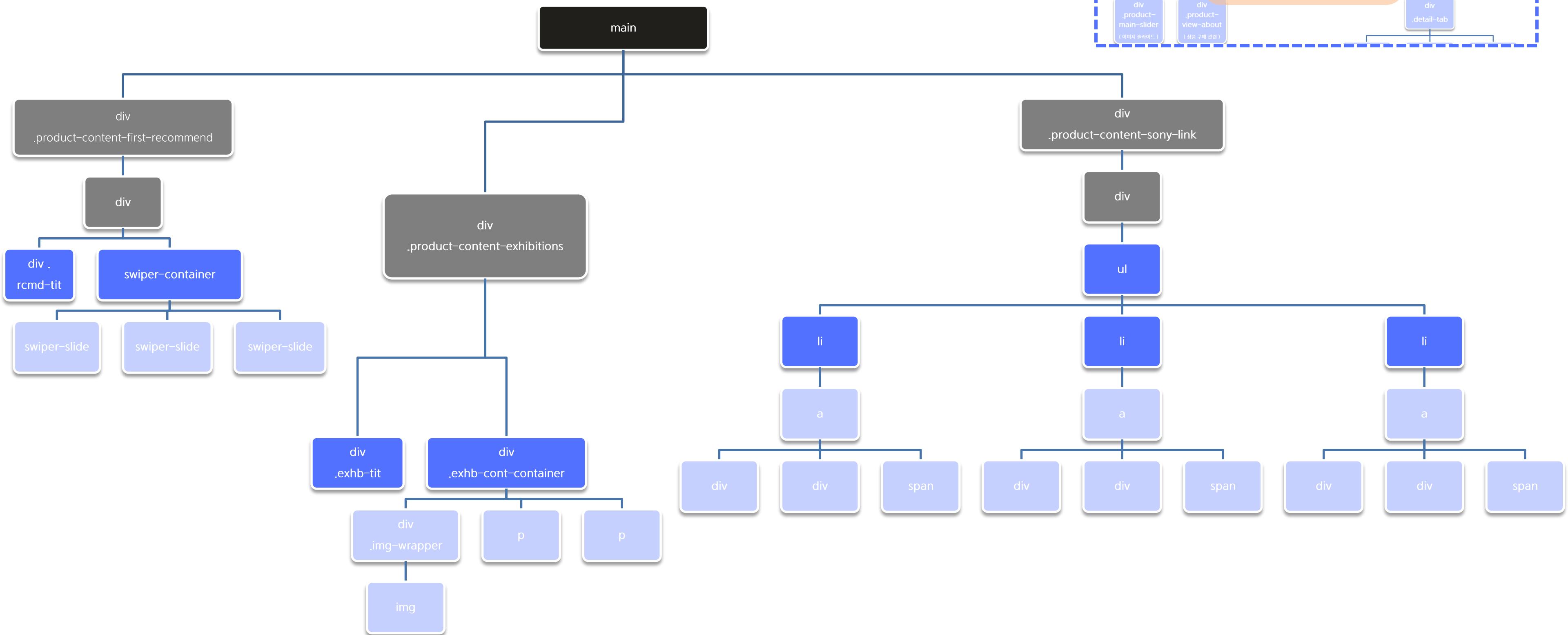
body > main > 이미지 슬라이드, 상품 구매 관련



2-5

DOM 구조 - 상품 상세 페이지

body > 추천제품, 진행중인 기획전, 기타 소니 링크

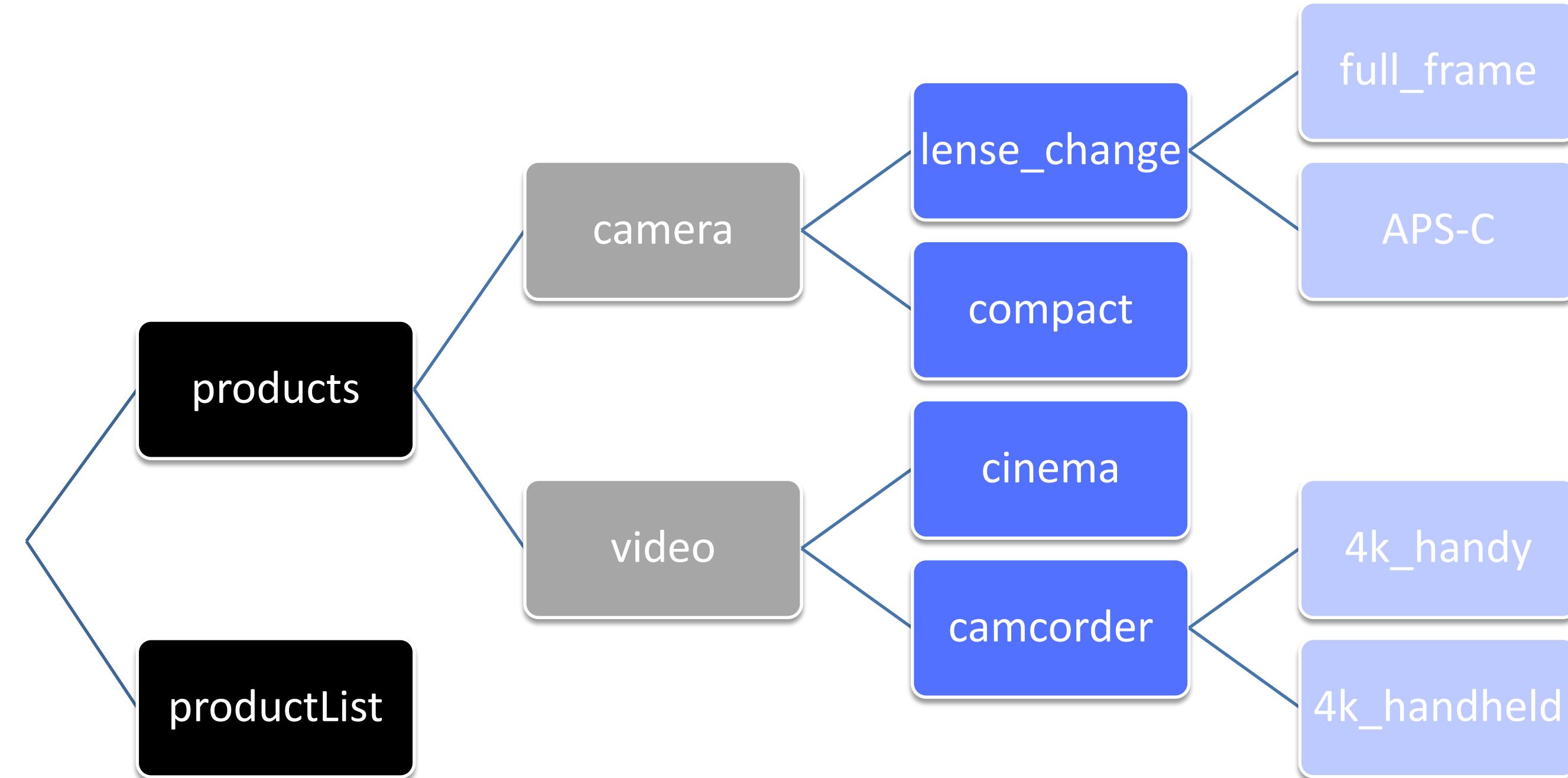


PART 3

구현 결과

3-0

구현 결과 (JSON)



3-0

구현 결과 (JSON)

```

{
  "products": [
    {
      "camera": {
        "lens_change": {
          "full_frame": [
            {
              "id": 3,
              "thumbnail": "clr0_0.png",
              "title": "ILCE-9M3",
              "info": "The Power of One Frame",
              "price": 7980000,
              "date": 202401,
              "color": ["black"],
              "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}]]
            },
            {
              "id": 4,
              "thumbnail": "clr0_0.png",
              "title": "ILCE-7CM2",
              "info": "원핸드 컴팩트 풀프레임",
              "price": 2690000,
              "date": 202309,
              "color": ["silver", "black"],
              "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}], [{"clr1_0.png"}, {"clr1_1.png"}, {"clr1_2.png"}, {"clr1_3.png"}]]
            }
          ],
          "APS-C": [
            {
              "id": 1,
              "thumbnail": "clr0_0.png",
              "title": "ZV-E10M2",
              "info": "기록의완성",
              "price": 1340000,
              "date": 202407,
              "color": ["black", "white"],
              "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}, {"clr0_4.png"}, {"clr0_5.png"}], [{"clr1_0.png"}, {"clr1_1.png"}, {"clr1_2.png"}, {"clr1_3.png"}, {"clr1_4.png"}, {"clr1_5.png"}]]
            }
          ]
        }
      }
    }
  ],
  "video": [
    {
      "cinema": [
        {
          "id": 1,
          "thumbnail": "video1.png",
          "title": "BURANO",
          "info": "풀프레임 8.6K 시네마 카메라",
          "price": 36990000,
          "date": 202402,
          "color": ["black"],
          "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}, {"clr0_4.png"}, {"clr0_5.png"}]]
        },
        {
          "id": 2,
          "thumbnail": "video2.png",
          "title": "ILME-FR7",
          "info": "시네마라인 렌즈교환식 PTZ 카메라",
          "price": 14990000,
          "date": 202211,
          "color": ["black"],
          "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}]]
        },
        {
          "id": 3,
          "thumbnail": "video3.png",
          "title": "ILME-FX30B (바디컷)",
          "info": "시네마라인 Super 35mm 카메라 (바디 컷)",
          "price": 2699000,
          "date": 202210,
          "color": ["black"],
          "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}]]
        }
      ]
    }
  ]
}

```

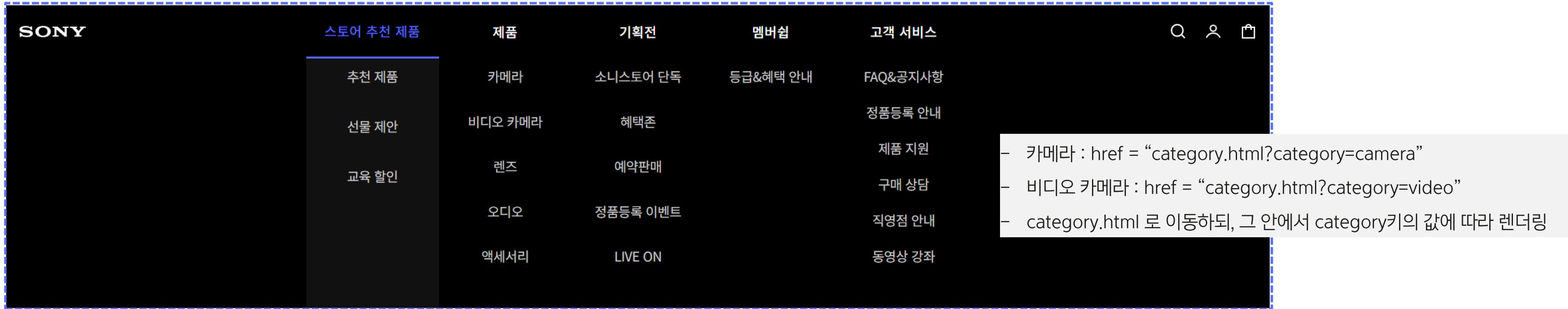
3-0

구현 결과 (JSON)

```
"camcorder": {
    "4k_handy": [
        {
            "id": 4,
            "thumbnail": "video4.png",
            "title": "FDR-AX43A",
            "info": "콘텐츠 크리에이터를 위한 4K 캠코더",
            "price": 1449000,
            "date": 202206,
            "color": ["black"],
            "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}, {"clr0_4.png"}]]
        },
        {
            "id": 5,
            "thumbnail": "video5.png",
            "title": "FDR-AX700",
            "info": "4K HDR(HLG), Fast Hybrid AF가 탑재된 전문가급 1인치 핸디캠",
            "price": 2499000,
            "date": 201710,
            "color": ["black"],
            "img": [[{"clr0_0.png"}, {"clr0_1.png"}, {"clr0_2.png"}, {"clr0_3.png"}, {"clr0_4.png"}]]
        }
    ],
    "4k_handheld": [
        {
            "id": 6,
            "thumbnail": "video6.png",
            "title": "HXR-NX800 + UWP-D27",
            "info": "바디 본품 + UWP-D27 마이크 번들 구성",
            "price": 5820000,
            "date": 202409,
            "color": ["black"],
            "img": [[{"clr0_0.png"}]]
        }
    ],
    "7k": [
        {
            "id": 7,
            "thumbnail": "video7.png",
            "title": "PXW-Z200 + UWP-D27",
            "info": "바디 본품 + UWP-D27 마이크 번들 구성",
            "price": 6690000,
            "date": 202409,
            "color": ["black"],
            "img": [[{"clr0_0.png"}]]
        }
    ]
},
"productList": [
    {
        "prodName": "Camera",
        "prodImg": "main_product_img_1.png",
        "category": "camera"
    },
    {
        "prodName": "Video Camera",
        "prodImg": "main_product_img_2.png",
        "category": "video"
    },
    {
        "prodName": "Lens",
        "prodImg": "main_product_img_3.png",
        "category": "#"
    },
    {
        "prodName": "Audio",
        "prodImg": "main_product_img_4.png",
        "category": "#"
    }
]
```

3-1

구현 결과 (header)



SONY

스토어 추천 제품	제품	기획전	멤버쉽	고객 서비스
추천 제품	카메라	소니스토어 단독	등급&혜택 안내	FAQ&공지사항
선물 제안	비디오 카메라	혜택존		정품등록 안내
교육 할인	렌즈	예약판매		제품 지원
	오디오	정품등록 이벤트		구매 상담
	액세서리	LIVE ON		직영점 안내
				동영상 강좌

Q ☰ ☰

- 카메라 : href = “category.html?category=camera”
 - 비디오 카메라 : href = “category.html?category=video”
 - category.html 로 이동하되, 그 안에서 category키의 값에 따라 렌더링

HTML

```
<!-- PC header -->


<a href="/" class="logo">
    
  </a>
  <nav class="gnb">
    <ul class="gnb_menu">
      <li>
        <a href="#" class="gnb_title" data-num="1">스토어 추천 제품</a>
      </li>
      <li>
        <a href="#" class="gnb_title" data-num="2">제품</a>
      </li>
      <li>
        <a href="#" class="gnb_title" data-num="3">기획전</a>
      </li>
      <li>
        <a href="#" class="gnb_title" data-num="4">멤버쉽</a>
      </li>
      <li>
        <a href="#" class="gnb_title" data-num="5">고객 서비스</a>
      </li>
    </ul>
  </nav>


```

HTML

```
<div class="secondary_menu_bg">
  <div class="secondary_menu">
    <ul class="secondary_menu_list" data-num="1">
      <li>
        <a href="#">추천 제품</a>
      </li>
      <li>
        <a href="#">선물 제안</a>
      </li>
      <li>
        <a href="#">교육 할인</a>
      </li>
    </ul>
    <ul class="secondary_menu_list" data-num="2">
      <li>
        <a href="category.html?category=camera">카메라</a>
      </li>
      <li>
        <a href="category.html?category=video">비디오 카메라</a>
      </li>
      <li>
        <a href="#">렌즈</a>
      </li>
      <li>
        <a href="#">오디오</a>
      </li>
      <li>
        <a href="#">액세서리</a>
      </li>
    </ul>
  </div>

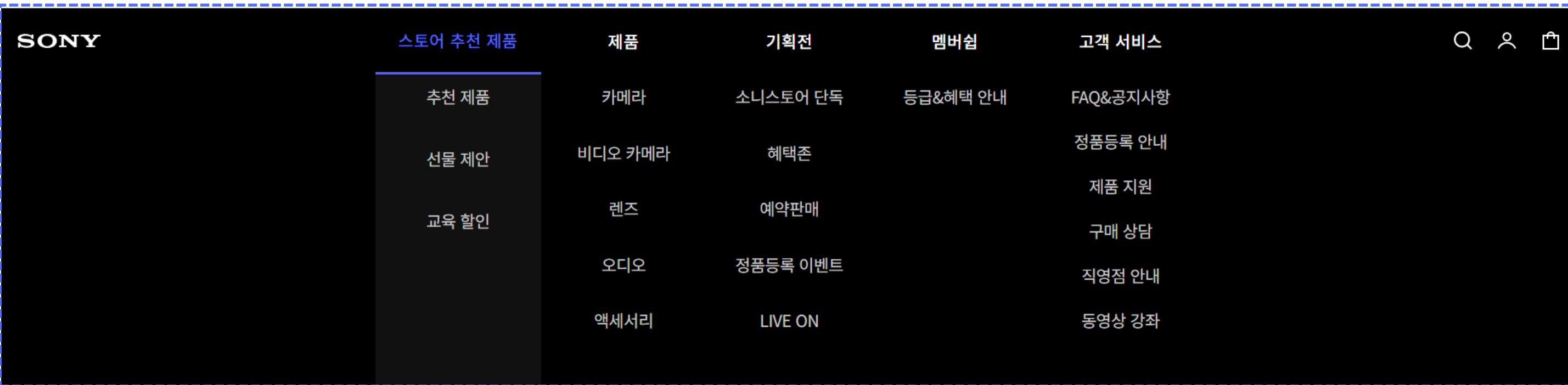
```

