

abcd

# 클론 코딩 과제

## HTML, CSS, JavaScript

---

박재한 박진수 이승현

---

# 목차

CONTENT

1	레이아웃 분석	2	DOM 구조 설정
3	구현 결과	4	소요 시간 및 후기
5	사업운영계획	6	마케팅 전략

PART 1

# 레이아웃 분석

# 1 레이아웃 분석 (전체)

header

- 메뉴바

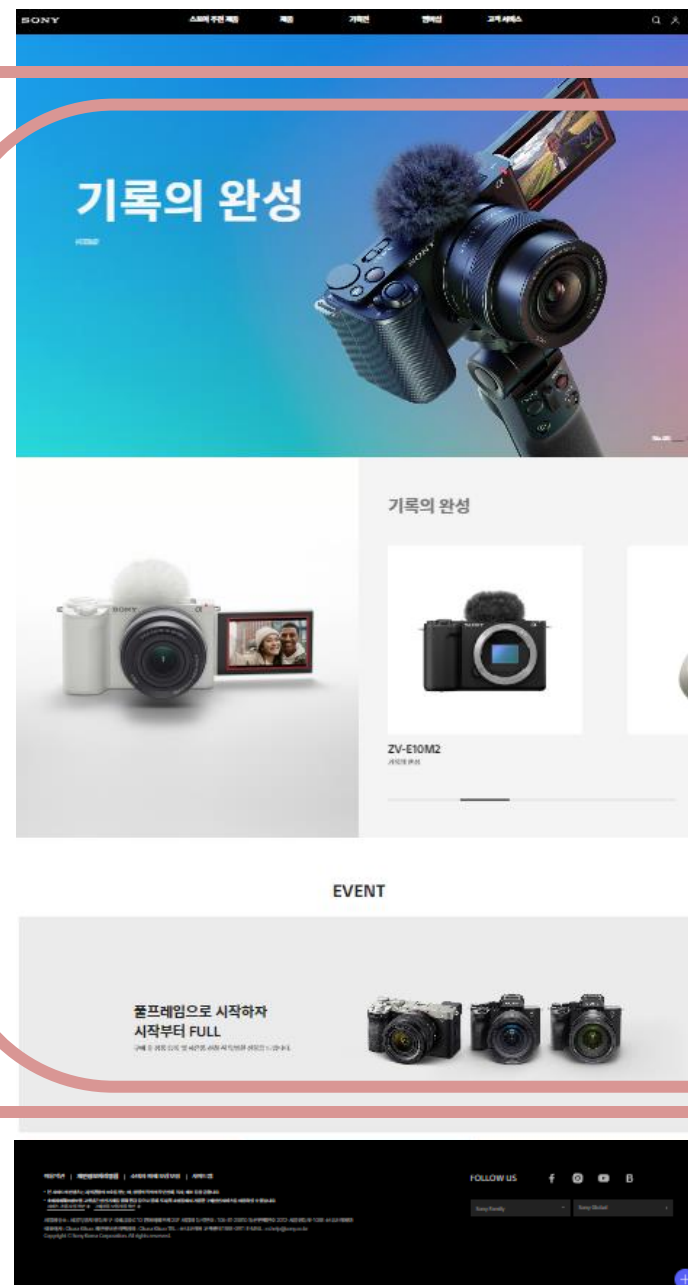
main

- 주요 콘텐츠

