

0 0 0 프로젝트 소개

## Q 프로젝트 소개



이번 프로젝트의 주요 목표는 HTML과 CSS를 활용해 화면을 구현하고, JavaScript의 Ajax를 활용하여 카카오 open API로부터 데이터를 요청 및 수집하여 웹 애플리케이션을 개발하는 것이었습니다. 이를 위해 기존의 Open API 기반 인터넷 서점 사이트인 교보문고의 구성, 레이아웃, 이미지 등을 참고하였으며, 일부 변형하여 프로젝트를 완성하였습니다.



교보문고는 대한민국의 대표적인 인터넷 서점으로, 방대한 도서 정보와 다양한 카테고리, 편리한 검색 기능을 제공하는 사이트입니다. 또한 사용자 친화적인 레이아웃과 직관적인 디자인으로 도서 정보를 쉽게 확인하고 구매할 수 있으며, 출판사, MD, Al 기반의 유용한 추천 기능을 갖추고 있습니다. 0 0 0 기획

## **Q** Persona



• 이름 : 호 랑

• 나이:27세

• 직업: 프리랜서

• 취미:만화책

구매, 영화 관람

## [라이프 스타일]

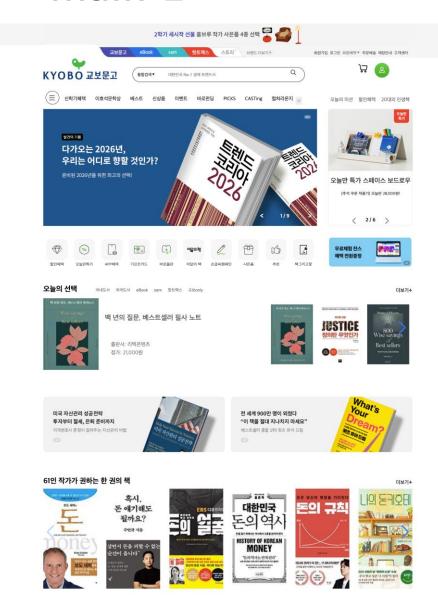
27세 프리랜서로 자유로운 생활 패턴을 가지고 있으며 주로 새벽에 일하고 오후에는 취미 생활을 즐긴다. 만화책을 구매해 읽고 모으는 것을 좋아하며 최근에는 e-book에도 큰 흥미를 가지고 있다. 평소에는 영화 감상으로 영감을 얻으며 새로운 스토리와 문화를 접하는 것을 좋아한다.

## [바라는 점]

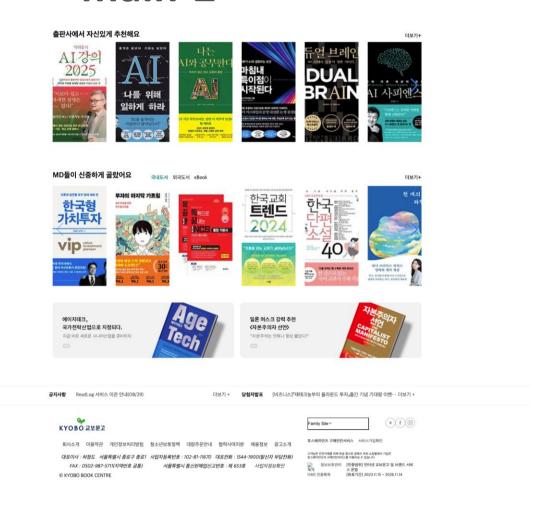
- 만화책을 포함해 e-book을 쉽게 구매하고 관리할 수 있는 사용자 친화적인 기능
- 전문적인 추천 도서나 맞춤형 가이드 콘텐츠