HTML

```
<ul class="secondary_menu_list" data-num="3">
  <li>
    <a href="#">소니스토어 단독</a>
  </li>
  <li>
    <a href="#">혜택존</a>
  </li>
  <li>
    <a href="#">예약판매</a>
  </li>
  <li>
    <a href="#">정품등록 이벤트</a>
  </li>
  <li>
    <a href="#">LIVE ON</a>
  </li>
</ul>
<ul class="secondary_menu_list" data-num="4">
  <li>
    <a href="#">등급&혜택 안내</a>
  </li>
</ul>
```

3-1

구현 결과 (header)



HTML

```

<ul class="secondary_menu_list" data-num="5">
  <li>
    <a href="#">FAQ&공지사항</a>
  </li>
  <li>
    <a href="#">정품등록 안내</a>
  </li>
  <li>
    <a href="#">제품 지원</a>
  </li>
  <li>
    <a href="#">구매 상담</a>
  </li>
  <li>
    <a href="#">직영점 안내</a>
  </li>
  <li>
    <a href="#">동영상 강좌</a>
  </li>
</ul>
</div>
</ul>

```

HTML

```

</nav>
<div class="header_pc_icon_menu">
  <a href="#">
    
  </a>
  <a href="#">
    
  </a>
  <a href="#">
    
  </a>
</div>
</div>
<!-- PC header end -->

```

3-1

구현 결과 (header)

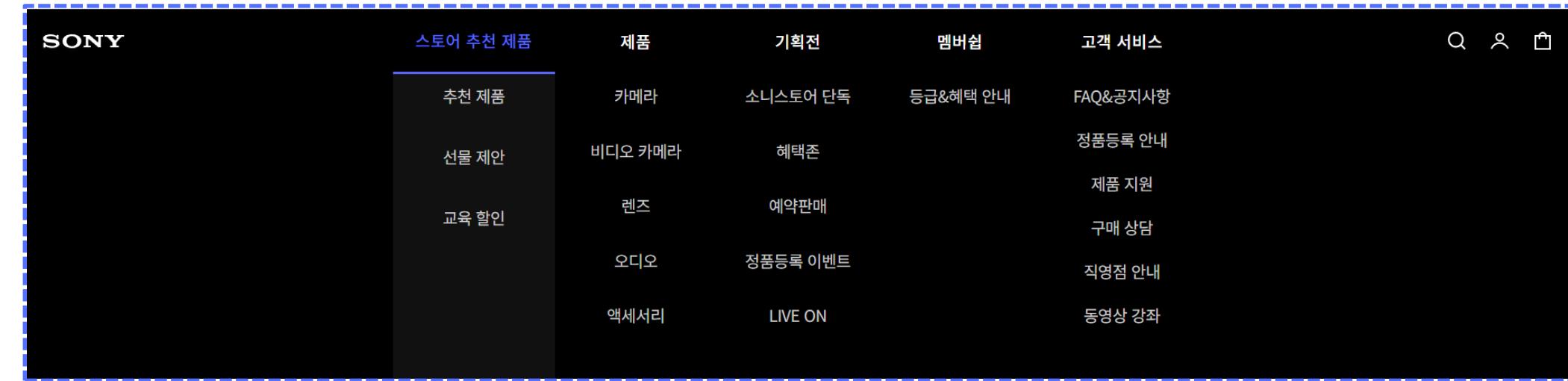
JavaScript

```
const secondary = document.querySelector(".secondary_menu_bg");
const secondaryList = document.querySelectorAll(".secondary_menu_list");
const gnbTitle = document.querySelectorAll(".gnb_title");
const gnbMenuChild = document.querySelectorAll("nav > ul > li > a");
const secondMenuChild = document.querySelectorAll("nav > ul > div > div > ul");

function addEvent(el) {
    for(let i=0;i<el.length;i++) {
        el[i].addEventListener("mouseover", () => {
            secondary.style.height = `${secondary.scrollHeight}px`;
            gnbTitle[i].classList.add("menu_active_border");
            secondaryList[i].classList.add("menu_active");
        });
    }
}

function removeEvent(el) {
    for(let i=0;i<el.length;i++) {
        el[i].addEventListener("mouseout", () => {
            secondary.style.height = `0px`;
            secondaryList[i].classList.remove("menu_active");
            gnbTitle[i].classList.remove("menu_active_border");
        });
    }
}

addEvent(gnbMenuChild);
addEvent(secondMenuChild);
removeEvent(gnbMenuChild);
removeEvent(secondMenuChild);
```



The image shows a screenshot of a Sony website's header. The header features the Sony logo on the left. To the right of the logo is a navigation menu with the following structure:

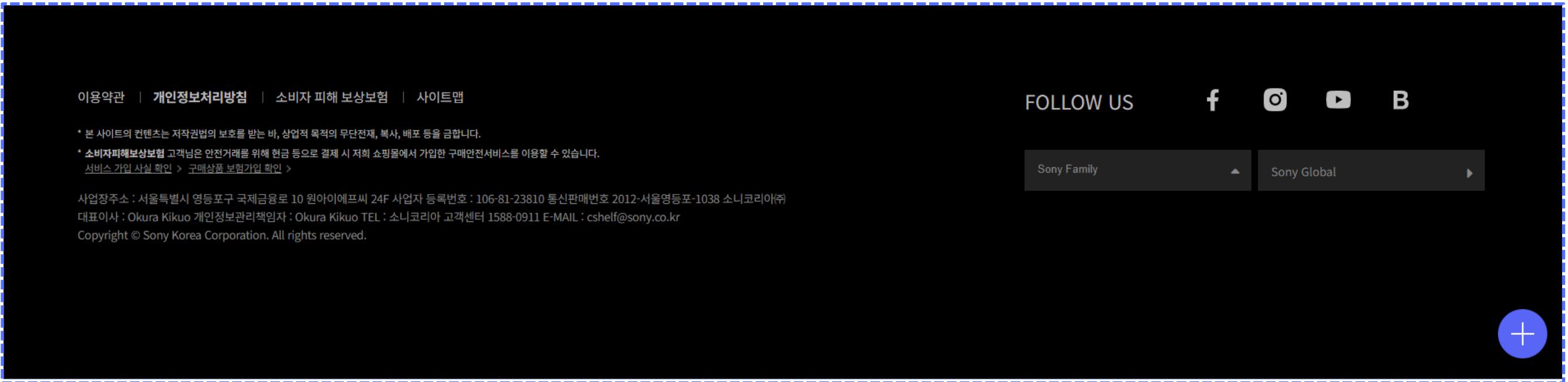
스토어 추천 제품	제품	기획전	멤버쉽	고객 서비스
추천 제품	카메라	소니스토어 단독	등급&혜택 안내	FAQ&공지사항
선물 제안	비디오 카메라	혜택존		정품등록 안내
교육 할인	렌즈	예약판매		제품 지원
	오디오	정품등록 이벤트		구매 상담
	액세서리	LIVE ON		직영점 안내
				동영상 강좌

On the far right of the header, there are three small icons: a magnifying glass (search), a user profile, and a cart.

- ex) depth1의 스토어 추천 제품(a 태그 영역) mouseover 시 depth2 숨김해제, 마우스 올라간 자기자신에게 border-bottom 추가 및 color 변경, depth2의 ui영역 배경색 변경하는 active클래스를 각각 추가
- mouseout 시 각각 추가된 active클래스를 삭제, depth2를 숨김
- depth1의 a 와 depth2의 ui마다 이벤트를 추가하는 함수

3-2

구현 결과 (footer)

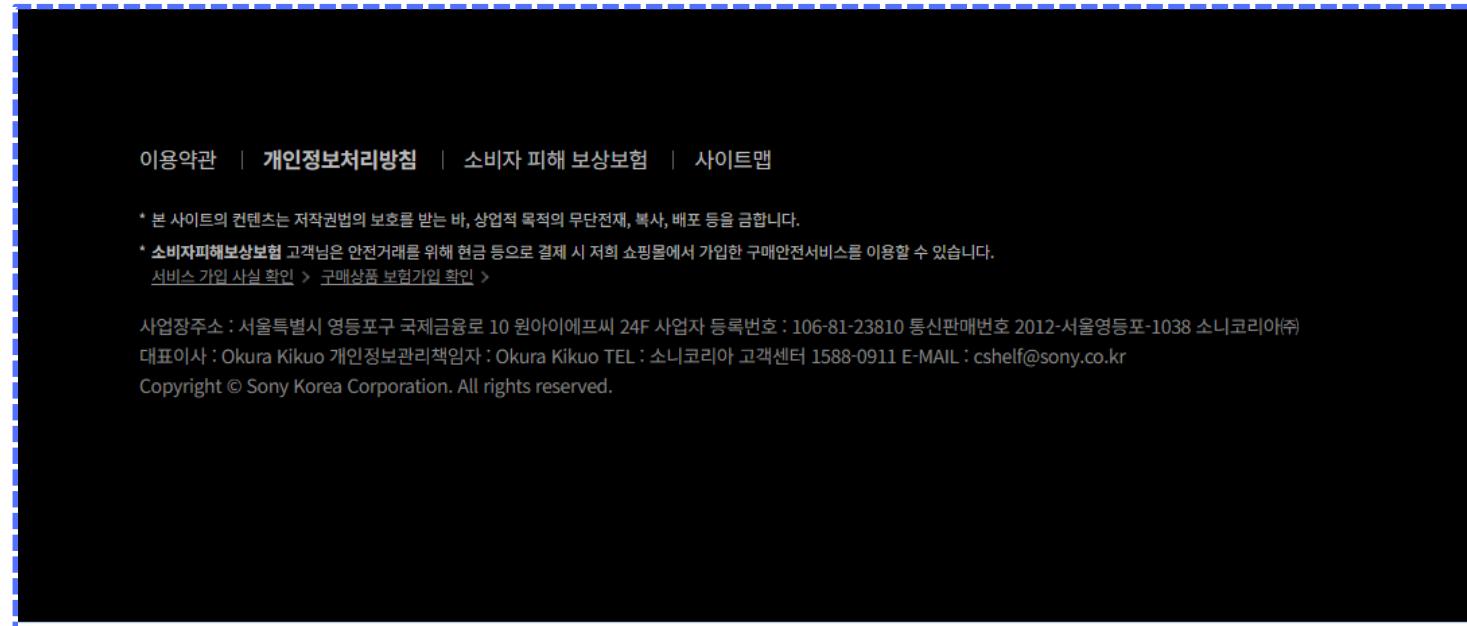


HTML

```
<!-- footer -->
<div class="footer">
  <div class="footer_inner">
    <div class="footer_left">
      <div class="footer_link_box">
        <a href="#" class="footer_link">이용약관</a>
        <a href="#" class="footer_link" style="font-weight:bold">개인정보처리방침</a>
        <a href="#" class="footer_link">소비자 피해 보상보험</a>
        <a href="#" class="footer_link">사이트맵</a>
      </div>
      <div class="footer_warning">
        <p>본 사이트의 컨텐츠는 저작권법의 보호를 받는 바, 상업적 목적의 무단전재, 복사, 배포 등을 금합니다.</p>
        <p><span style="font-weight:bold">소비자 피해 보상보험</span> 고객님은 안전거래를 위해 현금 등으로 결제 시 저희 쇼핑몰에서 가입한 구매안전서비스를 이용할 수 있습니다.</p>
        <div class="footer_confirm link">
          <a href="#">서비스 가입 사실 확인</a>
          <a href="#">구매상품 보험가입 확인</a>
        </div>
      </div>
      <div class="address">
        <span>사업장주소 : 서울특별시 영등포구 국제금융로 10 원아이에프씨 24F</span>
        <span>사업자 등록번호 : 106-81-23810 통신판매번호 2012-서울영등포-1038 소니코리아(주)</span>
        <span>대표이사 : Okura Kikuo</span>
        <span>개인정보관리책임자 : Okura Kikuo</span>
        <span>TEL : 소니코리아 고객센터 1588-0911</span>
        <span>E-MAIL : cshelf@sony.co.kr</span>
        <span>Copyright &copy; Sony Korea Corporation. All rights reserved.</span>
      </div>
    </div>
  </div>
</div>
```

3-2

구현 결과 (footer)



JavaScript

```
/** footer drop up **/ 

const dropUpBtn = document.querySelector('.drop_up_btn');
const dropUpContent = document.querySelector('.sony_family_content');

dropUpBtn.addEventListener("click", e => {
  dropUpContent.classList.toggle("show_family");
});

/** footer drop up end **/
```

HTML

```
<div class="footer_right">
  <div class="footer_social_box">
    <span class="footer_social_title">FOLLOW US</span>
    <div class="footer_social_icon_box">
      <a href="#" class="footer_social_icon">
        
      </a>
      <a href="#" class="footer_social_icon">
        
      </a>
      <a href="#" class="footer_social_icon">
        
      </a>
      <a href="#" class="footer_social_icon">
        
      </a>
    </div>
  </div>
  <div class="footer_family">
    <div class="drop_up">
      <button class="drop_up_btn">Sony Family <i class="fas fa-caret-up"></i></button>
      <div class="sony_family_content">
        <span class="sony_family_title">Sony Family</span>
      </div>
    </div>
  </div>

```

HTML

```
<ul>
  <li>
    <a href="#">소니코리아</a>
  </li>
  <li>
    <a href="#">소니코리아 고객지원</a>
  </li>
  <li>
    <a href="#">소니 알파</a>
  </li>
  <li>
    <a href="#">소니 방송/업무용 솔루션</a>
  </li>
  <li>
    <a href="#">소니 방송 업무용 솔루션 고객지원</a>
  </li>
</ul>
```

HTML

```
<span class="sony_family_title">Family Company</span>
<ul>
  <li>
    <a href="#">소니 인터랙티브 엔터테인먼트 코리아</a>
  </li>
  <li>
    <a href="#">소니 뮤직 엔터테인먼트 코리아</a>
  </li>
  <li>
    <a href="#">소니 ATV 뮤직 퍼블리싱 코리아</a>
  </li>
  <li>
    <a href="#">소니 픽쳐스 엔터테인먼트 코리아</a>
  </li>
  <li>
    <a href="#">소니 픽쳐스 텔레비전 코리아</a>
  </li>
</ul>
</div>
<a class="sony_global_link" href="#">Sony Global <i class="fas fa-caret-right"></i></a>
```

3-2

구현 결과 (footer)

HTML

```
<!-- go top -->


<div class="side_toggle_contents">
    <div class="side_toggle_content go_kakao"><span class="side_toggle_text">카톡 상담</span></div>
    <div class="side_toggle_content go_service"><span class="side_toggle_text">고객 센터</span></div>
    <div class="side_toggle_content go_top"></div>
  </div>
  <div class="side_toggle_btn"></div>
</div>

<!-- footer end -->

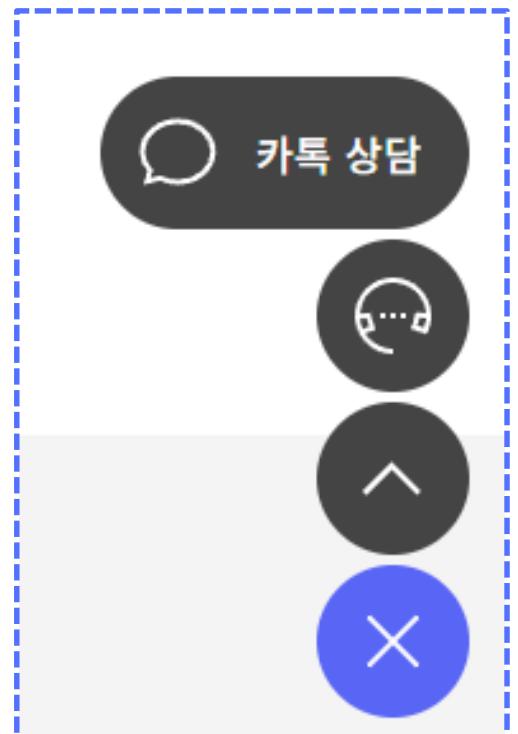

