



에뛰드 하우스

최지민





목 차

1


사용기술

2

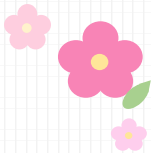
사이트 분석

3

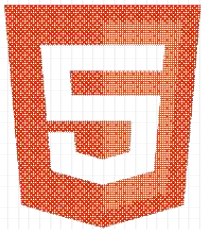
주요 기능



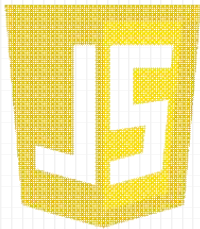
01 : 사용기술



HTML



JS

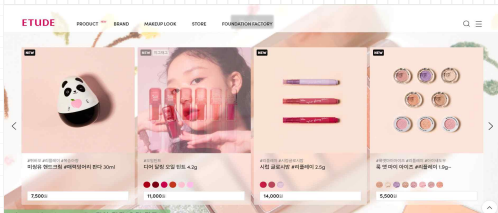


CSS

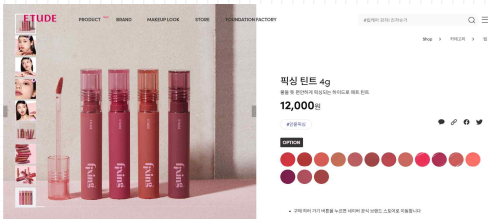


02 : 사이트 분석

분석1



분석2



1. 다양한 광고 및 상품 표출

사이트에는 다양한 광고 및 상품이 여러 개 나열되어 있음. 이는 사용자에게 다양한 제품 및 캠페인을 제공하여 선택의 폭을 확장하려는 의도로 보임.

2. 마우스 오버에 따른 사진 변환 기능

사용자에게 시각적 효과를 제공하며 제품에 대한 세부 정보를 미리 보여주어 사용자 경험을 향상시킬 수 있음.

3. 밝은 분위기

사용자에게 상쾌하고 긍정적인 느낌을 전달하여 브랜드의 이미지를 강화하려는 목적으로 보임.

03 : 주요 기능 (1)



```
// 네비 스크롤 내리시 hide 처리하기
let lastPosiont = 0;
const nav = document.querySelector(".nav-wrap");

const handleScroll = () => {
  let scrollPosition = window.scrollY || window.pageYOffset;
  console.log(scrollPosition);
  if (scrollPosition > lastPosiont) {
    nav.classList.add("hide");
    console.log("hi");
  } else {
    nav.classList.remove("hide");
  }
  lastPosiont = scrollPosition;
};

window.addEventListener("scroll", handleScroll);
```

페이지를 스크롤할 때 네비게이션 바를 숨기거나 표시하는 이벤트 처리

1. lastPosition이라는 변수를 사용하여 현재 스크롤 위치와 비교하기 위한 이전 스크롤 위치를 추적한다.
2. handleScroll 함수는 스크롤 이벤트가 발생할 때마다 호출된다.
3. 현재 스크롤 위치를 가져오고 이전 위치와 비교하여 스크롤이 아래로 내려갈 때는 네비게이션 바에 hide 클래스를 추가하고 위로 올라갈 때는 hide 클래스를 제거한다.

03 : 주요 기능 (2)



● LIP

막싱틴트
12,000원



SKINCARE

AC 클린업 링크 파우더 스킨 15ml
14,000원



● LIP

베러 립스-톡 3.5g
13,000원



● EYE

플레이 컬러 아이즈
30,000원



● EYE

컬 픽스 마스카라
15,000원



● LIP

막싱틴트
12,000원



● BODY

해피비중 코튼스노우 바다워시
300ml
12,000원



● EYE

플레이 컬러 아이즈
30,000원



● EYE

컬 픽스 마스카라
15,000원



● NAIL

헬프 마이 핑거 네일 톱거택 12ml
1,500원

```
const shuffledData = main1Data.sort(() => Math.random() - 0.5);
const uniqueImg = new Set();
const selectedData = shuffledData
  .filter((v) => {
    if (!uniqueImg.has(v.img)) {
      uniqueImg.add(v.img);
      return true;
    }
    return false;
  })
  .slice(0, 5);
```

```
const main1Bx = document.querySelector(".main1-wrap");
```

중복되지 않는 이미지를 랜덤으로 데이터 5개를 선택

1. main1Data 배열을 무작위로 섞어 shuffledData에 저장한다.
2. uniqueImg라는 Set을 만들어 이미지 중복을 방지한다.
3. shuffledData 배열에서 중복되지 않는 이미지를 가진 데이터를 최대 5개 선택하여 selectedData에 저장한다.