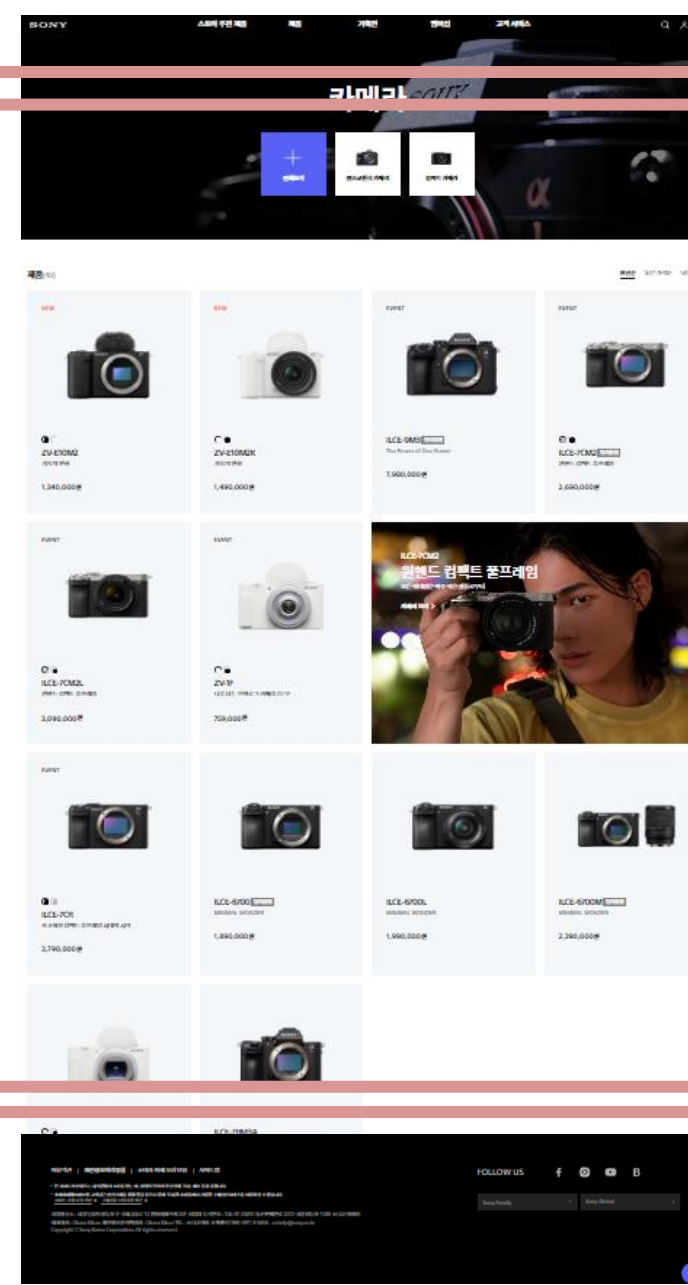
footer

- 기타 링크, 카피라이트

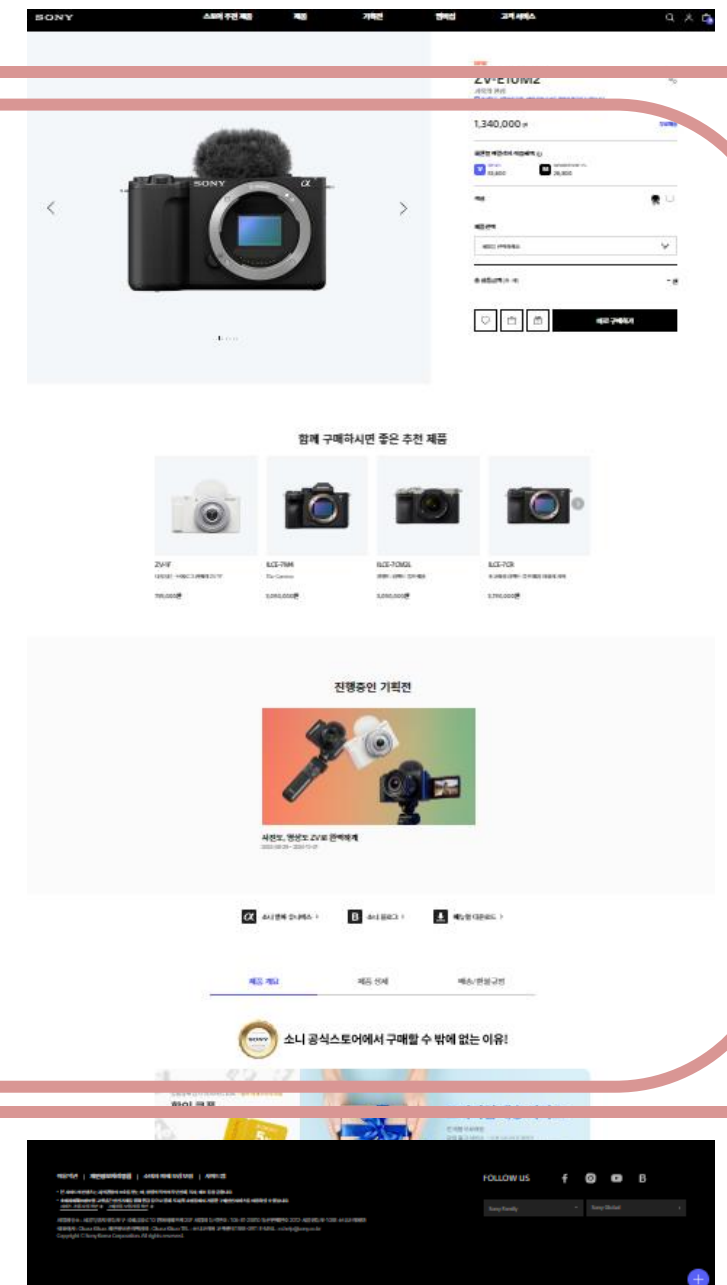
메인 페이지



상품 목록 페이지



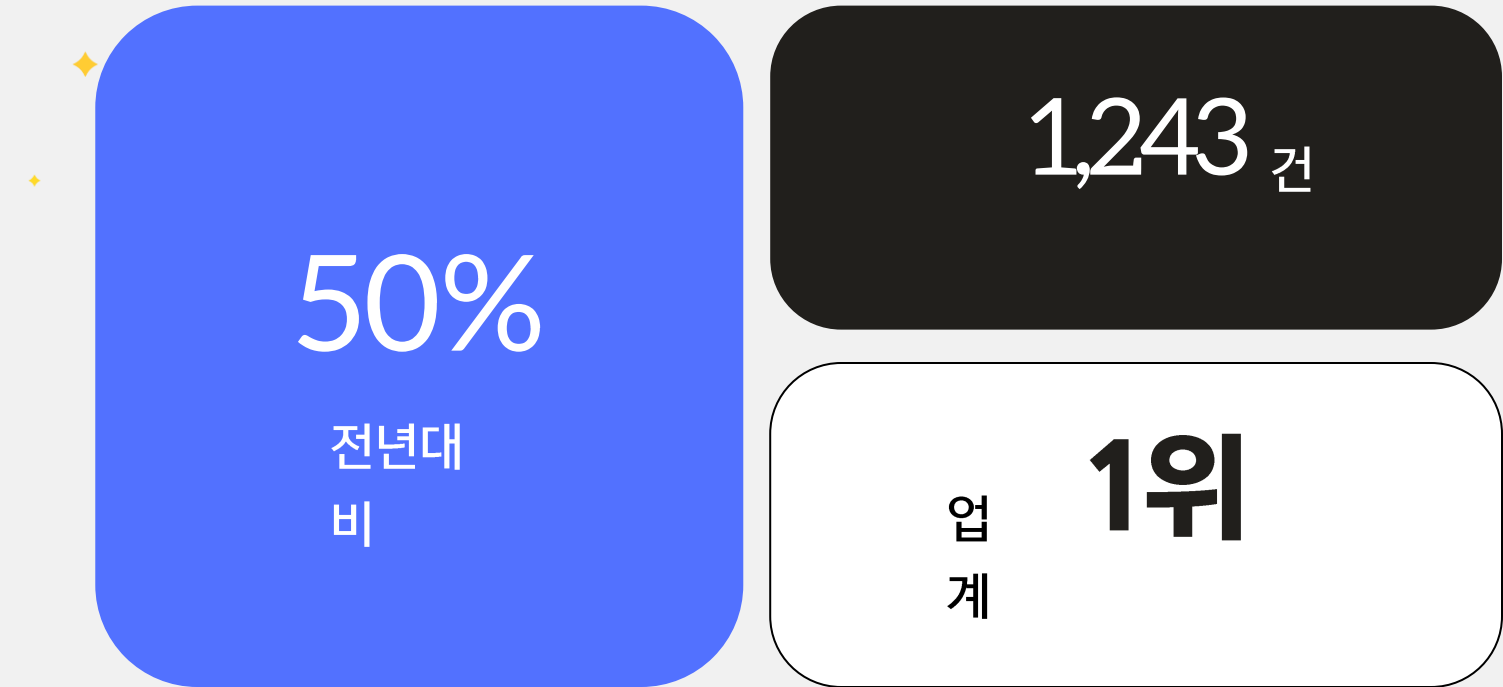
상품 상세 페이지



# 1 레이아웃 분석(Header)

## 회사소개 프레젠테이션

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.