```

JavaScript

```
const sideToggleContents = document.querySelector('.side_toggle_contents');

sideToggleBtn.addEventListener("click", e => {
  sideToggleContents.classList.toggle("side_toggle_contents_show");
  sideToggleBtn.classList.toggle("side_toggle_active");
});

goTop.addEventListener("click", () => {
  window.scrollTo({
    top: 0,
    left: 0,
    behavior: 'smooth'
});
});
```



- 우측 하단 toggle 버튼 클릭 시 active 클래스 추가 및 맨 위로가기 이벤트

3-2

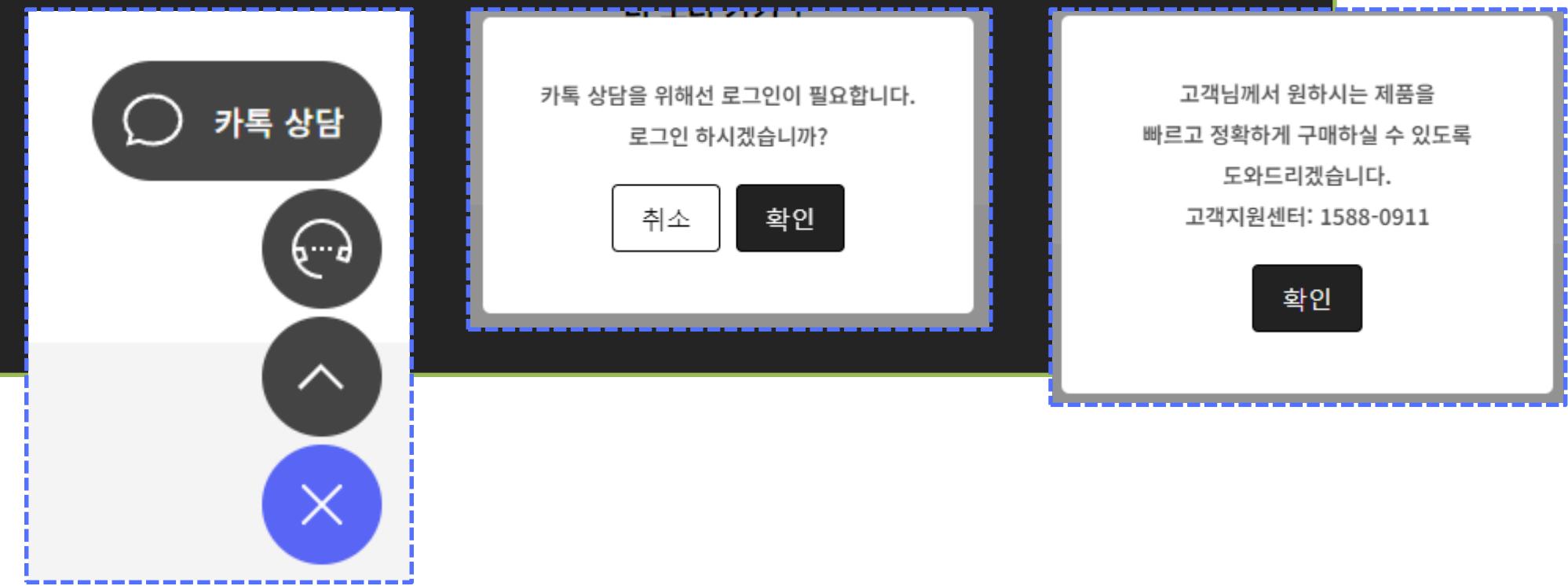
구현 결과 (footer)

JavaScript

```
/** sweetalert (modal) ***/
document.querySelector(".go_service").addEventListener('click', e => {
  new Swal({
    html: '<span class="alert_content">고객님께서 원하시는 제품을<br/> 빠르고 정확하게 구매하실 수 있도록<br/> 도와드리겠습니다.<br/>고객지원센터: 1588-0911</span>',
    confirmButtonColor: '#000',
    confirmButtonText: '확인',
    width: '312px',
    padding: '16px 0 34px 0',
  });
});

document.querySelector(".go_kakao").addEventListener('click', e => {
  new Swal({
    html: '<span class="alert_content">카톡 상담을 위해선 로그인이 필요합니다.<br/>로그인 하시겠습니까?</span>',
    confirmButtonColor: '#000',
    confirmButtonText: '확인',
    showCancelButton: true,
    cancelButtonColor: '#fff',
    cancelButtonText: '취소',
    width: '312px',
    padding: '16px 0 34px 0',
    reverseButtons: true
  });
});

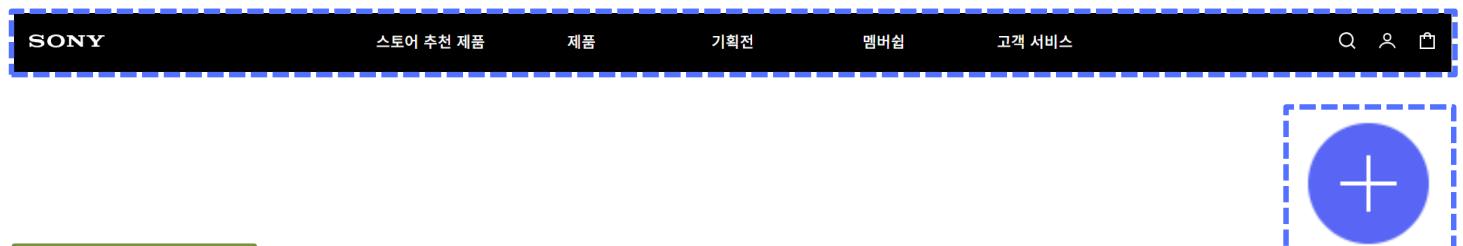
/** sweetalert (modal) end **/
```



- 카톡 상담 및 고객 센터 클릭 시 sweet alert2 라이브러리 창 표시
스크립트

3

구현 결과 (header, footer 공통)



JavaScript

```
/*** nav ***/
/* scrollEvent header && go_top append or hide */
const sideToggleBox = document.querySelector(".side_toggle_box");
const headerPc = document.querySelector(".header_pc");
const sideToggleBtn = document.querySelector(".side_toggle_btn");
const goTop = document.querySelector(".go_top");
let lastScrollY = 0;

window.addEventListener("scroll", () => {
  if(window.scrollY > lastScrollY) {
    //현재 스크롤 위치가 이전 위치보다 클 때 (내려갈 때)
    headerPc.classList.add("header_pc_hide");
    sideToggleBox.classList.add("side_toggle_show");
  } else {
    //현재 스크롤 위치가 이전 위치보다 작을 때 (올라갈 때)
    headerPc.classList.remove("header_pc_hide");
    sideToggleBox.classList.remove("side_toggle_show");
  }

  lastScrollY = window.scrollY;
});
```

- 스크롤 내려갈 때 => nav 숨김, sideButton 표시
- 스크롤 올라갈 때 => nav 표시, sideButton 숨김

HTML

```
<script src=".//assets/js/loadHTML.js"></script>
<script>loadHTML("./header.html");</script>
```

HTML

```
<script>loadHTML("./footer.html");</script>
```

JavaScript

```
function loadHTML(file) {
  let rawFile = new XMLHttpRequest();
  rawFile.open("GET", file, false);
  rawFile.onreadystatechange = function () {
    if(rawFile.readyState === 4)
    {
      if(rawFile.status === 200 || rawFile.status == 0)
      {
        let allText = rawFile.responseText;
        document.write(allText);
      }
    }
  }
  rawFile.send(null);
}
```

- header와 footer는 모듈화하여 호출

3-3

구현 결과 (메인 페이지)



HTML

```
<!-- banner -->


<div class="banner_swiper">
    <div class="swiper-wrapper">
      <div class="swiper-slide">
        
        <div class="banner_contents">
          <div class="banner_title_box">
            <span class="banner_title">기록의</span>
            <span class="banner_title">완성</span>
          </div>
          <span class="banner_item">ZV-E10M2</span>
          <span class="banner_view">자세히 보기 <i class="fas fa-angle-right"></i></span>
        </div>
      </div>
      <div class="swiper-slide">
        
        <div class="banner_contents">
          <div class="banner_title_box">
            <span class="banner_title">공부할</span>
            <span class="banner_title">땐</span>
          </div>
          <div class="banner_title_box">
            <span class="banner_title">스터디헤드폰</span>
          </div>
        </div>
      </div>
    </div>
  </div>

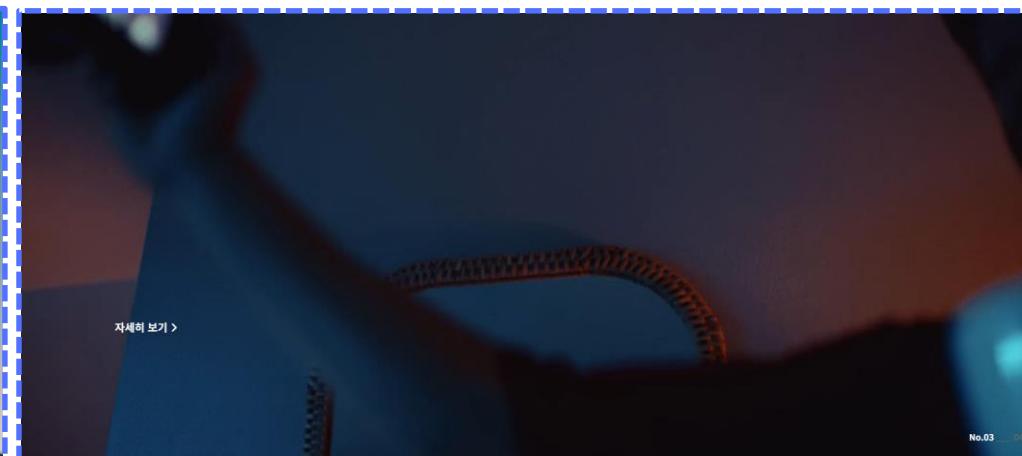

```

HTML

```
  <span class="banner_item">WH-1000XM5</span>
  <span class="banner_view">자세히 보기 <i class="fas fa-angle-right"></i></span>
</div>
</div>
<div class="swiper-slide">
  <video src="assets/img/banner/03_PC_main_vedio.mp3" loop autoplay="true" muted></video>
  <div class="banner_contents" style="top:70%">
    <span class="banner_view">자세히 보기 <i class="fas fa-angle-right"></i></span>
  </div>
</div>
<div class="swiper-slide">
  
  <div class="banner_contents" style="color: black">
    <div class="banner_title_box">
      <span class="banner_title">WF-C510</span>
    </div>
    <span class="banner_item">일상을 함께하는 나의 데일리 이어폰</span>
    <span class="banner_view">자세히 보기 <i class="fas fa-angle-right"></i></span>
  </div>
</div>
<div class="swiper-button-prev"></div>
<div class="swiper-button-next"></div>
<div class="swiper-pagination"></div>
</div>
<!-- banner end -->
```

3-3

구현 결과 (메인 페이지)



JavaScript

```
/** banner swiper ***/
let loop_slider = true;
const num_of_slides = 4;
let swiper_sliders = document.querySelectorAll(".banner_swiper");

for (const this_slider of swiper_sliders){
    let num_of_slides_before_init = this_slider.querySelectorAll(".swiper-slide").length;
    if(num_of_slides_before_init < num_of_slides){
        loop_slider = false;
    }

    let swiper = new Swiper(this_slider, {
        slidesPerView: 1,
        loop: loop_slider,
        allowTouchMove: false,
        speed:800,
        navigation: {
            prevEl: '.swiper-button-prev',
            nextEl: '.swiper-button-next',
        },
        pagination: {
            el: ".swiper-pagination",
            type: 'custom',
            renderCustom: function (swiper, current, total) {
                return 'No.' + current + '<span style="color:#666">' + '&ampnbsp' + '0' + total + '</span>';
            }
        },
    });
}

/** banner swiper end **/
```



- swiper 라이브러리 적용
- fullSize 이미지, 비디오 삽입
- 배너의 왼쪽과 오른쪽 구역을 각각 50%로 나누어 마우스 호버시 마우스커서가 왼쪽, 오른쪽화살표 이미지로 바뀌어 슬라이드 가능함을 시각적으로 표시

3-3

구현 결과 (메인 페이지)



HTML

```
<div class="wrap">
  <div class="box style02">
    <!--main-->

    <!-- dual swiper-->
    <div class="swiper-container main-slide">
      <div class="swiper-wrapper">
        <div class="swiper-slide">
          <div class="swiper-slide-container main_img">
            
          </div>
        </div>
        <div class="swiper-slide">
          <div class="swiper-slide-container main_img">
            
          </div>
        </div>
        <div class="swiper-slide">
          <div class="swiper-slide-container main_img">
            
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

HTML

```
<div class="box style01">
  <div class="lower pointer">
    <div class="sub_wrap">
      <div class="sub_inner">
        <div class="swiper-container sub-slide">
          <div class="swiper-wrapper">
            <div class="swiper-slide column">
              <span class="sub_swiper_title opacity_1">기록의 완성</span>
              <a href="view.html?id=1&path1=camera&path2=lens_change&path3=APS-C" class="swiper-slide-container sub_img">
                
              </a>
              <span class="sub_swiper_name opacity_1">ZV-E10M2</span>
              <span class="sub_swiper_desc opacity_1">기록의 완성</span>
            </div>
            <div class="swiper-slide column">
              <span class="sub_swiper_title">Best 노이즈 캔슬링</span>
              <div class="swiper-slide-container sub_img">
                
              </div>
              <span class="sub_swiper_name">WH-1000XM5</span>
              <span class="sub_swiper_desc">#물입의 즐거움 #스터디해드폰</span>
            </div>
            <div class="swiper-slide column">
              <span class="sub_swiper_title">1000X 이어폰</span>
              <div class="swiper-slide-container sub_img">
                
              </div>
              <span class="sub_swiper_name">WF-1000XM5</span>
              <span class="sub_swiper_desc">이것은 작게 진화한 핸드폰</span>
            </div>
          </div>
          <div class="swiper-button-next"></div>
        </div>
      </div>
    </div>
  </div>
<!-- dual swiper end-->
```

3-3

구현 결과 (메인 페이지)



JavaScript

```
/*** dual_swiper ***/
let swiperTopNum = $('.main-slide').find('.swiper-slide');
let swiperSubNum = $('.sub-slide').find('.sub-slide');

let mainSlide = new Swiper('.main-slide', {
  initialSlide: 2, // 슬라이드 시작지점
  loop: true, //슬라이드 반복
  loopedSlides: swiperTopNum.length,
  touchRatio: 0 // 슬라이드 control 막음
});

let subSlide = new Swiper('.sub-slide', {
  spaceBetween: 10, //슬라이드 간격
  slidesPerView: '1.5', //한번에 보여지는 슬라이드 개수
  // slideToClickedSlide: true, //클릭 시 해당 슬라이드 위치로 이동
  loop: true, //슬라이드 반복
  speed:800,
  loopedSlides: swiperSubNum.length, //loop 시 파라미터 duplicate 개수
  navigation: {
    nextEl: '.swiper-button-next'
  },
  controller: {
    control: mainSlide,
  }
});
```

JavaScript

- 좌측 이미지와 우측 이미지에 swiper라이브러리 적용
- 우측 이미지 슬라이드시 좌측 이미지도 함께 슬라이드
- 우측 제목, 상품명, 상품설명은 현재 슬라이드 위치만 표시하고 나머지 슬라이드에서는 숨김처리
- 상품 클릭시 상세페이지로 이동 (swiper-active 클래스의 add, remove를 감지하여 변경이 있을 때마다 함수 실행)
- 우측 슬라이드의 미리보기 슬라이드에 마우스 호버시 커서모양 화살표 모양으로 변경

```
//swiper title, desc opacity
const SubSlide = document.querySelectorAll('.column');
const SubSlideTitle = document.querySelectorAll('.sub_swiper_title');
const SubSlideName = document.querySelectorAll('.sub_swiper_name');
const SubSlideDesc = document.querySelectorAll('.sub_swiper_desc');