03 : 주요 기능 (3)



```
const main2Img = [  
  "../image/main/main2/img1.jpg",  
  "../image/main/main2/img2.jpg",  
  "../image/main/main2/img3.jpg"  
];  
  
const main_img_bx = document.querySelector(".main-img-bx");  
const main_img = document.createElement("img");  
main_img.classList.add("main-img");  
  
const randomImg = Math.floor(Math.random() * 3);  
main_img.setAttribute("src", main2Img[randomImg]);  
  
main_img_bx.appendChild(main_img);
```

랜덤 이미지 선택

1. `Math.random()` 함수를 사용하여 0 이상 1 미만의 난수를 생성하고, `Math.floor()` 함수를 사용하여 소수점을 버린 뒤 0부터 2까지의 정수를 얻는다.
2. `randomImg`에 저장된 값에 해당하는 `main2Img` 배열의 이미지 경로 중 하나를 선택하여 `main_img`의 `src` 속성에 설정한다.

03 : 주요 기능 (4)



#보습크림 #순정

순정 2x 베리어 보습 크림

17,000원



#순정 #앰플에센스



입꼬리 메이커 0.8g

12,000원

크레마 벨벳 틴트 3.6g

12,000원



#룩엣마이아이즈 #리플레이 #아이세도우
(리플레이) 룩엣 마이 아이즈

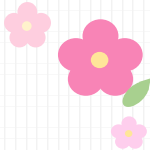
5,500원

```
const changeImgFn = (index) => {  
  const img = document.getElementById(`image${index}`);  
  img.src = `../image/main/main4/${index + 1}-2.jpg`;  
};  
  
const restoreImgFn = (index) => {  
  const img = document.getElementById(`image${index}`);  
  img.src = main4Data[index].img;  
};
```

마우스 오버 / 아웃시 해당 이미지 변경

1. 마우스 오버 및 아웃 이벤트가 발생하면 각각 `changeImgFn`과 `restoreImgFn` 함수를 호출한다.
2. `changeImgFn` 함수는 인덱스를 받아와서 해당 인덱스의 이미지의 소스 경로를 변경함.
3. `restoreImgFn` 함수는 인덱스를 받아와서 해당 인덱스의 이미지를 초기 경로로 복원함.

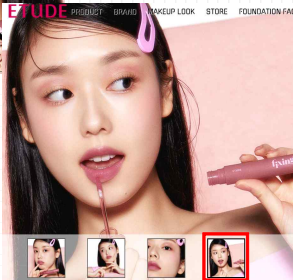
03 : 주요 기능 (5)



```
const changeSubImg = (event) => {  
  const clickedSubImg = event.target;  
  subImg.querySelector('img').src = clickedSubImg.src;  
  subSubImg.forEach(ele => ele.classList.remove('click'));  
  clickedSubImg.classList.add('click');  
};
```

여러 개의 서브 사진 중 하나를 클릭하면
해당 사진이 메인 이미지로 표시

1. changeSubImg 함수는 서브 사진을 클릭할 때 호출되는 함수임.
2. 클릭된 서브 사진의 소스(src)를 메인 이미지로 설정하고, 클릭된 사진에 "click" 클래스를 추가하고 나머지 사진들에게는 "click" 클래스를 제거함.



03 : 주요 기능 (6)

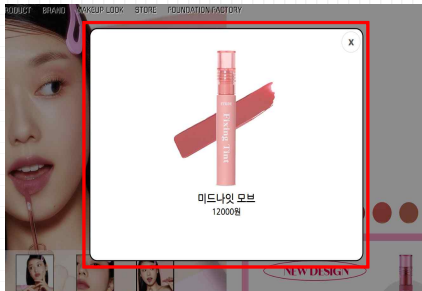
픽싱틴트 4g

물들 듯 편안하게 픽싱되는 하이드로 매트 틴트

12,000원

#안물픽싱

OPTION



제품 색상 클릭시 모달창 구현

1. for 루프로 틴트데이터 배열의 길이만큼 반복하여 틴트 제품에 대한 각각의 이미지를 temp 배열에 문자열 형태로 추가한다.
2. 틴트 제품 이미지를 클릭하면 모달 창에 해당 틴트 제품의 정보를 표시한다. 클릭된 이미지의 정보를 인자로 받아와 모달 창에 정보를 업데이트하고, 모달 창을 화면에 보이게 한다.
3. `document.querySelector(".close-btn").onclick`: 닫기 버튼을 클릭하면 모달 창을 숨기고 이전에 가려져 있던 요소를 다시 보이게 한다.