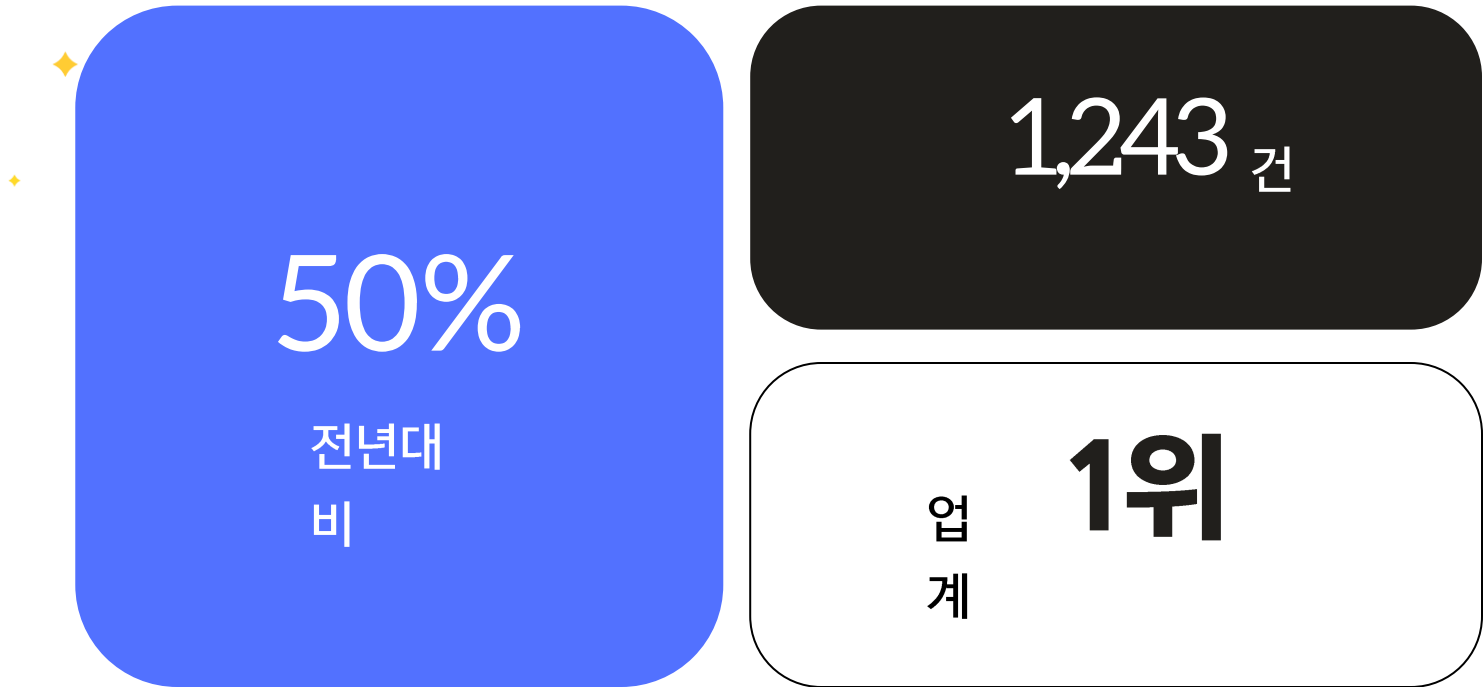


이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.

# 1 레이아웃 분석(Footer)

## 회사소개 프레젠테이션

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



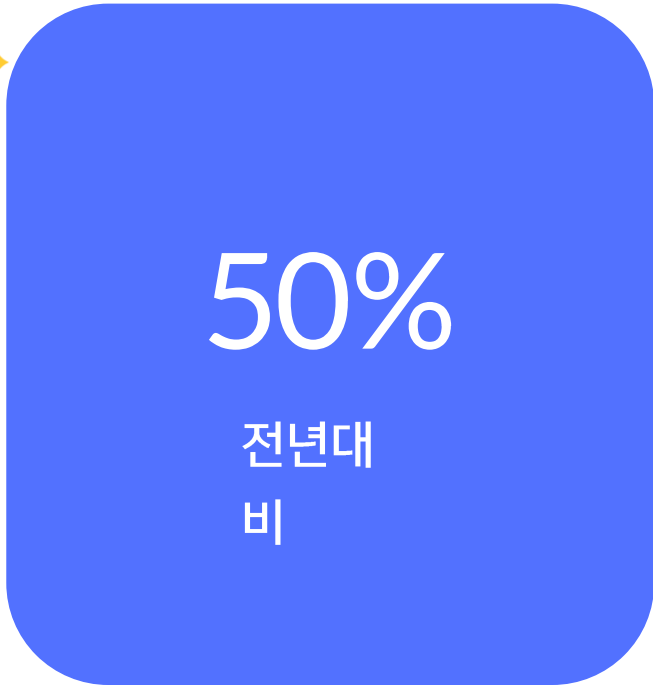
이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.

1

# 레이아웃 분석(Main-메인 페이지)

## 회사소개 프레젠테이션

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



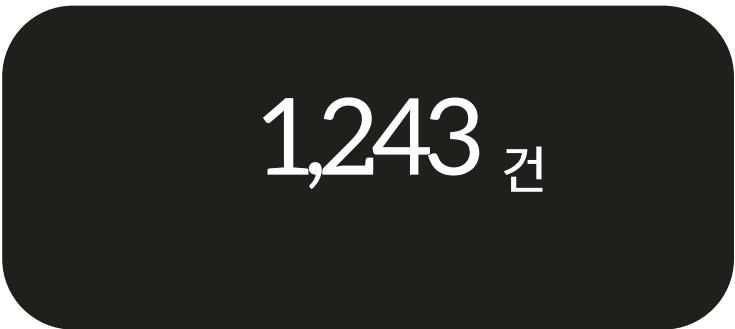
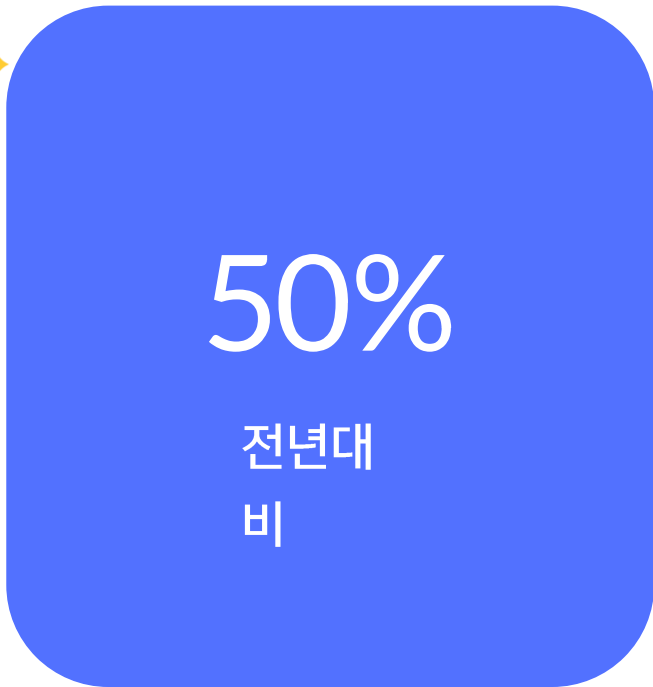
이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.