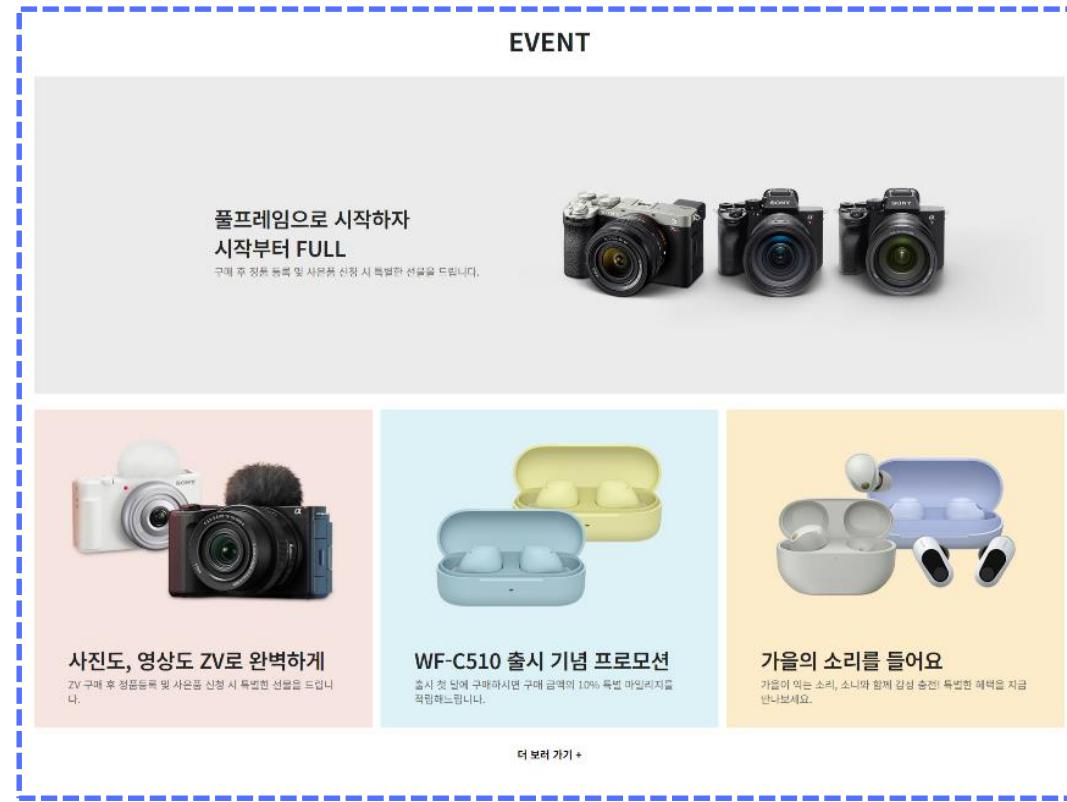
// 클래스 변경을 감지할 콜백 함수
const callback = function(mutationsList) {
  mutationsList.forEach((mutation) => {
    if (mutation.type === 'attributes' && mutation.attributeName === 'class') {
      SubSlide.forEach((slide, index) => {
        const hasClass = slide.classList.contains('swiper-slide-active');

        if (hasClass) {
          SubSlideTitle[index].classList.add('opacity_1');
          SubSlideName[index].classList.add('opacity_1');
          SubSlideDesc[index].classList.add('opacity_1');
        } else {
          SubSlideTitle[index].classList.remove('opacity_1');
          SubSlideName[index].classList.remove('opacity_1');
          SubSlideDesc[index].classList.remove('opacity_1');
        }
      });
    }
  });
};

// 각 슬라이드에 대해 MutationObserver를 설정
SubSlide.forEach((slide) => {
  const observer = new MutationObserver(callback);
  observer.observe(slide, { attributes: true });
});
```

3-3

구현 결과 (메인 페이지)



- 아래 아이템박스 내 이미지 호버시 이미지cale 1.1배

HTML

```

<!-- event section-->
<div class="section_pd">
  <h2>EVENT</h2>

  <div class="event_box_w100 pointer">
    <div class="event_contents">
      <span class="event_title">풀프레임으로 시작하자</span>
      <span class="event_title">시작부터 FULL</span>
      <span class="event_info">구매 후 정품 등록 및 사은품 신청 시 특별한 선물을 드립니다.</span>
    </div>
  </div>
  <div class="event_item_container">
    <div class="event_item_w33 pointer">
      <span class="event_title">사진도, 영상도 ZV로 완벽하게</span>
      <span class="event_info">ZV 구매 후 정품등록 및 사은품 신청 시 특별한 선물을 드립니다.</span>
    </div>
    <div class="event_item_w33 pointer">
      <span class="event_title">WF-C510 출시 기념 프로모션</span>
      <span class="event_info">출시 첫 달에 구매하시면 구매 금액의 10% 특별 마일리지를 적립해드립니다.</span>
    </div>
    <div class="event_item_w33 pointer">
      <span class="event_title">가을의 소리를 들어요</span>
      <span class="event_info">가을이 익는 소리, 소니와 함께 깊은 충전! 특별한 혜택을 지금 만나보세요.</span>
    </div>
  </div>
  <a href="#" class="event_more_view">더 보러 가기 +</a>
</div>

<div class="gray_bg_section pd_6vw">
  <h3>PRODUCT</h3>
  <div class="product_container">
  </div>
</div>
<!-- event section end-->

```

3-3

구현 결과 (메인 페이지)



JavaScript

```

(async (e) => {
  let response = null;

  try {
    response = await axios.get('backend/data.json');
  } catch (error) {
    console.error(`[Error Code] ${error.code}`);
    console.error(`[Error message] ${error.message}`);
    let errorMsg = error.message;

    if(error.response !== undefined) {
      const errorText = `${error.response.statusText} - ${error.response.statusText}`;
      console.error(`[HTTP Status] ${errorText}`);
      errorMsg += `\n${errorText}`;
    }
  }

  alert(errorMsg);
  return;
}

const prodContainer = document.querySelector(".product_container");
const productList = response.data.productList;
  
```

- async 를 이용한 비동기처리로 받아온 json파일의 productList의 배열길이 만큼 반복하여 상품을 표시
- PRODUCT 아이템박스에 호버시 이미지 가운데로 이동 및 배경색 변경
- 클릭시 상품목록페이지로 이동

JavaScript

```

const prodContainer = document.querySelector(".product_container");
const productList = response.data.productList;

productList.forEach((v, i) => {
  const prodItem = document.createElement("a");
  const prodName = document.createElement("span");

  prodItem.classList.add("product_item");
  prodItem.setAttribute('href', `category.html?category=${v.category}`);
  prodItem.classList.add("pointer");
  prodName.innerHTML = v.prodName;

  prodContainer.appendChild(prodItem);
  prodItem.appendChild(prodName);
  prodItem.style.backgroundImage = `url(assets/img/${v.prodImg})`;

});
  
```

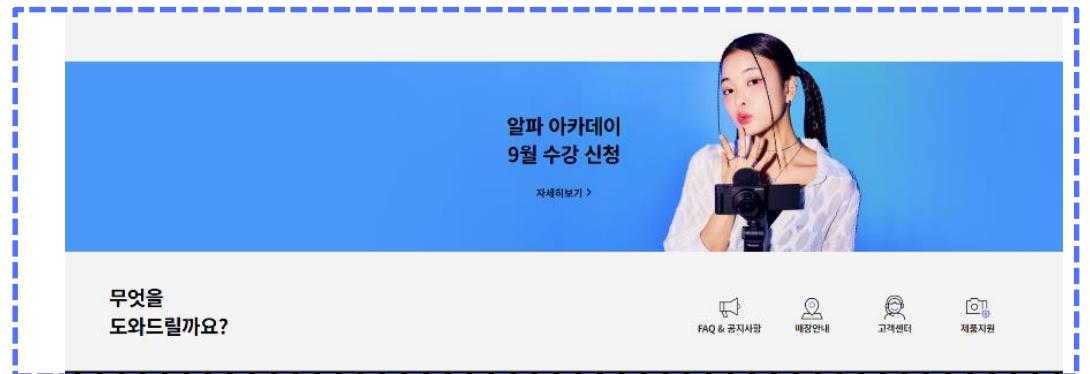
HTML

```

<!-- product section-->
<div class="gray_bg_section pd_6vw">
  <h3>PRODUCT</h3>
  <div class="product_container">
  </div>
</div>
<!-- product section end -->
  
```

3-3

구현 결과 (메인 페이지)



HTML

```
<!-- hr banner section -->


<div class="hr_banner">
    <div class="hr_banner_contents">
      <span class="hr_banner_text">알파 아카데미<br />9월 수강 신청</span>
      <a href="#" class="event_more_view">자세히보기&ampnbsp<i class="fas fa-angle-right"></i></a>
    </div>
  </div>
<!-- hr banner section end -->


```

HTML

```
<!-- help section -->

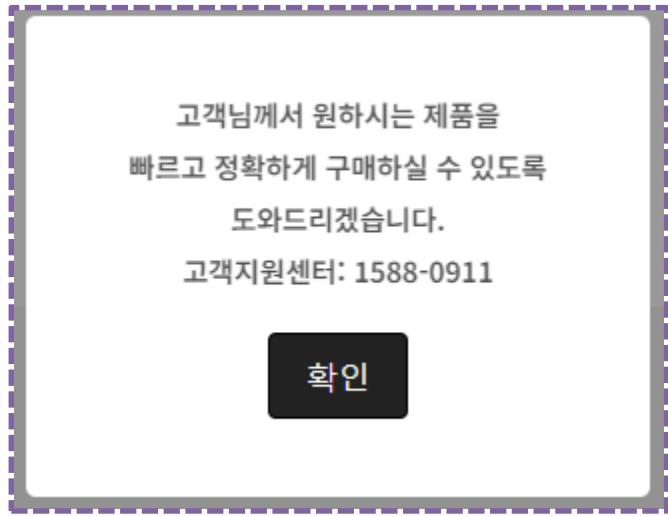
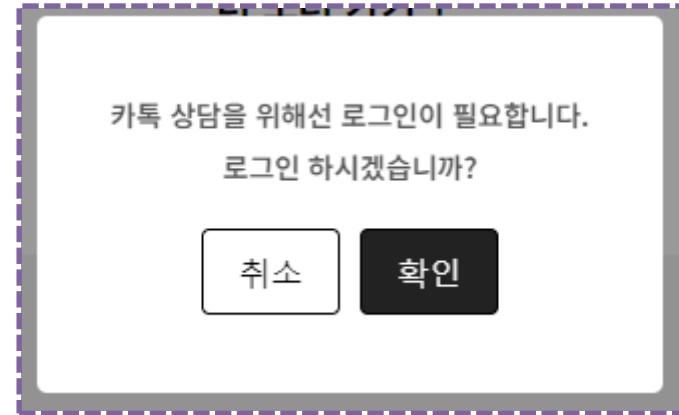

<div class="main_help">
    <span class="main_help_banner_text">무엇을<br />도와드릴까요?</span>
    <ul class="main_help_lists">
      <li class="main_help_list notice pointer">
        <a href="#">FAQ & 공지사항</a>
      </li>
      <li class="main_help_list location pointer">
        <a href="#">매장안내</a>
      </li>
      <li class="main_help_list service pointer">
        <a href="#">고객센터</a>
      </li>
      <li class="main_help_list advice pointer">
        <a href="#">제품지원</a>
      </li>
    </ul>
  </div>
<!-- help section end -->


```

3-3

구현 결과 (사용한 라이브러리)

sweetalert2



```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/sweetalert2/11.13.1/sweetalert2.min.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/sweetalert2/11.13.1/sweetalert2.all.min.js"></script>
```

- css, js 파일 cdn을 입력하고 Swal 속성 값을 조정하여 사용

swiper

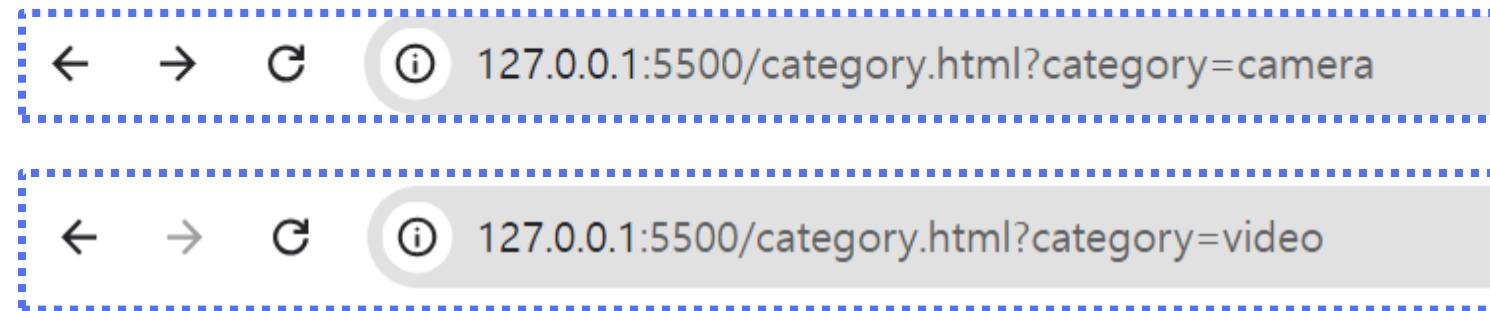


```
<link href="https://cdn.jsdelivr.net/npm/swiper@8.4.5/swiper-bundle.min.css" rel="stylesheet" />
<script src="https://cdn.jsdelivr.net/npm/swiper@8/swiper-bundle.min.js"></script>
```

- css, js 파일 cdn을 입력하고 Swiper 속성 값을 조정하여 사용

3-4

구현 결과 (상품 목록 페이지)



JavaScript

```
let product = {} // 전역 변수 선언

window.onload = async (e) => [
  let response = null;

  try {
    response = await axios.get('http://localhost:3001/products'); // 승현
    product = response.data; // 데이터 로딩 후 전역 변수에 할당
  } catch (error) {
    console.error(`[Error Code] ${error.code}`);
    console.error(`[Error message] ${error.message}`);
    let errorMsg = error.message;

    // HTTP 상태 메시지가 포함되어 있다면 해당 내용을 에러 문자열에 추가
    if (error.response !== undefined) {
      const errorText = `${error.response.status} ${error.response.statusText}`;
      console.error(`[HTTP Status] ${errorText}`);
      errorMsg += `\n${errorText}`;
    }

    alert(errorMsg);
    return;
  }
]
```

3-4

구현 결과 (상품 목록 페이지)



HTML

```
<!-- 메인 영역 -->


<!-- navbar 하단 사진영역 -->
  <div class="category" id="categoryBackground">
    <!-- 동적으로 변경될 부분 -->
    <a href="#" class="go_back"></a>
    <!-- 사진영역 제목 -->
    <h1 class="category_name"></h1>
    <!-- 사진 영역 박스 -->
    <div class="category_list">
      <ul class="list">
        <!-- 동적으로 생성될 리스트 -->
      </ul>
    </div>
  </div>


```

JavaScript

```
// URL 파라미터에서 카테고리, 서브 카테고리, 서브 서브 카테고리 키 가져오기
const params = new URLSearchParams(location.search);
const categoryKey = params.get('category') || 'camera'; // 기본 카테고리는 'camera'
let subCategoryKey = params.get('subCategory'); // 서브 카테고리 키
let subSubCategoryKey = params.get('subSubCategory'); // 서브 서브 카테고리 키

// 카테고리 이름 맵핑 (영어 키를 한글로 변환)
const categoryNames = {
  'camera': '카메라',
  'video': '비디오 카메라'
};

// 카테고리 및 서브 카테고리에 따른 배경 이미지 맵핑 객체를 생성
const backgroundImages = {
  camera: {
    default: 'bg_cat_camera.jpg',
    lens_change: 'bg_lens_change.jpg',
    compact: 'bg_compact_camera.jpg'
  },
  video: {
    default: 'bg_cat_video.jpg',
    cinema: 'bg_cinema_camera.jpg',
    camcorder: 'bg_camcorder.jpg'
  }
};
```

3-4

구현 결과 (상품 목록 페이지)

JavaScript

```
// 배경 이미지를 설정하는 함수
function updateBackgroundImage() {
  const category = document.querySelector(".category");
  const categoryBackground = document.getElementById('categoryBackground');
  const categoryList = document.querySelector(".category_list");
  const categoryLink = document.querySelector(".category_link");
  const goBack = document.querySelector(".go_back");

  let imageUrl = '';

  if (subCategoryKey && backgroundImages[categoryKey][subCategoryKey]) {
    // 서브 카테고리의 배경 이미지 설정
    imageUrl = `url(assets/img/background/${backgroundImages[categoryKey][subCategoryKey]})`;
    category.style.padding = "110px 0 0";
    categoryList.classList.add("display_none");
    categoryLink.classList.remove("display_none");
    goBack.classList.remove("display_none");
  } else if (backgroundImages[categoryKey].default) {
    // 카테고리 메인 페이지의 배경 이미지 설정
    imageUrl = `url(assets/img/background/${backgroundImages[categoryKey].default})`;
    category.style.padding = "110px 0 124px";
    categoryList.classList.remove("display_none");
    categoryLink.classList.add("display_none");
    goBack.classList.add("display_none");
  } else {
    // 기본 배경 이미지 설정 (필요에 따라 수정)
    imageUrl = 'none';
  }

  categoryBackground.style.backgroundImage = imageUrl;
}
```

JavaScript

```
// 서브 카테고리 이름 매핑 함수
function getSubCategoryName(subCategoryKey) {
  const subCategoryNames = {
    'lens_change': '렌즈교환식 카메라',
    'compact': '콤팩트 카메라',
    'cinema': '시네마 라인 카메라',
    'camcorder': '캠코더'
  };
  return subCategoryNames[subCategoryKey] || subCategoryKey; // 매핑이 없으면 원래 키 반환
}
```

JavaScript