03 : 주요 기능 (7)



텍스처	부드럽게 픽싱되어 보이는 컬러 그대로 생성하게 지속되는 세미 매트 텍스처	물들듯 촉촉한 발림성과 오랜 컬러 지속력의 하이드로 매트 텍스처	끈적임 없이 시간이 지날수록 투명하고 은은한 광이 차오르는 글로우 텍스처
광택감	 보송광	 매끈한 무광	 물막광
발색력			
미무리	매끈한 블러딩 미무리	건조함없는 보송한 미무리	촉촉한 코팅 광택 미무리
컬러	피부톤이 화사해 보이는 선명하고 맑은 컬러감	베이스부터 포인트까지 레이어링이 용이한 내추럴 컬러 레인지	침아하고 맑은 투명한 컬러감



```
subBx1Btn.onclick = function () {
  subBx1Wrap.classList.toggle("on");

  const icon = subBx1Btn.querySelector(".fa-solid");
  icon.classList.toggle("fa-caret-down");
  icon.classList.toggle("fa-caret-up");

  const spanEle = subBx1Btn.querySelector("span:first-child");
  spanEle.textContent = subBx1Wrap.classList.contains("on") ? "닫기" : "더 보기";
};
```

더보기 / 닫기 버튼

1. "더보기" 버튼을 클릭하면 "on" 클래스를 토글(toggle)하여 나타내거나 숨기는 효과를 줍니다.
2. "on" 클래스가 존재하는 경우, 아이콘의 클래스를 "▼"에서 "▲"으로 변경한다.
3. "on" 클래스가 존재하는 경우 버튼 내의 텍스트를 "더보기"에서 "닫기"로 변경한다.
4. "on" 클래스가 없는 경우에는 그 반대로 변경한다.



03 : 주요 기능 (8)



```
const goExplanationArea = () => {
  if(stsClick) return;
  btnScrollly = window.pageYOffset
  if(btnScrollly > 0 && btnScrollly < subBtn2){
    subBtn.forEach((ele) => ele.classList.remove('redBtn'));
    subBtn[0].classList.add('redBtn');
  }else if(btnScrollly >= subBtn2 && btnScrollly < subBtn3){
    subBtn.forEach((ele) => ele.classList.remove('redBtn'));
    subBtn[1].classList.add('redBtn');
  }else if(btnScrollly >= subBtn3){
    subBtn.forEach((ele) => ele.classList.remove('redBtn'));
    subBtn[2].classList.add('redBtn');
  }
}
```

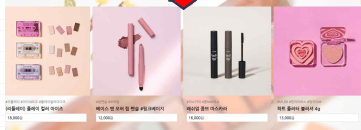
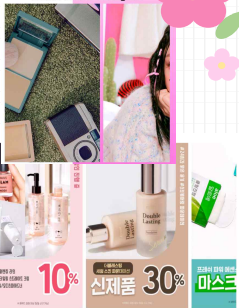
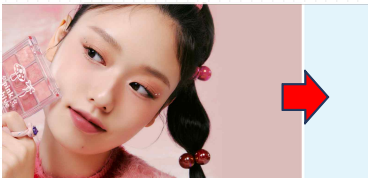
```
const colorChange = (event) => {
  stsClick = 1;
  const clickedElement = event.target;
  // event.preventDefault();

  // 클릭된 요소가 subBtn에 포함된 것인지 확인
  if (Array.from(subBtn).includes(clickedElement)) {
    console.log('hi click')
    // 클릭된 요소에 "redBtn" 클래스를 추가하고 나머지 요소들에서는 제거
    subBtn.forEach(x => {x.classList.remove("redBtn")});
    clickedElement.classList.add("redBtn");
  }
}
```

스크롤 또는 클릭 이벤트에 섹션 버튼의 색상을 변경

1. `getBoundingClientRect()`를 이용하여 각각 서브 페이지의 특정 섹션의 상단 위치를 저장한다.
2. 스크롤 이벤트 핸들러에서는 `window.pageYOffset`을 이용하여 스크롤 위치에 따라 버튼 색상 변경한다.
3. 클릭이벤트 핸들러에서는 클릭된 버튼에 색상 클래스를 추가하고 나머지 버튼에서는 제거한다.
4. `stsClick`는 클릭 상태를 나타내는 변수로, 초기값은 0이며, 클릭이 발생하면 1로 설정된다.

03 : 주요 기능 (9)



Swiper 라이브러리를 사용하여 슬라이드 생성

1. 각 슬라이드를 Swiper에 추가한 후, Swiper를 업데이트하는 기능을 구현한다.
2. 해당 데이터를 순회하면서 각각의 슬라이드를 생성하고, 이를 Swiper의 wrapper에 추가한 뒤, 업데이트를 통해 새로운 슬라이드가 반영되도록 한다.



THANK YOU