1

# 레이아웃 분석(Main-상품 목록 페이지)

## 회사소개 프레젠테이션

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



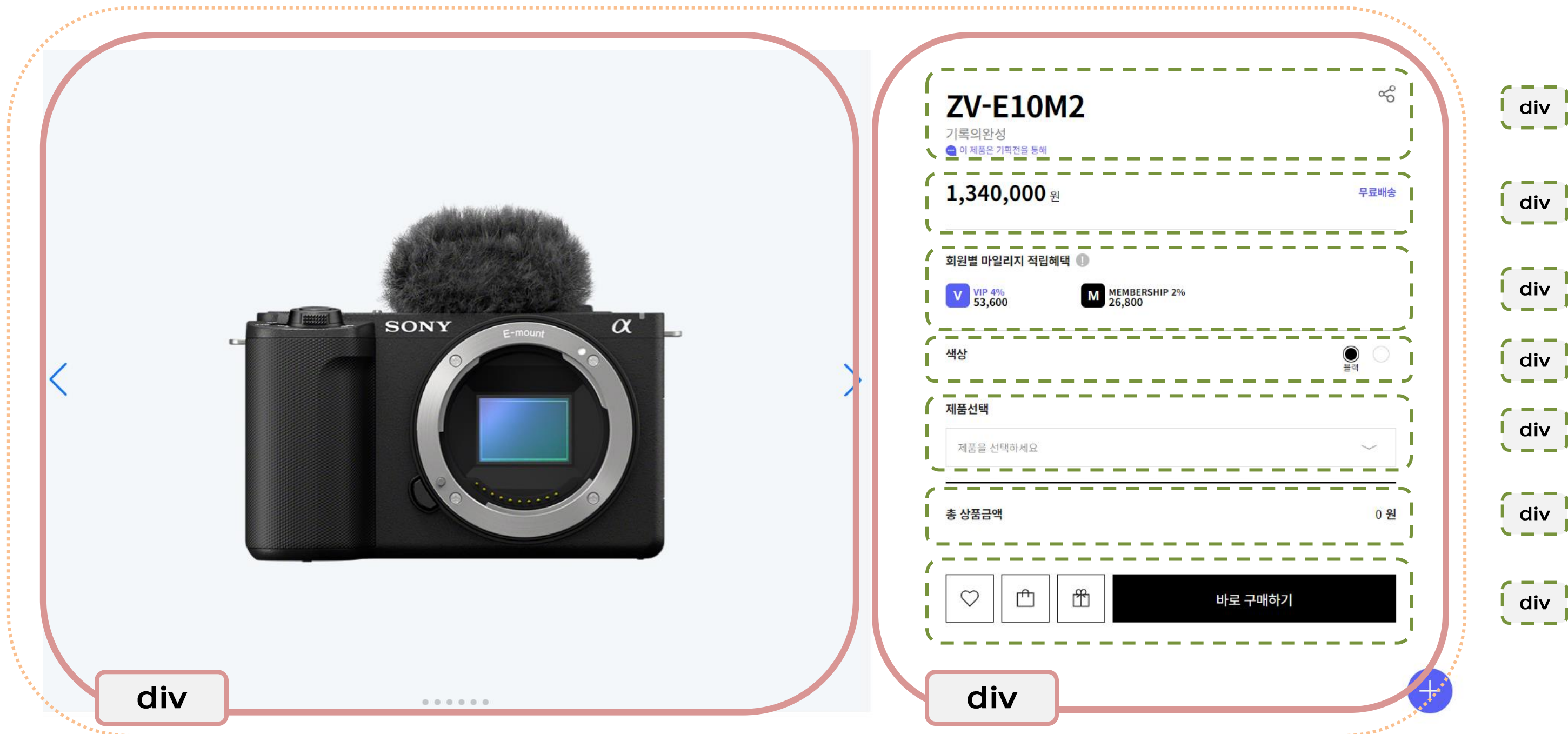
이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