```
// 서브 카테고리 키와 이미지 파일명을 매핑하여 이미지 경로를 생성
function getSubCategoryImage(subCategoryKey) {
  return `assets/img/subcategory/icon_${subCategoryKey}.png`;
}
```

JavaScript

```
// 렌더링 함수들 호출
renderCategoryList(); // 카테고리 리스트 렌더링
updateBackgroundImage(); // 배경 이미지 업데이트
```

3-4

구현 결과 (상품 목록 페이지)

JavaScript

```
// 카테고리 리스트 렌더링 함수
function renderCategoryList() {
  const categoryList = document.querySelector('.category_list .list');
  categoryList.innerHTML = ''; // 기존 리스트 초기화

  // '전체보기' 버튼 추가
  const allItem = document.createElement('li');
  allItem.innerHTML = `<a href="?category=${categoryKey}" class="All">전체보기</a>`;
  categoryList.appendChild(allItem);

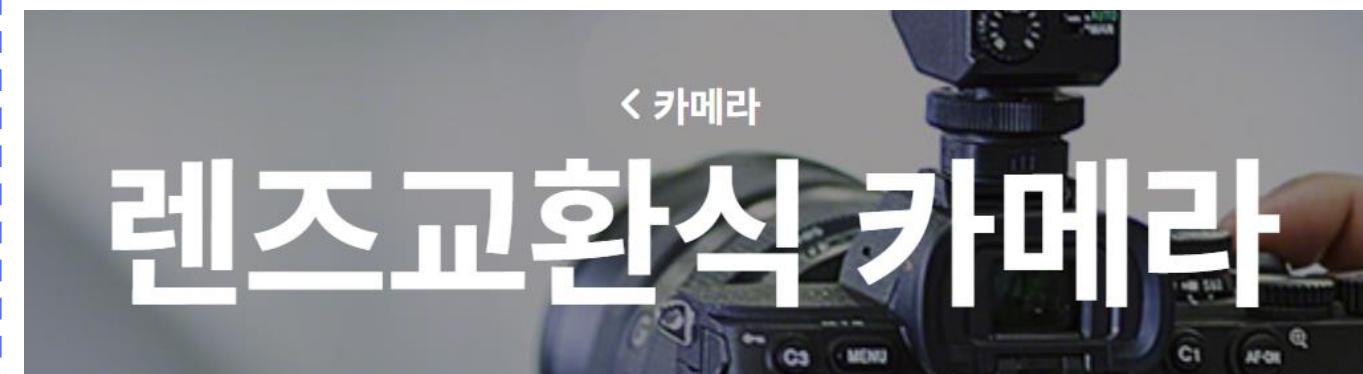
  const categoryData = product[categoryKey]; // 현재 카테고리의 데이터
  for (const subCategory in categoryData) {
    const li = document.createElement('li');
    const imgSrc = getSubCategoryImage(subCategory);
    li.innerHTML = `<a href="?category=${categoryKey}&subCategory=${subCategory}" class="${subCategory}-filter">${getSubCategoryName(subCategory)}</a>`;
    categoryList.appendChild(li); // 서브 카테고리 리스트에 추가
  }
}
```

JavaScript

```
// 카테고리 리스트 클릭 이벤트
document.querySelector('.category_list .list').addEventListener('click', e => {
  e.preventDefault();
  const target = e.target.closest('a');
  if (target) {
    window.location.href = target.href; // 해당 링크로 이동하여 페이지 갱신
  }
});
```

3-4

구현 결과 (상품 목록 페이지)



JavaScript

```
// 페이지에 카테고리 이름 업데이트
document.querySelector('.go_back').innerHTML = '<i class="fas fa-angle-left"></i> ' + categoryNames[categoryKey]; // 'go_back' 버튼 텍스트 설정

const categoryNameElement = document.querySelector('.category_name'); //서브 카테고리 이름으로 업데이트
if (subCategoryKey) {
  const subCategoryName = getSubCategoryName(subCategoryKey);
  categoryNameElement.textContent = subCategoryName;
} else {
  categoryNameElement.textContent = categoryNames[categoryKey];
}
```

- 서브 카테고리에 따라 배경 및 글씨 변경

JavaScript

```
// 'go_back' 버튼 클릭 이벤트
document.querySelector('.go_back').addEventListener('click', e => {
  e.preventDefault();
  window.location.href = '?category=' + categoryKey; // 현재 카테고리의 메인 페이지로 이동
});
```

3-4

구현 결과 (상품 목록 페이지)

카메라



- 렌즈교환식 카메라 버튼을 눌렀을 때만 탭 메뉴가 나옴

HTML

```
<div class="category_link">
  <a href="#">제품 비교</a>
  <a href="#">연장 서비스 플랜 ESP 보기</a>
</div>
```

전체보기

풀프레임

APS-C

제품 비교

연장 서비스 플랜 ESP 보기

비디오 카메라



- 캠코더 버튼을 눌렀을 때만 탭 메뉴가 나옴

HTML

```
<!-- 탭메뉴 영역 -->
<div class="tapMenu_wrapper">
  <ul class="tapMenu_list">
    <!-- 동적으로 생성될 탭 메뉴 -->
  </ul>
</div>
```

전체보기

4K 핸디캠

4K 핸드헬드

제품 비교

연장 서비스 플랜 ESP 보기

3-4

구현 결과 (상품 목록 페이지)

JavaScript

```
// 서브 서브 카테고리 이름 매핑 함수
function getSubSubCategoryName(subSubCategoryKey) {
  const subSubCategoryNames = {
    'full_frame': '풀프레임',
    'APS-C': 'APS-C',
    '4k_handy': '4K 핸디캠',
    '4k_handheld': '4K 핸드헬드'
  };
  return subSubCategoryNames[subSubCategoryKey] || subSubCategoryKey; // 매핑이 없으면 원래 키 반환
}
```

JavaScript

```
renderTapMenu(); // 탭 메뉴 렌더링
```

```
// 탭 메뉴 렌더링 함수
function renderTapMenu() {
  const tapMenuWrapper = document.querySelector('.tapMenu_wrapper');
  const tapMenuList = document.querySelector('.tapMenu_list');
  tapMenuList.innerHTML = ''; // 기존 탭 메뉴 초기화

  if (!subCategoryKey) {
    tapMenuWrapper.style.display = 'none'; // 서브 카테고리가 없으면 탭 메뉴 숨김
    return;
  }

  const categoryData = product[categoryKey];
  const subCategoryData = categoryData[subCategoryKey];

  if (typeof subCategoryData === 'object' && !Array.isArray(subCategoryData)) {
    // 서브 서브 카테고리가 있는 경우
    tapMenuWrapper.style.display = 'block'; // 탭 메뉴 표시

    // '전체보기' 탭 추가
    const allTab = document.createElement('li');
    allTab.innerHTML = `<a href="?category=${categoryKey}&subCategory=${subCategoryKey}" class="all-filter"><span>전체보기</span></a>`;
    tapMenuList.appendChild(allTab);

    // 활성화 상태 설정: subSubCategoryKey가 없으면 '전체보기' 탭 활성화
    if (!subSubCategoryKey) {
      allTab.querySelector('a').classList.add('active'); // '전체보기' 탭 활성화
    }
  }
  // 각 서브 서브 카테고리별 탭 추가
  for (const subSubCategory in subCategoryData) {
    const li = document.createElement('li');
    li.innerHTML = `<a href="?category=${categoryKey}&subCategory=${subCategoryKey}&subSubCategory=${encodeURIComponent(subSubCategory)}" class="${subSubCategory}-filter"><span>${getSubSubCategoryName(subSubCategory)}</span></a>`;
    tapMenuList.appendChild(li);
  }
} else {
  tapMenuWrapper.style.display = 'none'; // 서브 서브 카테고리가 없으면 탭 메뉴 숨김
}
```

JavaScript

3-4

구현 결과 (상품 목록 페이지)

전체보기

풀프레임

APS-C

JavaScript

```
// 탭 메뉴에서 선택된 카테고리에 따라 해당 버튼에 활성화 상태를 설정하는 함수
function activateTab(tabElement) {
  document.querySelectorAll('.tapMenu_list a').forEach(v => {
    v.classList.remove('active'); // 모든 버튼의 active 클래스 제거
  });
  tabElement.classList.add('active'); // 클릭된 버튼에만 active 클래스 추가
}
```

JavaScript

```
// 탭 메뉴 클릭 이벤트
document.querySelector('.tapMenu_list').addEventListener('click', e => {
  e.preventDefault();
  const target = e.target.closest('a');
  activateTab(e.target.closest('a'));
  if (target) {
    // 링크의 URL에서 파라미터 추출
    const url = new URL(target.href, location.origin);
    const params = new URLSearchParams(url.search);

    // 파라미터 값 업데이트
    subSubCategoryKey = params.get('subSubCategory') || null;

    // 페이지 콘텐츠 업데이트
    renderProducts();
  }
});
```

- renderProducts 함수는
뒤에 설명할 제품 렌더링
함수임

3-4

구현 결과 (상품 목록 페이지)

제품(5) 최신순 높은가격순 낮은가격순

- 제품 개수 반영
- 정렬 버튼 클릭 시 액티브 적용

HTML

```
<div class="list_info_wrapper">
  <div class="list_info">
    <div class="list_info_name">제품</div>
    <div class="list_info_num"></div>
  </div>
  <ul class="item_sort">
    <li><a href="#" class="sort sort_recent sort_active">최신순</a></li>
    <li><a href="#" class="sort sort_high_price">높은가격순</a></li>
    <li><a href="#" class="sort sort_low_price">낮은가격순</a></li>
  </ul>
</div>
```

JavaScript

```
// 제품 개수를 업데이트하는 함수
function updateProductCount(count) {
  const productCount = document.querySelector('.list_info_num');
  productCount.innerHTML = `${count}`; // 제품 개수 표시
}
```

JavaScript

```
// 가격순 정렬 함수
function sortProductsByPrice(order) {
  const sortedList = [...currentList].sort((a, b) => {
    if (order === 'high_to_low') {
      return b.price - a.price; // 높은 가격순
    } else {
      return a.price - b.price; // 낮은 가격순
    }
  });
  renderProductList(sortedList); // 정렬된 리스트 렌더링
}

// 최신순 정렬 함수
function sortProductsByDate() {
  const sortedList = [...currentList].sort((a, b) => b.date - a.date); // 최신순
  renderProductList(sortedList); // 정렬된 리스트 렌더링
}
```

JavaScript

```
// 정렬 버튼 클릭 이벤트 설정
document.querySelector('.sort_high_price').addEventListener('click', e => {
  e.preventDefault();
  sortProductsByPrice('high_to_low');
});

document.querySelector('.sort_low_price').addEventListener('click', e => {
  e.preventDefault();
  sortProductsByPrice('low_to_high');
});

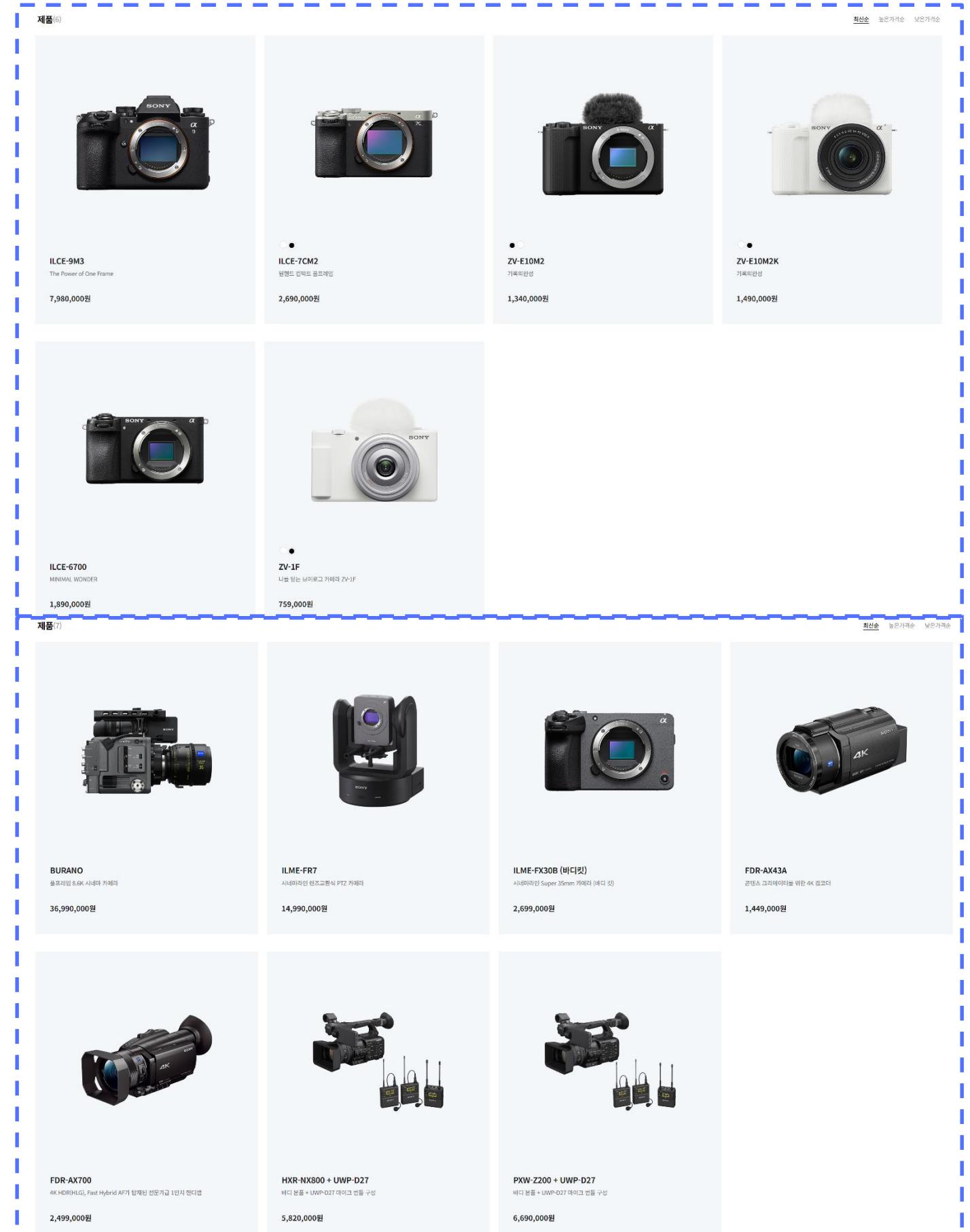
document.querySelector('.sort_recent').addEventListener('click', e => {
  e.preventDefault();
  sortProductsByDate();
});
```

JavaScript

```
// 정렬 버튼 active 추가
document.querySelectorAll('.sort').forEach( (v, i) => {
  v.addEventListener("click", (e) => {
    document.querySelectorAll('.sort').forEach( (v, i) => {
      v.classList.remove("sort_active");
    });
    e.currentTarget.classList.add("sort_active");
  })
});
```

3-4

구현 결과 (상품 목록 페이지)



HTML

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

JavaScript

```
renderProducts(); // 제품 리스트 렌더링
```

JavaScript

```
// DOM 요소 선택
const list_container = document.querySelector('.list_container'); // 제품 리스트를 표시할 컨테이너
let currentList = []; // 현재 표시되는 제품 리스트를 저장하는 배열
```

JavaScript

```
// 제품 렌더링 함수
function renderProducts() {
  currentList = []; // 현재 리스트 초기화
  const categoryData = product[categoryKey];

  if (subCategoryKey) {
    const subCategoryData = categoryData[subCategoryKey];

    if (subSubCategoryKey) {
      // 특정 서브 서브 카테고리의 제품만 표시
      const productList = subCategoryData[subSubCategoryKey];
      if (productList) {
        currentList.push(...productList); // 제품 리스트에 추가
      }
    } else {
      if (Array.isArray(subCategoryData)) {
        // 서브 카테고리가 배열인 경우 (제품 리스트)
        currentList.push(...subCategoryData);
      } else {
        // 서브 서브 카테고리가 있는 경우 모든 제품 수집
        for (const subSubCategory in subCategoryData) {
          const productList = subCategoryData[subSubCategory];
          currentList.push(...productList);
        }
      }
    }
  } else {
    // 서브 카테고리가 없는 경우 카테고리 내 모든 제품 수집
    for (const subCategory in categoryData) {
      const subCategoryData = categoryData[subCategory];
      if (Array.isArray(subCategoryData)) {
        currentList.push(...subCategoryData);
      } else {
        for (const subSubCategory in subCategoryData) {
          const productList = subCategoryData[subSubCategory];
          currentList.push(...productList);
        }
      }
    }
  }
}

renderProductList(currentList); // 제품 리스트 렌더링
updateProductCount(currentList.length); // 제품 개수 업데이트
```

3-4

구현 결과 (상품 목록 페이지)

JavaScript

JavaScript



ZV-E10M2

기록의 완성

1,340,000원



ZV-E10M2

기록의 완성

1,340,000원

```
// 제품 리스트를 화면에 렌더링하는 함수
function renderProductList(productList) {
  list_container.innerHTML = ''; // 기존 내용 초기화

  productList.forEach(v => {
    const item = document.createElement("a");
    item.classList.add("item");

    const qs = queryStringById( product, v );

    item.href = `view.html?id=${v.id}&${qs}`;

    // 이미지 컨테이너 생성
    const imgContainer = document.createElement("div");
    imgContainer.classList.add("img-container");

    // 이미지 요소 생성
    const img = document.createElement("img");

    // 이미지 경로 설정
    let imgSrc;
    let folderName;

    // 폴더명 생성: 카테고리 키와 ID를 조합하여 폴더명 생성
    // 예: camera1, video2 등
    folderName = `${categoryKey}${v.id}`;

    if (v.color && v.color.length > 0) {
      // 초기 이미지는 첫 번째 색상(clr0)
      imgSrc = `assets/img/${folderName}/clr0_0.png`;
    } else {
      // 색상이 없는 경우 기본 썸네일 사용
      imgSrc = `assets/img/${folderName}/${v.thumbnail}`;
    }

    img.setAttribute("src", imgSrc);
    imgContainer.appendChild(img);
    item.appendChild(imgContainer);
  });
}
```

```
// 색상 선택 버튼들
if [v.color && v.color.length > 1] {
  const colorButtons = document.createElement("div");
  colorButtons.classList.add("color-buttons");

  v.color.forEach(color, index) => {
    const button = document.createElement("button");
    const buttonInner = document.createElement("button");

    button.classList.add("color-button");
    button.setAttribute("data-color-index", index);

    // 버튼 배경색을 실제 색상으로 설정
    button.style.backgroundColor = color;

    if(button.style.backgroundColor == "black") {
      button.appendChild(buttonInner);
      buttonInner.classList.add("color-button-inner");
    }

    // 마우스 오버 이벤트 리스너
    button.addEventListener('mouseover', () => {
      img.setAttribute("src", `assets/img/${folderName}/clr${index}_0.png`);
    });

    colorButtons.appendChild(button);
  };

  item.appendChild(colorButtons);
} else {
  img.style.marginBottom = "24px";
}

// 제품 제목
const title = document.createElement("h3");
title.innerHTML = v.title;
item.appendChild(title);

// 제품 정보
const info = document.createElement("p");
info.innerHTML = v.info;
item.appendChild(info);

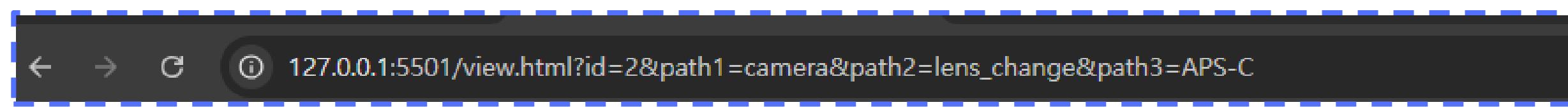
// 제품 가격
const price = document.createElement("span");
price.innerHTML = v.price.toLocaleString('ko-KR') + '원'; // 숫자 형태를 한국 스타일의 쉼표 구분 형식으로 변환
item.appendChild(price);

list_container.appendChild(item); // 리스트 컨테이너에 아이템 추가
}
```

- 색상 원형에 마우스 오버 시, 색상에 맞는 이미지 변화
- 정수 형태의 금액을 한국 스타일의 쉼표 구분 형식으로 변환

3-5

구현 결과 (상품 상세 페이지)



```
( async () => {
  const params = utilHelper.getQuery();

  const paramsArr = Object.values(params).slice(1); // id 값 제외

  const curId = params.id;

  if ( !curId ) {
    alert("제품이 없습니다");
    history.back();
    return;
  }

  let response = null;
  let products, item;

  try {
    // 전체 json 조회
    response = await axios.get('http://localhost:3001/products'); //console.log(response.data);

    // paramsArr 요소로 백엔드 json 접근
    products = paramsArr.reduce((acc, key) => {
      return acc[key]; // 각 단계에서 접근
    }, response.data);
    //console.log(products);

    // 접근한 배열에서 현재 제품의 id가 포함된 객체 가져오기
    item = products.find(item => item.id == curId);

  } catch (e) {
    console.error(e);
    alert("요청 실패");
    return;
  }
})
```

JavaScript

다른 페이지에서 query string으로 받아온 id와 상위 key값들로 백엔드에 접근하여 상품의 정보 조회

3-5

구현 결과 (상품 상세 페이지)



- 이미지 슬라이드 → Swiper 라이브러리
- 각 상품의 이미지 개수 별로 슬라이드 생성

HTML

```
<!-- body -->
<div class="main">

  <div class="product-view-wrap">

    <div class="product-view-main" > <!-- flex 좌측 우측 나누기 -->

      <!-- 메인 좌측 - 사진 슬라이드 --> <!-- flex width 58% -->

      <div class="product-main-slider">
        <swiper-container class="view_swiper" navigation="true" pagination="true" loop="true">
          <!-- ... -->
        </swiper-container>
      </div>
```

JavaScript

```
/* ----- 메인 좌측 - 슬라이더 ----- */

const swiperContainerViewSwiper = document.querySelector('.view_swiper');
// 이미지 개수 별로 슬라이드 생성

for ( let i=0; i

```

3-5

구현 결과 (상품 상세 페이지)

<https://swiperjs.com/>

Swiper 라이브러리로 이미지 슬라이드 구현



HTML

```

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100..900&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Swiper/5.4.5/css/swiper.css">
<link rel="icon" href="assets/img/favicon_144x144.png">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.6.0/css/all.min.css" integrity="sha512-Kc323vGRFa7TmouAFcNvceOavadsSiaLOTSRI29allW4II/M7nSPA/xFU70av1cwX4OnYxTxve5UMa5GT6" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-element-bundle.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/5.4.5/js/swiper.js"></script>
<script src="assets/js/ArrayHyeon.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" integrity="sha512-fjqhx/qtq/1itJ0C2ejDxltZVFg==" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/swiper@11.1.2/dist/swiper.min.js" integrity="sha512-4+int-https://cdn.jsdelivr.net/npm/swiper@11.1.2/dist/swiper.min.js" crossorigin="anonymous"></script>
<div class="product-main-slider">
  <swiper-container class="view_swiper" navigation="true" pagination="true">
    <!--
    <swiper-slide class="hidden"><div class="img-container">  </div></swiper-slide>
    <swiper-slide><div class="img-container">  </div></swiper-slide>
    <swiper-slide><div class="img-container">  </div></swiper-slide>
    <!--
  </swiper-container>

```

Swiper 속성 설정

```

/* --- 메인 좌측 - 슬라이더 --- */
const swiperContainerViewSwiper = document.querySelector('.view_swiper');
// 이미지 개수 별로 슬라이드 생성

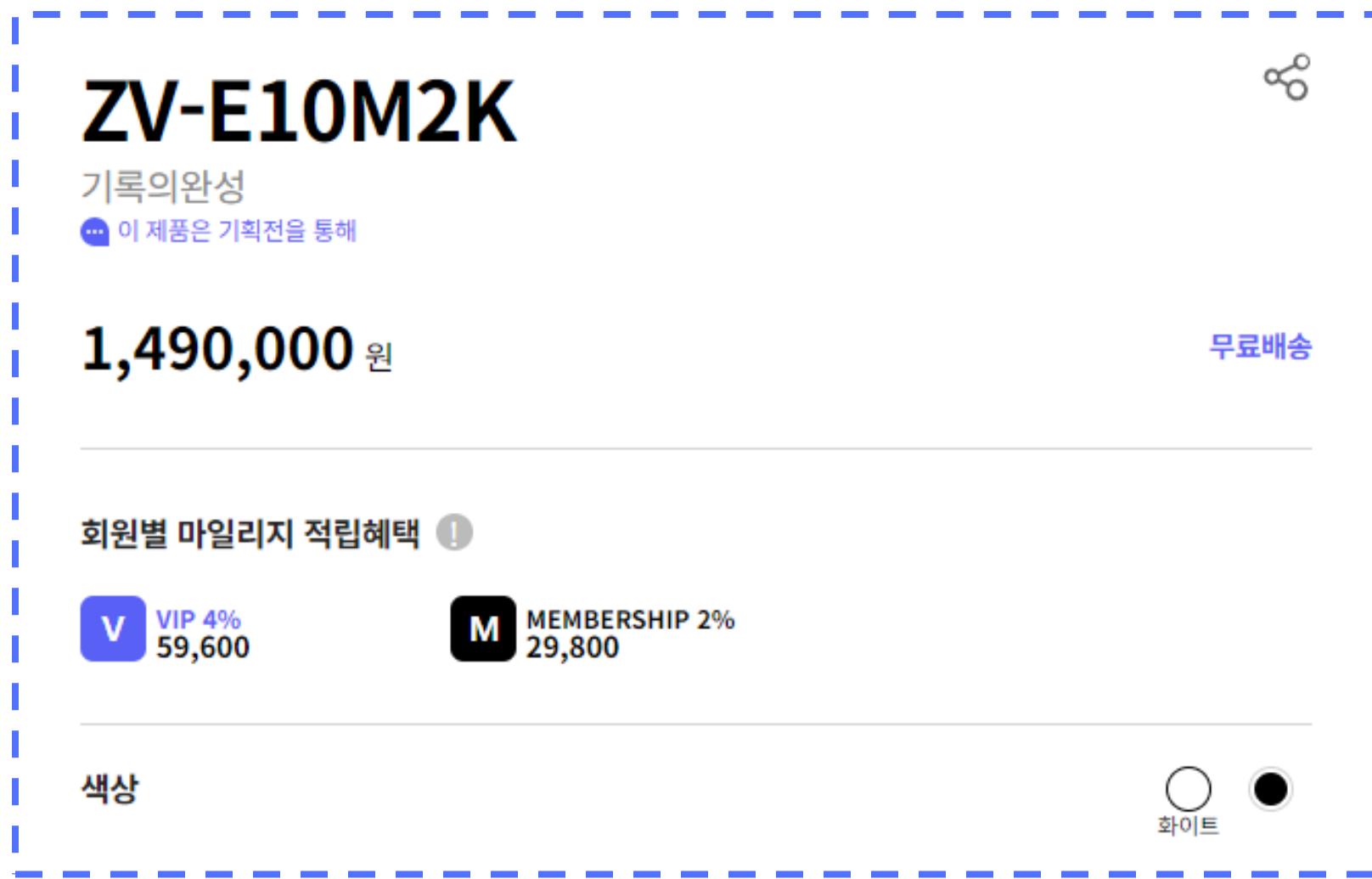
for ( let i=0; i

```

JavaScript로 HTML에 Swiper 레이아웃 추가

3-5

구현 결과 (상품 상세 페이지)



- 백엔드 연결 전, HTML, CSS로 틀 잡은 후,
- 자바스크립트로 백엔드를 통해 데이터 수신하여 상품별 정보 조회
(→ 자바스크립트 다음 페이지에)

HTML

```

<!-- 메인 우측 - 구매관련 -->
<div class="product-view-about" style="flex: 42%; position: relative; height: 100%;>
  <!-- 상품명 -->
  <div class="product-cont-top">
    <div>
      <!-- <div class="flag"> NEW </div> -->
      <div class="prd-name"> ZV-E10M3 </div>
      <div class="prd-text"> 기록의 완 </div>
      <div class="prd-desc"> <div class="icon"></div> <p>이 제품은 기획전을 통해 </p> </div>
    </div>
    <div><a class="share"></a></div>
  </div>
  <!-- 가격, 무료배송 -->
  <div class="product-cont">
    <div class="product-price">
      <div class="sale">
        <strong class="cam-price"> 134000 </strong> 원
      </div>
      <div class="delivery-txt"> 무료배송 </div>
    </div>
  </div>
  <!-- 구분선 -->
  <div class="divider-light"></div>
  <!-- 회원별 마일리지 적립혜택-->
  <div class="product-cont member">
    <div class="tit"> 회원별 마일리지 적립혜택 <span class="icon-question">!</span> </div>
    <ul class="membership-rating">
      <li class="vip">
        
        <div class="save-info">
          <span class="percentage vip"> VIP 4% </span>
          <p class="mileage vip-mlg"> 53600 </p>
        </div>
      </li>
      <li class="family">
        
        <div class="save-info">
          <span class="percentage"> MEMBERSHIP 2% </span>
          <p class="mileage family-mlg"> 53600 </p>
        </div>
      </li>
    </ul>
  </div>
  <!-- 구분선 -->
  <div class="divider-light"></div>
  <!-- 색상 -->
  <div class="product-cont-color-select">
    <!-- ... -->
  </div>

```

3-5

구현 결과 (상품 상세 페이지)

JavaScript

ZV-E10M2K

기록의 완성

이 제품은 기획전을 통해

1,490,000 원

회원별 마일리지 적립혜택

V VIP 4%
59,600M MEMBERSHIP 2%
29,800

색상

/* --- 메인 우측 - 구매관련

```
document.querySelector(".prd-name").innerHTML = item.title;
document.querySelector(".prd-text").innerHTML = item.info;

const camPrice = item.price;
document.querySelector(".cam-price").innerHTML = camPrice.toLocaleString();

document.querySelector(".vip-mlg").innerHTML = (camPrice*0.04).toLocaleString();

document.querySelector(".family-mlg").innerHTML = (camPrice*0.02).toLocaleString();
```



1,340,000 원

무료배송

색상

화이트

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

●

```
/* --- 색상 --- */
if ( colorArr.length > 1 ) {
  const div = document.createElement('div');
  div.classList.add('color-select');
  document.querySelector('.product-cont-color-select').appendChild(div);

  const p = document.createElement('p');
  p.classList.add('tit');
  p.innerHTML = "색상";

  const ul = document.createElement('ul');
  ul.classList.add('circle-color-box');

  for ( let i=0; i<2; i++ ) {
    let clr = colorArr[i];

    const li = document.createElement('li');

    const a = document.createElement('a');
    a.classList.add('color-btn');
    a.classList.add(clr);

    const span1 = document.createElement('span');
    span1.classList.add('circle-color');
    if( i==0 ) span1.classList.add('active');

    const span2 = document.createElement('span');
    span2.classList.add('c-bg');

    span1.appendChild(span2);

    const span3 = document.createElement('span');
    span3.classList.add('color-name');
    if( i==0 ) span3.classList.add('active');
    span3.innerHTML = clr_name(clr);

    a.appendChild(span1);
    a.appendChild(span3);

    li.appendChild(a);
    ul.appendChild(li);
  }
}

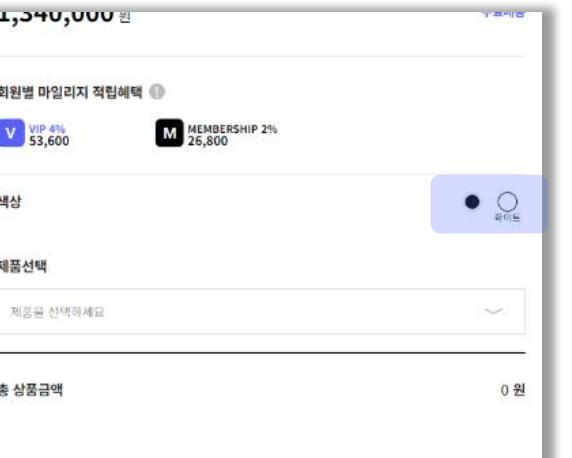
div.appendChild(p);
div.appendChild(ul);
```