1

# 레이아웃 분석 (상품 상세 페이지)

div



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

- 주요 내용 마다 div으로 분리
- div으로 div을 감싸는 형태

1

# 레이아웃 분석 (상품 상세 페이지)

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

## 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

p

선택

ul

li, a, span

span, span

제품선택

제품을 선택하세요

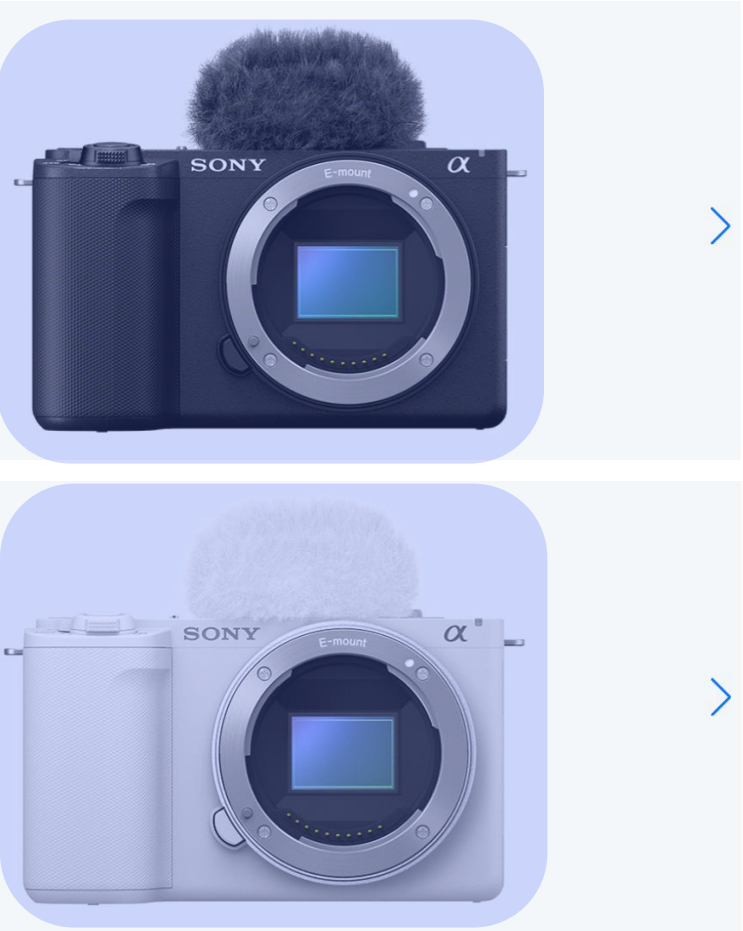
총 상품금액 0 원

♡

🛒

🎁

바로 구매하기



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

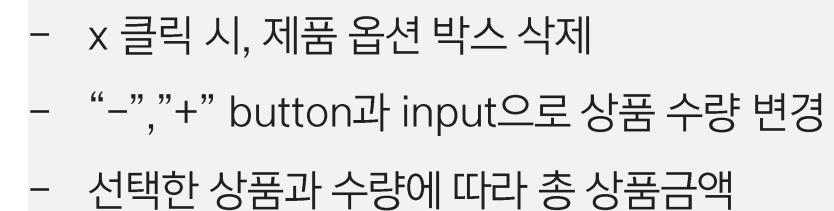
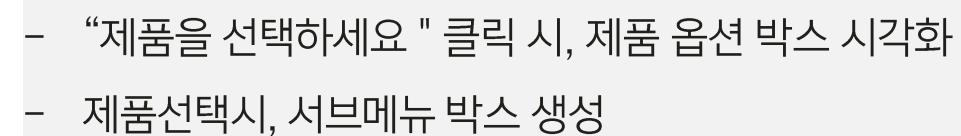
색상

제품선택

제품을 선택하세요

총 상품금액 0 원

# 레이아웃 분석 (상품 상세 페이지)



# 1

## 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

# 레이아웃 분석 (상품 상세 페이지)



– “제품개요”, “제품상세“, “배송/환불규정“ 호버or클릭 시, 색상으로 강조, 각 항목 시각화



## 제품 상세

### 일반

카메라 유형	렌즈 교환식 디지털 카메라
렌즈 마운트	E 마운트

### 카메라 섹션

센서 유형	APS-C 타입(23.3 x 15.5mm), Exmor R CMOS 센서
-------	--



## 배송 / 환불 규정

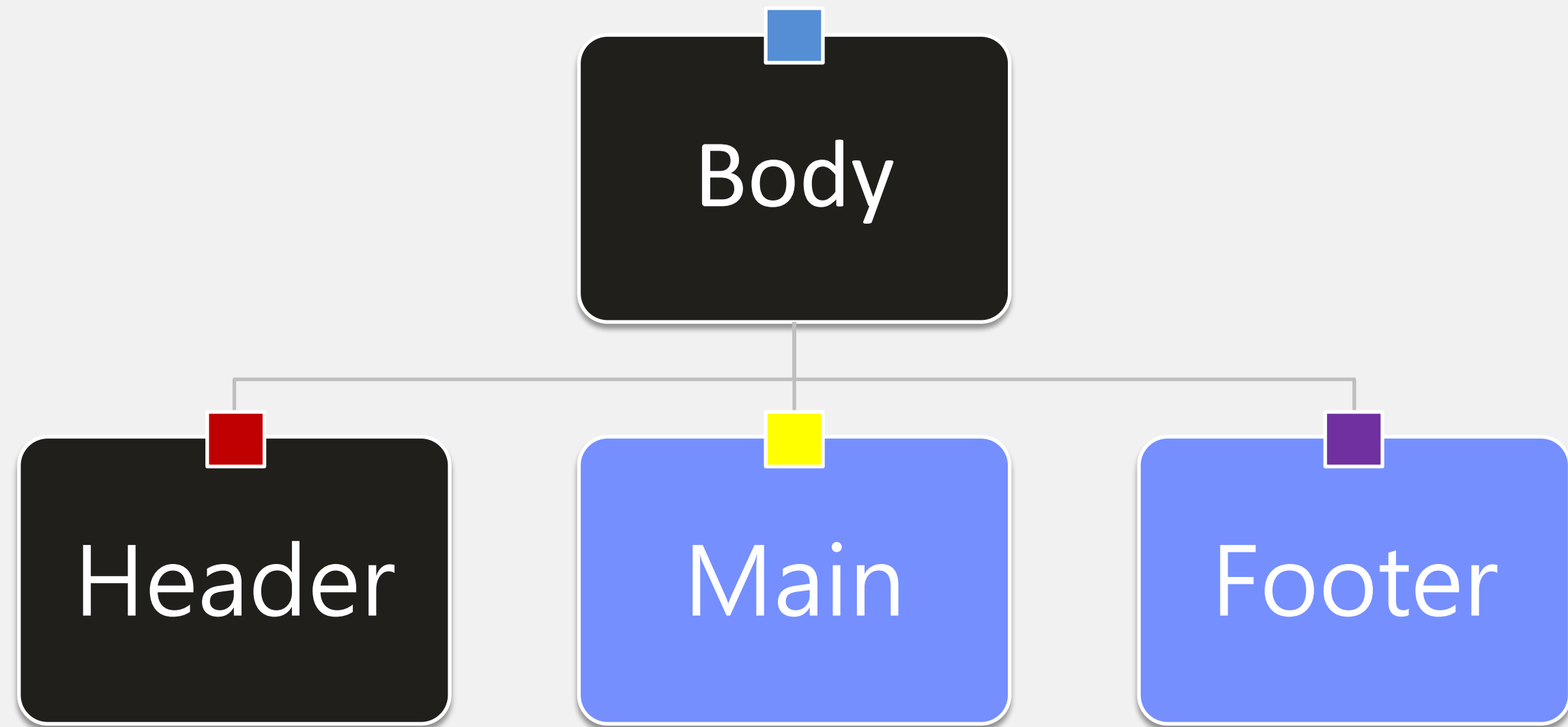
### 교환 / 반품 안내 및 보증 조건과 절차

교환 / 반품 기준	고객 변심으로 인한 교환/반품은 상품을 수령하신 날로부터 7일 이내에 신청 가능합니다. 반품 시 제공된 사은품은 모두 회수하며 회수가 되지 않을 경우 교환/반품이 불가능합니다. 교환/환불 시 제품 특성에 따라 처리 소요시간이 상이할 수 있습니다. 교환은 동일한 상품만 가능하며, 다른 상품으로 교환을 원하실 경우 해당 상품을 주문주소 후 재주문하여 주시기 바랍니다. (색상변경 포함) 전자상거래 등에서의 소비자 보호에 관한 법률로 규정되어 있는 소비자 청약철회 가능 범위에 해당하는 경우 신청 가능합니다. 책배송 등으로 주문하신 상품과 설치상품의 교환/반품 접수는 고객센터(1588-0911)로 연락주시기 바랍니다.
교환 / 반품 불가사유	포장을 개봉하여 사용하거나 또는 설치가 완료되어 상품들의 가치가 훼손된 경우 고객님의 단순변심으로 인한 교환/반품 요청이 상품으로 수령하신 날로부터 7일 이후에 접수된 경우