```
document.querySelectorAll('.color-btn').forEach( (v,i) => {
  v.addEventListener( 'click', e => {
    e.preventDefault();
    const clickIdx = i;

    document.querySelectorAll('.circle-color').forEach( (v1,i1) =>
      if ( clickIdx == i1 ) {
        v1.classList.add('active');
      } else {
        v1.classList.remove('active');
      }
    );
    document.querySelectorAll('.color-name').forEach( (v2,i2) =>
      if ( clickIdx == i2 ) {
        v2.classList.add('active');
      } else {
        v2.classList.remove('active');
      }
    );
  });
}

document.querySelectorAll('.main-img').forEach( (v3,i3) => {
  v3.setAttribute( "src", `assets/img/${paramsArr[0]}${[curId]}/${imgArr[i][i3]}` );
})
```

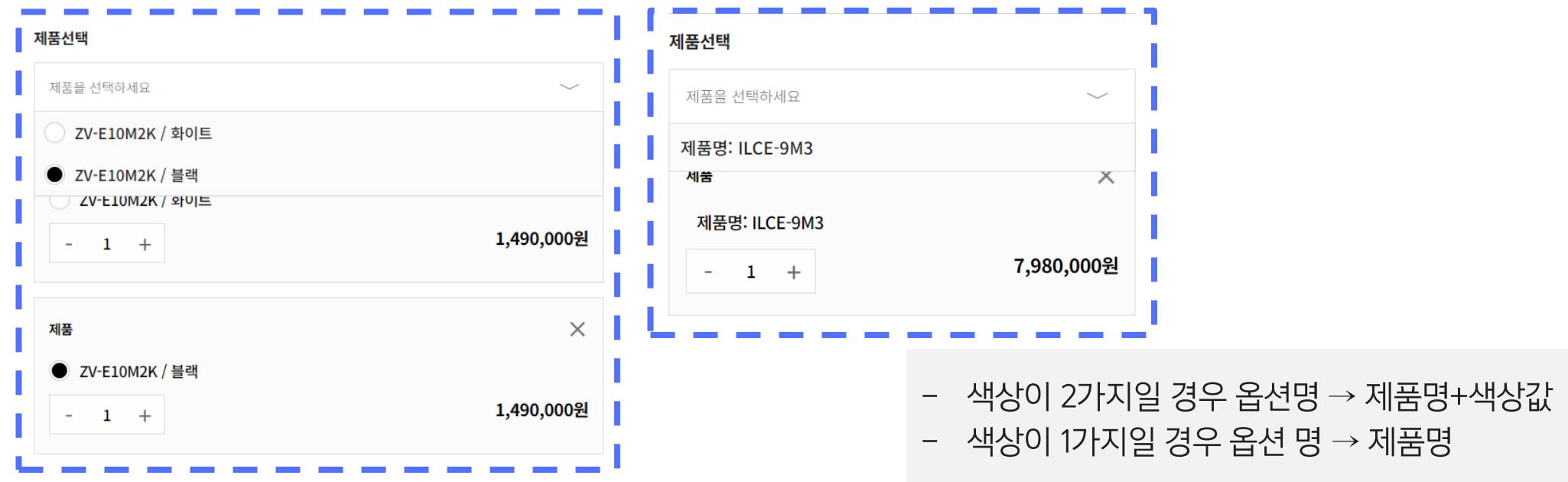
- 색상 원형 버튼 클릭 이벤트
- active 클래스 추가/제거
- 백엔드에 이미지 파일이름을 배열로 저장하여, 색상 인덱스 값에 따라 이미지 색상 변경



- 백엔드에 색상값을 배열로 저장
- 색상이 두 가지일 경우 색상 영역, 원형버튼 생성
- active 클래스 유무로 슬라이드의 이미지와 원형버튼 강조표시 변화
- 기본색상은 색상배열0번째 값

3-5

구현 결과 (상품 상세 페이지)



HTML

```
<!-- 제품선택 -->
<div class="product-cont prd-select-wrap">
  <div class="prd-select-inner">
    <div class="prd-select-box">
      <p class="tit">제품선택</p>
      <!-- 제품을 선택하세요-->
      <div>
        <div class="select-ui-zone">
          <a class="selected-btn">
            <span>제품을 선택하세요</span>
            <span class="arrow-down"></span>
          </a>
          <!-- 선택박스 submenu -->
          <ul class="select-inner">
            <!-- ...>
          </ul>
        </div>
      </div>
      <!-- 선택된 제품 박스 -->
      <!-- ...>
    </div>
  </div>
</div>
```

색상 원형을 재사용 하기 위한 메서드

```
/* -- -- -- -- -- circleColor 메서드 -- -- -- -- -- */
const circleColor = selector => {
  if( !selector ) return;

  selector.classList.add('color-btn');

  const span1 = document.createElement('span');
  span1.classList.add('circle-color');

  const span2 = document.createElement('span');
  span2.classList.add('c-bg');

  span1.appendChild(span2);
  selector.appendChild(span1);
};
```

JavaScript

```
/* 제품을 선택하세요 */
const ulSelectInner = document.querySelector('.select-inner');

const colorNum = colorArr.length;
for ( let i=0; i<colorNum; i++ ) {
  const li = document.createElement('li');
  li.classList.add('inner-list');

  const a = document.createElement('a');
  li.appendChild(a);
  ulSelectInner.appendChild(li);
}

document.querySelectorAll('.inner-list').forEach( (v,i) => {
  let span1, span2;
  if ( colorNum > 1 ) {
    span1 = document.createElement('span');
    span1.classList.add('circle-prd-clr');
    span1.classList.add('pad');

    circleColor(span1);

    const colorName = colorArr[i];
    span1.classList.add( colorName );
  }
  else {
    span2 = document.createElement('span');
    span2.classList.add('submenu-txt');
    span2.classList.add('pad');
    span2.innerHTML = `제품명: ${item.title}`;

    const a = v.children[0];
    if ( span1 ) a.appendChild(span1);
    a.appendChild(span2);
  }
});

document.querySelector('.selected-btn').addEventListener( 'click', e => {
  e.preventDefault();
  ulSelectInner.classList.toggle('active');
} );
```

“제품을 선택하세요” 클릭이벤트로 active 클래스 토글

3-5

구현 결과 (상품 상세 페이지)

```
/* --- 제품 선택 - 선택된 제품 박스 --- */
const innerList = document.querySelectorAll('.inner-list'); // submenu list 선택할 제품 목록

innerList.forEach( (v,i) => {
  v.addEventListener( 'click', e => {
    e.preventDefault();
    const current = e.currentTarget; // 클릭한 선택한 제품 ex) "I

    const createBox = () => {
      const div1 = document.createElement('div');
      div1.classList.add('selected-opt');

      const div2 = document.createElement('div');
      div2.classList.add('selected-opt-container');

      const div3 = document.createElement('div');
      div3.classList.add('selected-prd-top');

      const p = document.createElement('p');
      p.classList.add('prd');
      p.innerHTML = '제품';

      const a = document.createElement('a');
      a.classList.add('prd-delete');
      a.innerHTML = '구매목록에서 삭제';

      div3.appendChild(p);
      div3.appendChild(a);

      const div4 = document.createElement('div');
      div4.classList.add('selected-name');

      if ( innerList.length > 1 ) {
        const circlePrdClr = current.querySelector('.circle-prd-clr');
        div4.appendChild( circlePrdClr.cloneNode(true) );
      }

      const submenuTxt = current.querySelector('.submenu-txt');
      div4.appendChild( submenuTxt.cloneNode(true) );

      const div5 = document.createElement('div');
      div5.classList.add('selected-bottom');

      const div6 = document.createElement('div');
      div6.classList.add('count-container');
    
```

- 상품 선택 옵션 클릭 이벤트
- 선택한 제품의 정보 박스 생성
- 선택한 옵션내용을 박스에 복제

```
      const btn1 = document.createElement('button');
      btn1.classList.add('minus');
      btn1.classList.add('cnt-box');
      btn1.innerHTML = '-';

      const input = document.createElement('input');
      input.classList.add('cnt-box');
      input.setAttribute( 'type', 'text' );
      input.setAttribute( 'value', 1 );

      const btn2 = document.createElement('button');
      btn2.classList.add('plus');
      btn2.classList.add('cnt-box');
      btn2.innerHTML = '+';

      div6.appendChild(btn1);
      div6.appendChild(input);
      div6.appendChild(btn2);

      const div7 = document.createElement('div');
      div7.classList.add('selected-price');
      div7.innerHTML = `${(item.price).toLocaleString()}원`;

      div5.appendChild(div6);
      div5.appendChild(div7);

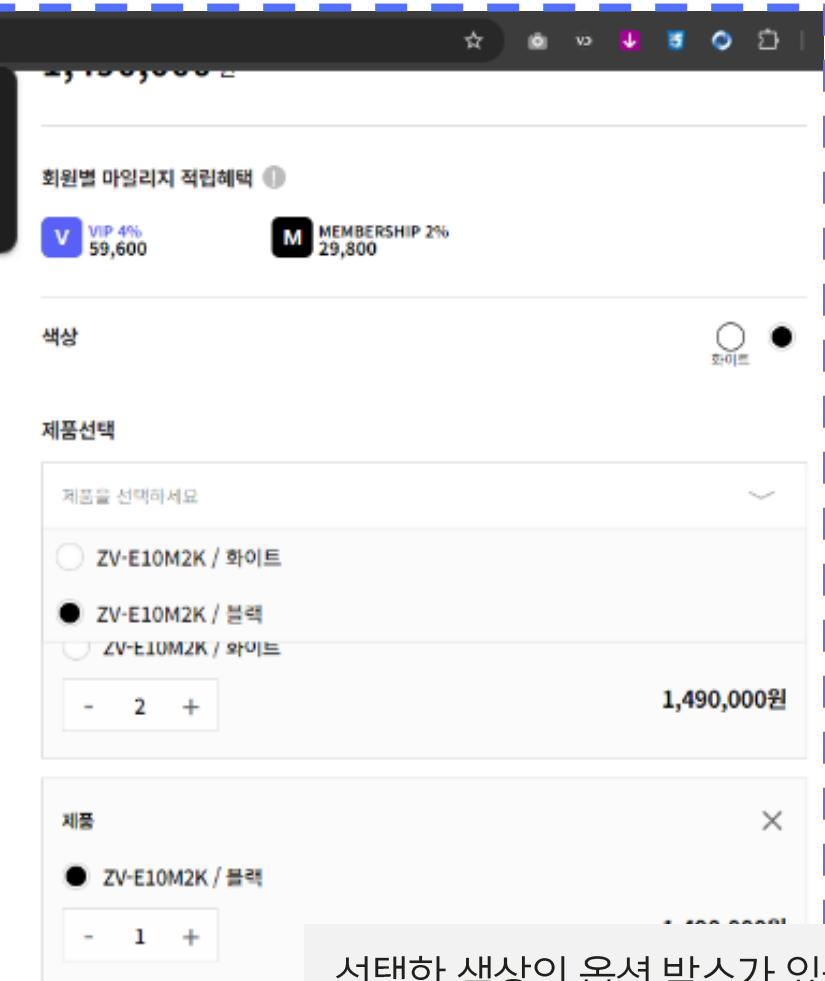
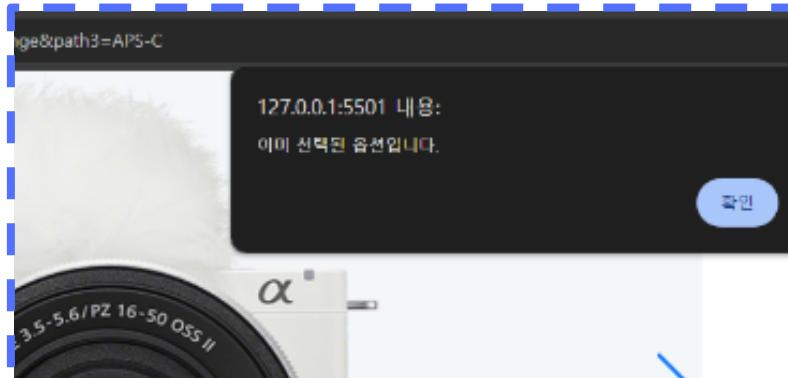
      div2.appendChild(div3);
      div2.appendChild(div4);
      div2.appendChild(div5);

      div1.appendChild(div2);
      //console.log(div1);

      document.querySelector('.prd-select-box').appendChild(div1);

      a.addEventListener('click', (e) => {
        e.preventDefault();
        div1.remove(); // 해당 선택된 옵션 삭제
      });
    
```

X 버튼 클릭 시, 선택된 옵션박스 삭제



```
// 제품 선택 박스 중복 방지
const existingOpt = Array.from(document.querySelectorAll('.selected-opt')).some(opt => {
  if ( innerList.length>1 ) {
    return opt.querySelector(`.${colorArr[i]}`);
  } else {
    return true;
  }
});

if (existingOpt) {
  alert('이미 선택된 옵션입니다.');
} else {
  createBox();
}

ulSelectInner.classList.remove('active');
});
```

JavaScript

3-5

구현 결과 (상품 상세 페이지)

제품

ZV-E10M2K / 화이트

- 2 +

1,490,000원

제품

ZV-E10M2K / 블랙

- 1 +

1,490,000원

총 상품금액

4,470,000 원

바로 구매하기

```
<!-- 총 상품금액 -->
<div class="result-list">
  <div class="result-chk-box">
    <p class="tit">
      총 상품금액
      <span class="s-txt">
        <span class="count"></span>
      </span>
    </p>
    <p class="result-price">
      <span class="num"> </span>
      <span class="tit">원</span>
    </p>
  </div>
</div>
```

```
<!-- 찜, 장바구니, 선물하기, 구매하기 -->
<div class="result-btn-inner">
  <div class="result-btn-box">
    <ul>
      <li class="heart btn-icon-container"> <!-- <a href="" class="btn-icon"></a> --> </li>
      <li class="cart btn-icon-container"></li>
      <li class="gift btn-icon-container"></li>
      <li class="final">
        <a> 바로 구매하기 </a>
      </li>
    </ul>
  </div>
</div>
```

- 비동기 처리 이후 클릭 이벤트 등 DOM의 변화를 감지하기 위해 MutationObserver 사용
- 변화가 감지되면 calculateTotal 함수 호출
- "+"/-" 버튼 클릭 시, 수량을 증가/감소 → calculateTotal 함수 호출 → 총 금액 재계산

클래스 이름 (heart, cart, gift)에 따라 css의 background url 변경

```
/* --- 찜, 장바구니, 선물하기, 구매하기 --- */
document.querySelectorAll('.btn-icon-container').forEach( (v,i) => {
  const a = document.createElement('a');
  a.classList.add('btn-icon');
  v.appendChild(a);

  const iconUrl = `../assets/img/ico_${v.classList[0]}.svg`;
  v.style.background = `url(${iconUrl}) no-repeat center`;
} );
```

JavaScript

```
/* --- 총 상품금액 계산 --- */
const totalPrice = document.querySelector('.result-price .num');

function calculateTotal() {
  const selectedOpts = document.querySelectorAll('.selected-opt');
  let total = 0;

  selectedOpts.forEach(opt => {
    const priceText = opt.querySelector('.selected-price').textContent;
    const countInput = opt.querySelector('.cnt-box[type="text"]');

    if (priceText & countInput) {
      const price = parseInt(priceText.replace(/\d/g, ''), 10);
      const count = parseInt(countInput.value, 10);
      total += price * count;
    }
  });

  totalPrice.textContent = total.toLocaleString();
}

// MutationObserver 설정
const observer = new MutationObserver(() => {
  calculateTotal();
});

// 관찰할 설정
const config = {
  childList: true,
  subtree: true
};

// prdSelectBox 요소에 대해 관찰 시작
const prdSelectBox = document.querySelector('.prd-select-box');
if (prdSelectBox) {
  observer.observe(prdSelectBox, config);
} else {
  console.error('prd-select-box 요소가 존재하지 않습니다.');
}

// 버튼 클릭 이벤트 설정
document.addEventListener('click', (e) => {
  if (e.target.classList.contains('plus') || e.target.classList.contains('minus')) {
    const inputField = e.target.parentElement.querySelector('.cnt-box[type="text"]');
    if (inputField) {
      let count = parseInt(inputField.value, 10);

      if (e.target.classList.contains('plus')) {
        count++;
      } else if (e.target.classList.contains('minus')) {
        count = count > 1 ? count - 1 : 1; // 최소값 1로 설정
      }

      inputField.value = count;
      calculateTotal(); // 총 금액 재계산
    }
  }
});

// 초기 총 금액 계산
calculateTotal();
```

HTML

3-5

구현 결과 (상품 상세 페이지)

JavaScript

현재 상세조회 하고 있는 상품을 제외한,
같은 카테고리에 있는 제품들을 랜덤으로 표시

함께 구매하시면 좋은 추천 제품

			
ILCE-9M3	ZV-1F	ILCE-6700	ILCE-7CM2
The Power of One Frame	나를 담는 브이로그 카메라 ZV-1F	MINIMAL WONDER	원핸드 컴팩트 플프레임
7,980,000원	759,000원	1,890,000원	2,690,000원

HTML

```

<!-- 함께 구매하시면 좋은 추천 제품 -->


<div>
    <div class="rcmd-tit">함께 구매하시면 좋은 추천 제품 </div>
    <!-- <div class="swiper-rcmd rcmd-prd-list">-->
    <!-- <ul class="swiper-wrapper-rcmd"> --> <!-- navigation="true" -->
    <swiper-container class="recommendswiper" space-between="30" slides-per-view="4">
    <!-- ... </swiper-container>
  </div>
</div>


```

```

/* -- 함께 구매하시면 좋은 추천 제품 -- */

/**
 * 현재 제품과 같은 category(camera, video, lens, ...) 인 객체들을
 * @param {object} data 탐색할 object
 * @param {string} category category 이름
 * @returns array
 */
function getProducts(data, category) {
  const result = [];

  // 카테고리에서 데이터 찾기
  const categoryData = data[category];
  if (!categoryData) return result;

  // 재귀적으로 탐색하는 내부 함수
  function searchProducts(obj) {
    if (Array.isArray(obj)) {
      result.push(...obj);
    } else if (typeof obj === 'object' && obj !== null) {
      for (const value of Object.values(obj)) {
        searchProducts(value);
      }
    }
  }

  // 카테고리 구조 탐색 시작
  searchProducts(categoryData);
  return result;
}

const productList = getProducts( response.data, paramsArr[0] );

```

- getProducts 메서드
- 현재 상품과 같은 category인 객체들을 배열로 반환
- productList에 전체 제품에서 현재 상품과 같은 category인 객체들의 배열 저장

- ArrayHyeon 클래스에 shuffleArray, removeElementAtIndex 메서드 생성
- 클래스 참조

ArrayHyeon.js

```

> js > ArrayHyeon.js > ...
```
 * ArrayHyeon.js
 *
 */

class ArrayHyeon {

 /**
 * 배열의 순서를 랜덤하게 바꾼다
 * @param {Array} array - 랜덤하게 순서를 바꿀 배열
 * @returns {Array} - 순서가 랜덤하게 바뀐 배열
 */
 shuffleArray(array) {
 // 배열을 복사하여 원본 배열을 변경하지 않도록 한다
 let shuffledArray = array.slice();

 for (let i = shuffledArray.length - 1; i > 0; i--) {
 // 0부터 i까지의 인덱스 중에서 무작위로 선택한다
 const j = Math.floor(Math.random() * (i + 1));

 // 현재 인덱스 i와 무작위로 선택된 인덱스 j의 요소를 교환한다
 [shuffledArray[i], shuffledArray[j]] = [shuffledArray[j], shuffledArray[i]];
 }

 return shuffledArray;
 }

 /**
 * 배열에서 특정 인덱스의 요소를 삭제한다
 * @param {Array} array - 요소를 삭제할 배열
 * @param {number} index - 삭제할 요소의 인덱스
 * @returns {Array} - 요소가 삭제된 배열
 */
 removeElementAtIndex(array, index) {
 // 유효한 인덱스인지 확인
 if (index < 0 || index >= array.length) {
 throw new Error('Index out of bounds');
 }

 // 배열에서 특정 인덱스의 요소를 삭제한다
 array.splice(index, 1);

 return array;
 }
}

const arrayHyeon = new ArrayHyeon();

```

3-5

# 구현 결과 ( 상품 상세 페이지 )

## 함께 구매하시면 좋은 추천 제품

|                                                                                    |                                                                                     |                                                                                      |                                                                                      |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|  |  |  |  |
| ILCE-9M3                                                                           | ZV-1F                                                                               | ILCE-6700                                                                            | ILCE-7CM2                                                                            |
| The Power of One Frame                                                             | 나를 담는 브이로그 카메라 ZV-1F                                                                | MINIMAL WONDER                                                                       | 원핸드 컴팩트 줌프레임                                                                         |
| 7,980,000원                                                                         | 759,000원                                                                            | 1,890,000원                                                                           | 2,690,000원                                                                           |

JavaScript

- 현재 상품과 같은 category인 객체들 중 현재 상품은 배열에서 제외
- 배열의 인덱스 랜덤으로 변경
- 배열의 크기만큼 슬라이드 생성, 각각 수신 받은 데이터 입력
- 각 상품들마다 상품의 상세페이지로 이동

```

// 현재 제품 제외한 제품 배열 dataArr
let dataArr;
productList.some((v,i) => {
 if (v.id == curId) {
 dataArr = arrayHyeon.removeElementAtIndex(productList,i);
 return true;
 }
});
// some 메서드 탐색을 중단하는 기능 제공
// 콜백함수가 true 리턴하는 순간 전체 반복 중단
//console.log(dataArr);

const randomData = arrayHyeon.shuffleArray(dataArr);
//console.log(randomData);

// 슬라이드 생성

const swiperContainer = document.querySelector('.recommendSwiper');

randomData.forEach((v,i) => {
 const swiperSlide = document.createElement('swiper-slide');

 const div = document.createElement('div');
 div.classList.add('recommend-img-container');

 const a = document.createElement('a');
 const qs = queryStringById(response.data, v); //console.log(qs);
 a.setAttribute('href','view.html?id=${v.id}&${qs}');

 const img = document.createElement('img');
 img.classList.add('recommend-img');
 img.setAttribute('src','assets/img/${paramsArr[0]}${v.id}/${imgArr[0][0]}');

 a.appendChild(img);

 const span1 = document.createElement('span');
 span1.classList.add('recommend-title');
 span1.innerHTML = v.title;

 const span2 = document.createElement('span');
 span2.classList.add('recommend-desc');
 span2.innerHTML = v.info;

 const span3 = document.createElement('span');
 span3.classList.add('recommend-price');
 span3.innerHTML = v.price.toLocaleString() + '원';

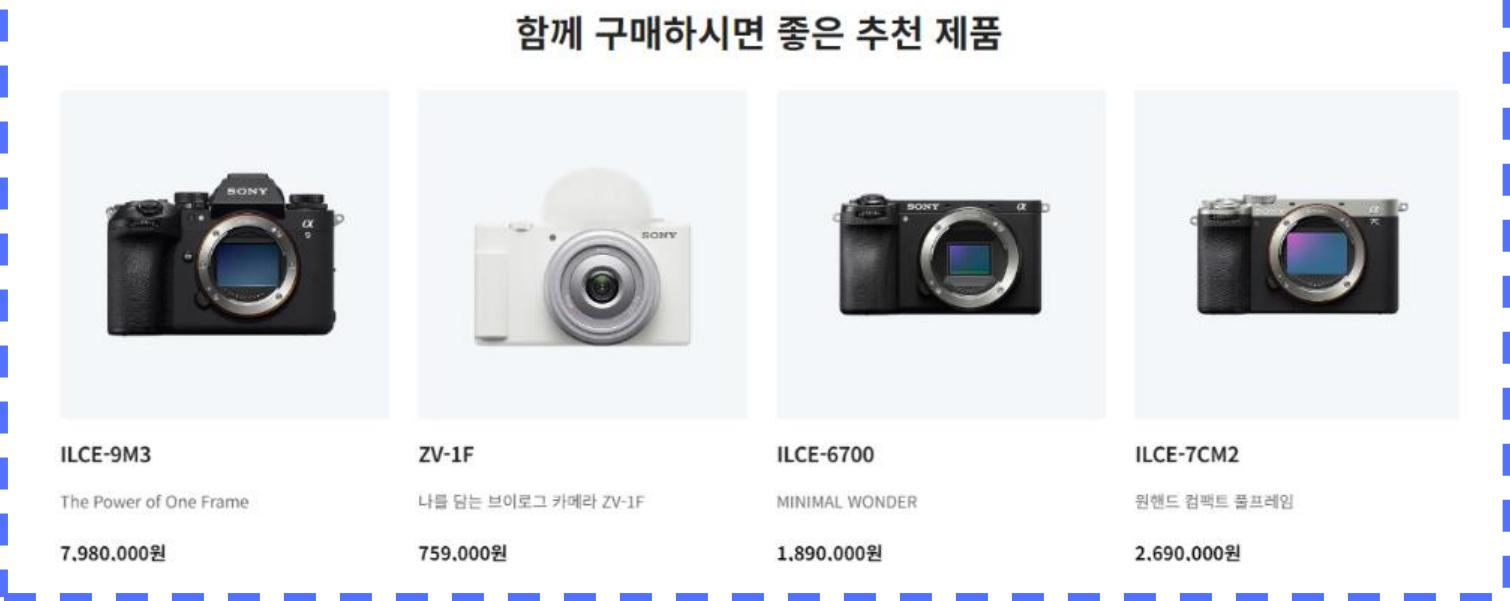
 div.appendChild(a);
 div.appendChild(span1);
 div.appendChild(span2);
 div.appendChild(span3);
 swiperSlide.appendChild(div);
 swiperContainer.appendChild(swiperSlide);
});
})();

```

3-5

# 구현 결과 ( 상품 상세 페이지 )

Swiper 라이브러리로 이미지 슬라이드 구현  
( 가로 스와이프로 가능 )



```
<!-- 함께 구매하시면 좋은 추천 제품 -->

<div class="product-content-first-recommend">
 <div>
 <div class="rcmd-tit">함께 구매하시면 좋은 추천 제품</div>
 <!-- <div class="swiper-rcmd rcmd-prd-list">-->
 <!-- <ul class="swiper-wrapper-rcmd"> --> <!-- navigation="true" -->
 <swiper-container class="recommendSwiper" space-between="30" slides-per-view="4">
 <!-- ... -->
 </swiper-container>
 </div>
</div>
```

```
<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-element-bundle.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/5.4.5/js/swiper.js"></script>
```

JavaScript

HTML

Swiper 속성 설정

CDN 사용

JavaScript로 HTML에 Swiper 레이아웃 추가

```
// 슬라이드 생성

const swiperContainer = document.querySelector('.recommendSwiper');

randomData.forEach((v,i) => {
 const swiperSlide = document.createElement('swiper-slide');

 const div = document.createElement('div');
 div.classList.add('recommend-img-container');

 const a = document.createElement('a');
 const qs = queryStringById(response.data, v); //console.log(qs);
 a.setAttribute('href', `view.html?id=${v.id}&${qs}`);

 const img = document.createElement('img');
 img.classList.add('recommend-img');
 img.setAttribute('src', `assets/img/${paramsArr[0]}${v.id}/${imgArr[0][0]}`);

 a.appendChild(img);

 const span1 = document.createElement('span');
 span1.classList.add('recommend-title');
 span1.innerHTML = v.title;

 const span2 = document.createElement('span');
 span2.classList.add('recommend-desc');
 span2.innerHTML = v.info;

 const span3 = document.createElement('span');
 span3.classList.add('recommend-price');
 span3.innerHTML = v.price.toLocaleString() + '원';

 div.appendChild(a);
 div.appendChild(span1);
 div.appendChild(span2);
 div.appendChild(span3);
 swiperSlide.appendChild(div);
 swiperContainer.appendChild(swiperSlide);
});});
```

- 현재 상품과 같은 카테고리의 상품들 개수대로 슬라이드 생성
- 백엔드에서 수신받은 상품별 데이터 입력

3-5

# 구현 결과 ( 상품 상세 페이지 )



탭메뉴 호버 시, 각메뉴 시각화  
클릭 시, 각 메뉴 콘텐츠 활성화

HTML

```
<!-- content -->
<div class="product-content">
 <!--제품개요/상세/배송,환불규정 TAB-->

 <div class="detail-tab">
 <div class="tab-menu outline active">
 <div class="text">제품 개요</div>
 <div class="bottom-line"></div>
 </div>
 <div class="tab-menu detail">
 <div class="text">제품 상세</div>
 <div class="bottom-line"></div>
 </div>
 <div class="tab-menu refund">
 <div class="text">배송/환불규정</div>
 <div class="bottom-line"></div>
 </div>
 </div>
</div>
```

JavaScript

```
/* -- -- -- -- 제품개요/상세/배송,환불규정 TAB -- -- -- */

document.querySelectorAll('.tab-menu').forEach((v,i) => {
 v.addEventListener('click', e => {
 const current = e.currentTarget;
 current.classList.add('active');

 document.querySelectorAll('.tab-menu').forEach((w,j) => {
 if (w !== current) {
 w.classList.remove('active');
 }
 });
 });
}

document.querySelectorAll('.tab-zone').forEach((v1,i1) => {
 if (i == i1) {
 v1.classList.add('active');
 } else {
 v1.classList.remove('active');
 }
});
```

PART 4

소요 시간

및 후기

# 소요 시간 및 후기

소요 시간 : 약 60시간

후기 :

## 재한

한 페이지지만 구현하는 게 아니라 JSON 데이터를 불러와 여러 페이지들을 구현해야 하는 게 생각보다 많이 힘들었습니다.

코드의 재사용과 유지 보수성이 좋게 구현하려 하니 많이 헷갈리고 코드 전체를 수정해야 하는 문제도 발생했습니다.

팀 작업이 처음이다 보니 구현해 놓은 걸 연동할 때 다른 팀원의 코드들을 분석하는 것도 어려웠고 코드를 수정하고 데이터를 일치 시키는 작업도 많은 의사소통이 필요하다는 걸 깨달았습니다.

파이널 프로젝트에는 좀 더 신중하게 의논하고 설계하여 좋은 결과물을 만들어 냈으면 좋겠습니다.

소요시간 : 약 60시간

후기 :

## 진수

학원에 다니기 전에는 javascript를 할 때 간단한 작업조차도 검색해서 하거나 해석은 할 수 있으나 직접 작성하라고 하면 막막했던 부분이나 의미를 모른채 되니까 복사해서 사용하는 것들이 대부분이었는데 수업시간에 기초부터 탄탄히 배우고 연습문제 등을 풀고나서 자신감이 조금은 생겼었는데 이번 과제를 통해서 어느정도 늘었는지 확인해 볼 수 있을 것 같아 기대를 가지고 시작했는데 시간은 걸렸지만 직접 코드를 작성할 수 있었고 라이브러리를 사용할 때에도 기본 기능 외에 추가기능을 적용시키는 코드도 해석하거나 수정하여 적용하는데에 문제없이 잘되어서 뿌듯했습니다. 팀으로 작업하면서 git을 사용하지 않아 불편했지만 다음 프로젝트 때는 배워서 사용하면 지금보다 더 발전할 수 있을 것 같습니다.

소요 시간 : 약 65시간

후기 :

## 승현

비동기 처리 부분에서 생각지 못한 이슈들이 많이 발생했고, 그 부분들을 해결하면서 많이 배운 것 같습니다.

“이렇게 하는 게 맞나?” 싶은 부분들이 많아, 끊임없는 의문의 연속으로 답답한 부분들이 많았습니다.

여러 페이지를 연동해야 하면서 데이터의 구조의 중요성을 알게 되었고, 파일 프로젝트 전에 많은 수정이 필요할 것으로 예상됩니다.

팀으로 처음 작업 하다 보니, 팀작업이 미숙했고, 계획과 의사소통, 데이터 관리 및 공유의 중요성을 알게 되었습니다.

직접 메서드를 만들어 보면서 코드를 재사용 하는 것이 재미있었습니다.

팀과 같이 공용으로 써야 하는 부분에서 자신에 대해서 확신이 없다 보니 미리 확정 짓고 넘어가야 했던 것들을 그렇게 하지 못해 작업이 효율적으로 되지 못했던 것 같아 아쉬웠습니다.

감사합니다

# API 명세서

## 제품 검색하기

### 기본 정보

메서드	URL
GET	http://localhost:3001/products

### 요청

#### 쿼리 파라미터

이름	타입	설명	필수
id	Integer	상품의 인덱스	O
path1	String	카테고리 이름 ( camera, video, lens, audio, ... )	X
path2	String	타입 이름1 ( lens_change, compact, cinema, ... )	X
path3	String	타입 이름2 ( full_frame, APS-C, 4k_handy, ... )	X

### 응답

#### products

이름	타입	설명
title	String	상품 이름
info	String	상품 설명
price	Integer	상품 가격
date	Integer	출시 연월 [YYYYMM]
color	Array	상품 색상 [color1] or [color1,color2]
img	Array	상품 이미지 예) [ ["clr0_0.png", "clr0_1.png", "clr0_2.png"], [ "clr1_0.png", "clr1_1.png", "clr1_2.png"] ]

### 기본 정보

메서드	URL
GET	http://localhost:3001/productList

### 요청

#### 쿼리 파라미터

이름	타입	설명	필수
category	String	카테고리 이름 ( camera, video, lens, audio, ... )	X

### 응답

#### productList

이름	타입	설명
prodName	String	상품목록 메뉴 이름
prodImg	String	상품목록에 메뉴에 들어가는 이미지 이름 및 확장자