PART 2

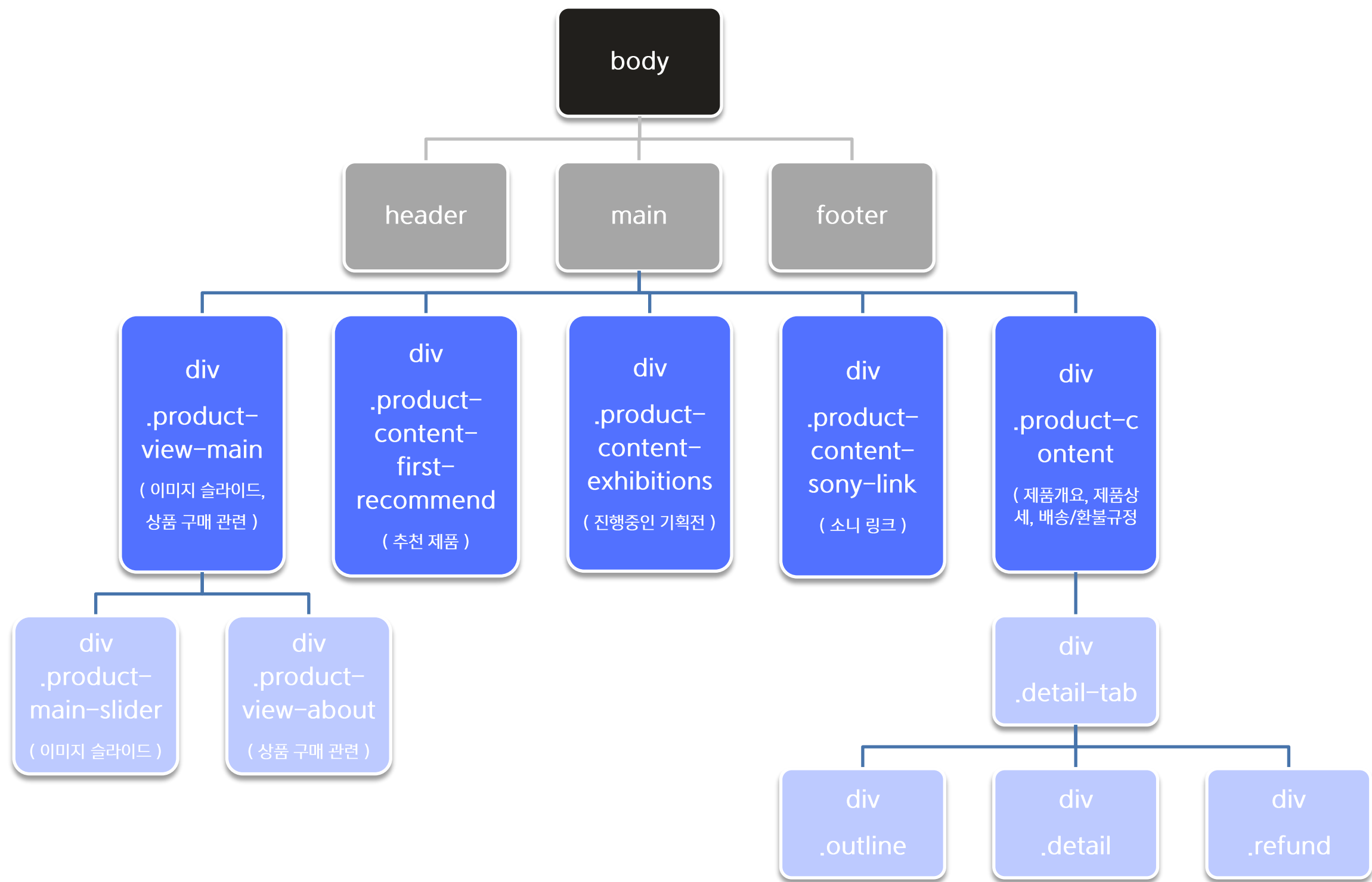
# DOM 구조 설정

## 2-1 DOM 구조 설정



2-1

# DOM 구조 설정 - 상품 상세 페이지

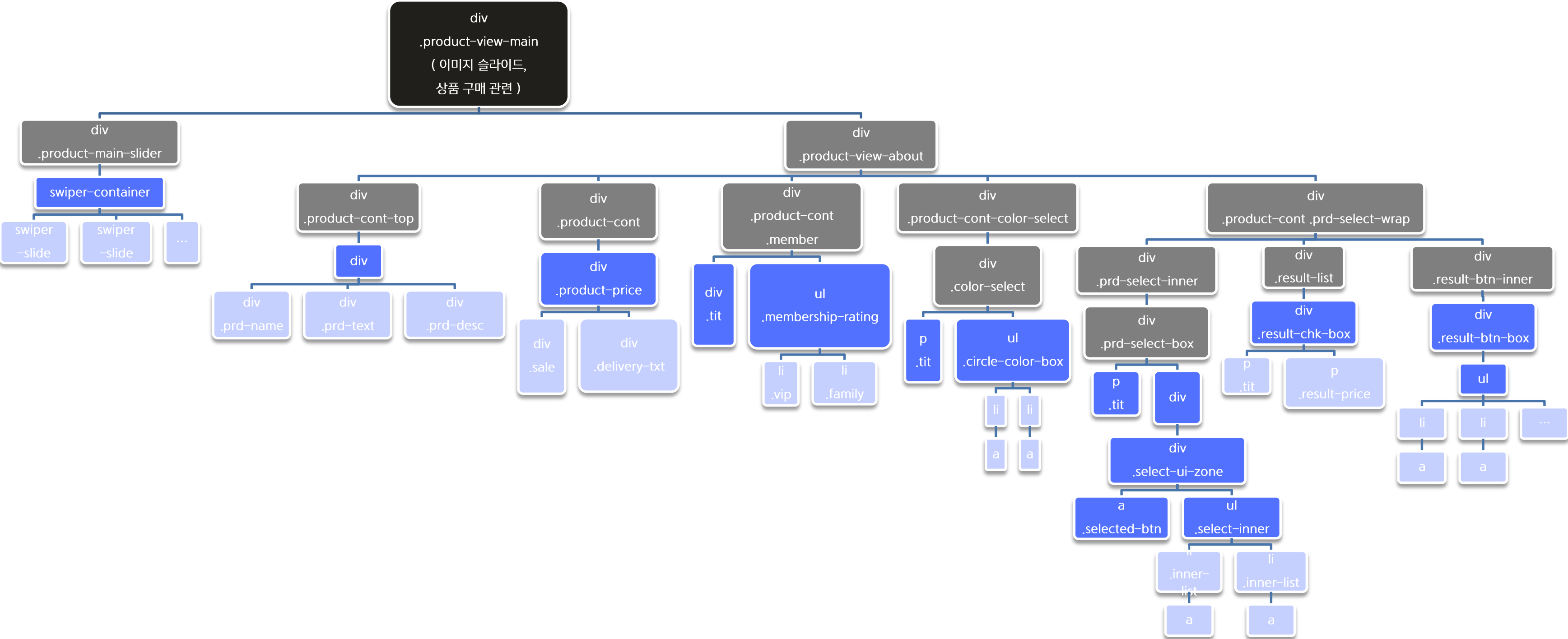




2-1

# DOM 구조 설정 - 상품 상세 페이지

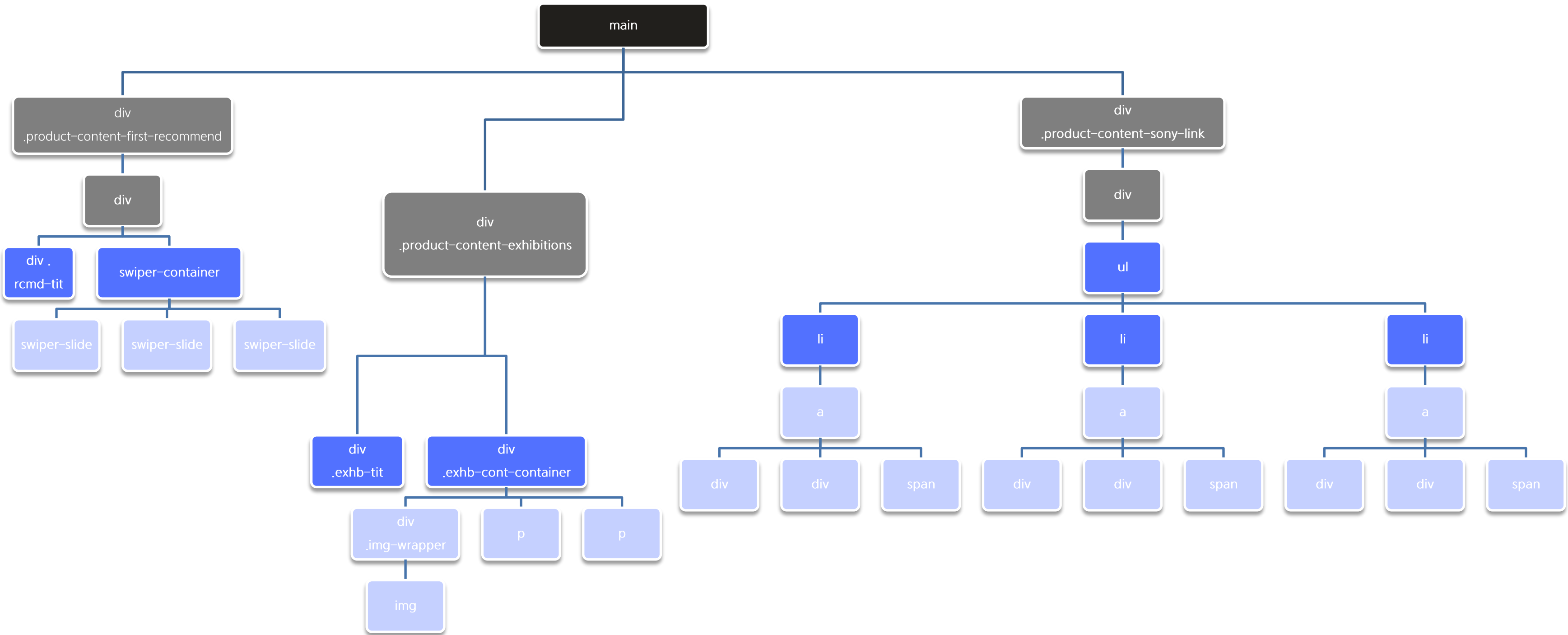
body > main >



2-1

# DOM 구조 설정 - 상품 상세 페이지

body >



PART 3

# 구현 결과

3

# 구현 결과(Header)

HTML

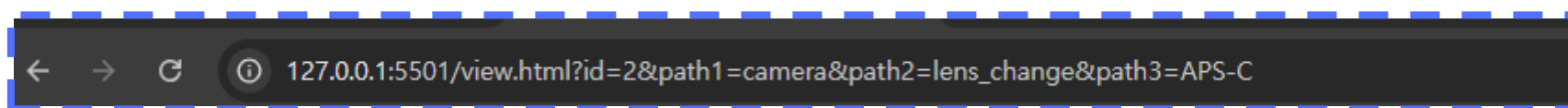


이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다.

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다.

## 구현 결과 ( 상품 상세 페이지 )



### JavaScript

```
( 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;  
    }  
}
```

## 구현 결과 ( 상품 상세 페이지 )



### 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<imgArr[0].length; i++ ) {
  const swiperSlide = document.createElement('swiper-slide');

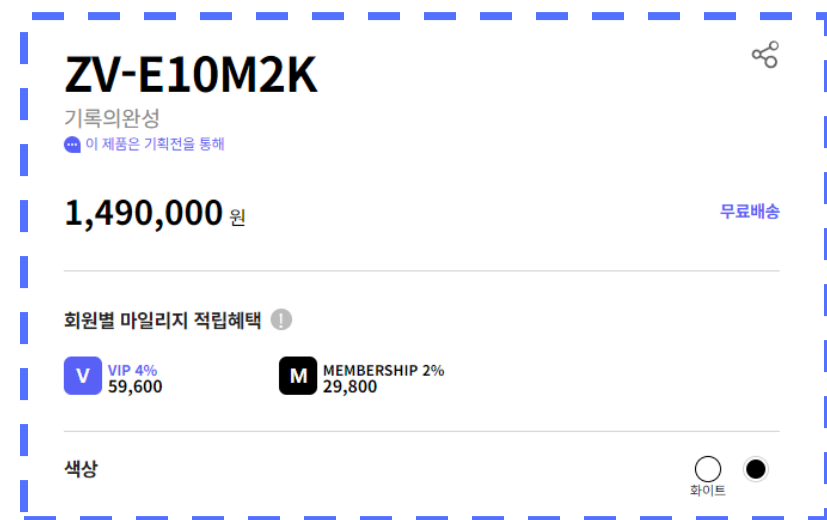
  const div = document.createElement('div');
  div.classList.add('img-container');

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

  div.appendChild(img);
  swiperSlide.appendChild(div);
  swiperContainerViewSwiper.appendChild(swiperSlide);
}
```

3

# 구현 결과 ( 상품 상세 페이지 )



## HTML

```
<!-- 메인 우측 - 구매관련 -->
<div class="product-view-about">    <!-- flex width 42% -->

    <!-- 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>
```

## JavaScript

```
/* -- -- -- 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();
```

```
/* -- -- -- 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++ ) {

        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]}` );
        });
    });
});
```



# 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>
</div>
```

```
/* 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 );

    span2 = document.createElement('span');
    span2.classList.add('submenu-txt');
    span2.classList.add('pad');
    span2.innerHTML = `${item.title} / ${clr_name(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');
});
```

```
/* --- 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) "ILCE-7CM2 / 실버"

    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');
    }
  });
});
```

- 선택한 옵션내용을 생성할 박스 안에 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
});

});
```

## JavaScript

```
/* --- circleColor 메서드 --- */
const circleColor = selector => {

  if( !selector ) return;

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

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

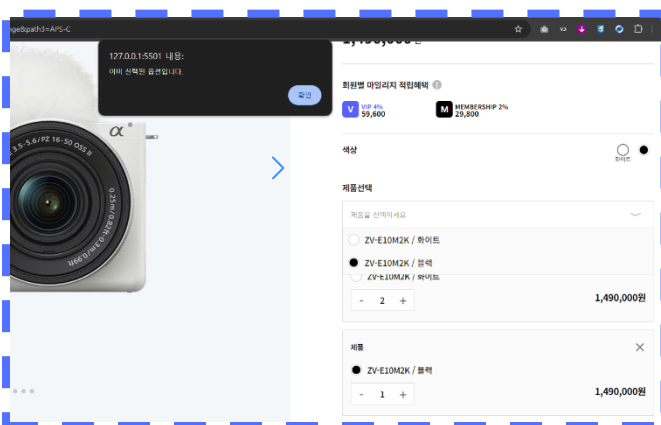
  const span2 = document.createElement('span');
  span2.classList.add('c-bg');

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

- 색상 원형을 재사용 하기 위해 메서드 생성

```
// 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.');
```





3

# 구현 결과 ( 상품 상세 페이지 )

JavaScript

HTML

```
<!-- 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 후 DOM 변화를 감지하기 위해  
MutationObserver 사용

```
/* -- -- -- 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`;
} );
```

```
/* -- -- -- 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();
```

제품

☐ ZV-E10M2K / 화이트

-

2

+

1,490,000원

제품

☒ ZV-E10M2K / 블랙

-

1

+

1,490,000원

총 상품금액

4,470,000 원

바로 구매하기



## HTML

```
<!-- 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>
```

```
/* -- -- 함께 구매하시면 좋은 추천 제품 -- -- */

/**
 * 현재 제품과 같은 category(camera, video, lens, ...) 인 객체들을 배열로 반환
 * @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] );    //console.log(productList);
```

```
> 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();
```

```
// 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;

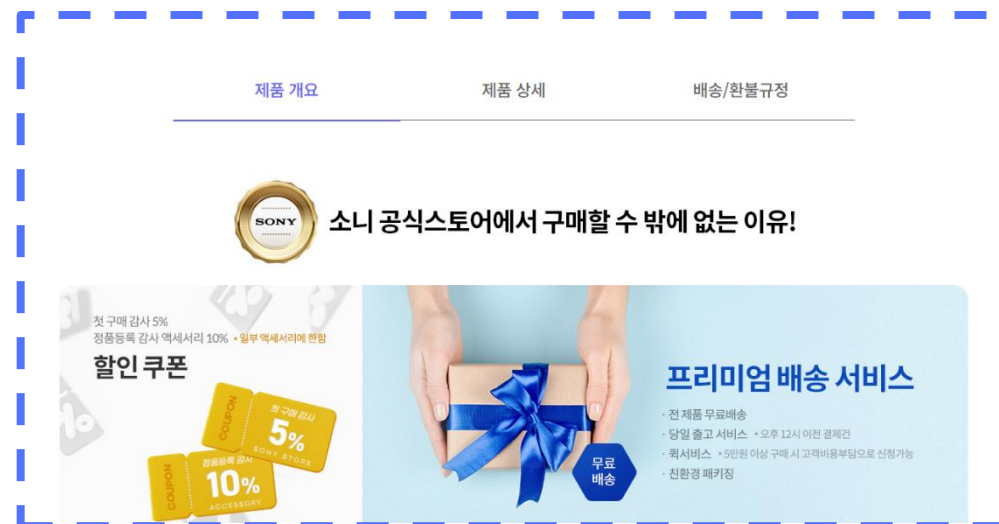
  const span3 = document.createElement('span');
  span3.classList.add('recommend-price');
  span3.innerHTML = v.price.toLocaleString() + '원';

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

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

- 각각의 상품들마다 클릭 시, 클릭한 상품의 상세페이지로 이동

# 구현 결과 ( 상품 상세 페이지 )



## 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

4

# 소요 시간 및 후기

01

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



여기에 항목을 입력해주세요

02

이 곳에 원하는 내용을 자유롭게 채워 넣으세요. 이 문장은 예시를 위한 더미 텍스트입니다. 실제 내용과는 무관합니다. 이 곳에 원하는 내용을 자유롭게 채워 넣으세요.



여기에 항목을 입력해주세요

승현

소요 시간 : 65시간

후기 :

비동기 처리 부분에서 생각지 못한 이슈들이 많이 발생했고, 그 부분들을 해결하면서 많이 배운 것 같습니다.  
“이렇게 하는게 맞나?” 싶은 부분들이 많아, 끊임없는 의문의 연속으로 답답한 부분들이 많았습니다.  
여러 페이지를 연동해야 하면서 데이터의 구조의 중요성을 알게 되었고, 파이널 프로젝트 전에 많은 수정이 필요할 것으로 예상됩니다.  
팀으로 처음 작업 하다보니, 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 없다보니 미리 확정 짓고 넘어가야 했던 것들을 그렇게 하지 못해 작업이 효율적으로 되지 못했던 것 같아 아쉬웠습니다.



여기에 항목을 입력해주세요

THANK YOU

감사합니다

123-456-7890

hello@reallygreatsite.com

제품 검색하기

기본 정보

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

쿼리 파라미터

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

Document

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