## <main-1>



## <main-2>



## <sub-1>



## <sub-2>

100	조성이 속 문장을 통한 병사와 일반의 식원
10 10 10 10	는 등 다 나는 사랑이 다가에도 안되를 되어보고, 전혀를 되어서는 무슨만 않는데 필요합니다. 그리고 세상에 들간한 수선, 수이 관리 무슨에서 그녀는 생산을 좀 수 있습니다. 자리가 집중점한 중앙 가운동, 그녀고 그 가는을 제상을 동일을 보니까 쓴 역동 그 수십년 중에서 영화점을 받아 수신한 육강한 '역 군 등 가격, 세스스템에 수 영한 바라'는 수 작가들이 해결에 찾는 문제를 남았습니다. 5세대 기상 없는 공단을 받는 '많이에 제외 문장을 선생하고 있다'을 주기하여 역 관리 일본, 에스트컵은 당시나라진다는 역을 즐긴 생성하다.
100	강마시어 됐다. 나를 되었어요. 시간 '시간' 나는 의명하면, 각 에면 경면 동네성에 동안을 만했으면서 아이템에 스스로를 이 사용하고 선 - 그런 안내면 JCL CAMT 의명을 통계 되는 건물 너희 설명 동에서고 나를 경안하는 스마리 시간을 건물한 건입니다. 기구에 당긴 하는 당신에 대한되었다. 글이어 하십시나는 건 비를 소해서 한 출시 힘이 되어를 만한다며, 느낌이어 설명 전혀나면 가지 됩니다. 등에 에 한 성명 한 경영을 수줍니다. 느끼고, 집단에 강하여 있다. 흔하여가 되는 경기를 어느록 스스로를 가 할 때에서는 하대로 하나 것 5
#	한 심과 하는 사건 전혀, 전건경에 등 Hamori 실어되었다. 한 지원한 그러져 있었다. 등업에 당당한 사내한 기차에 대한되는 현존 사서 주고 있습니다. 전한 시대한 사진에 자리에 자리를 즐겁고 할것같은 제공에 할 수 있는 건강에, 이사를 되어 있다는 제작회에서 용성을 자금할 것입니다. 이 에너 자리에는 하루에 하스한 등요가 되기를 하던데, 등사는 일반이 만들어되는 점점 예쁘게 당신한 지 수가기를 당긴하게 될 것입니다.
114	ing ing
	TORREST HOUSE TO
1869	BILD 에서 한지시 화매에게 하는 것은 에너의 당시는 성격성의 활항을 당했다는 것이 하니죠. 상성동안 나를 힘들어 했던 내 이름에서 여름이 들는 말이다. 때면서 수십에고 막을 한 번째 문제는 당면에면 용서는 낮은 일을 위해서 할 수 있는 하는데 단체하다 - 또, "사람에 들어서 적정한 공단에게
181	들지 옷은 이용을 바로고자 역시와 그렇다면 어떤 느리를 하고 있나요?
	등만원의 주는 제기업사 전에 #대표함은 ##하였다.
4.5 88	를 보고 한민단 남에에서 자근에게 다음과 같이 돌아야 "아니라면 나비들면이 모든 건강되는 나비들면이 모든 아이들을 바꾸면 잘 성도한 경험을 내려왔다. 전혀되어서 한자시간이 슬픈도 갖다 내려들면이 되어 나는 물론으로 내 자신이 중앙되어 내 성도 다른 나는 생긴에, "이 한민을 두 내 한 한쪽이다. EL MOL "이프는.
	경영에 나를 가장 무겁게 하고 있나요? 등을 시작들이 더 어떤 역에는 비슷하죠?

\$5.00 Seeding of Columns	men diam's more	ortection with their he real limit to	
84090 2882		- C(1)	
MAIN ARRY ADMINISTRA ALVAN	OM 400004 SAVOR MADE STATE	STATE LABOUR. SCI. DOZ.	
	ME THE THE GREE THE STREET	From Manney and	

0 0 0 구현

### **CSS**



# Q 메인 페이지 코드 구현 (1)

발견의 기쁨

다가오는 2026년,

준비된 2026년을 위한 최고의 선택!

우리는 어디로 향할 것인가?



```
mySwiper_2 {width: 274px; height: 380px; margin: 0 auto;}
 swiper_content {width: 274px; height: 200px; margin-bottom: 20px;}
.swiper_text {width: 274px; height: 130px; text-align: center; padding: 10px; font-size: 20px;}
 swiper_number {width: 274px; height: 35px; margin-bottom: 5px;}
 slider_b_btn {display: flex; gap: 0.25rem; z-index: 10; bottom: 0.75rem; background-color: ■aqua; right: 0.75rem;}
 slider1_side {width: 274px; height: 380px; border: 0.1px solid ■#eaeaea; border-radius: 0.75rem;}
 .slider1_side_slide {width: 274px; height: 330px;}
 slider1_side_slide_t {width: 274px; height: 200px;}
 slider1_side_slide_t img { border-radius: 0.75rem; border-style: 1px solid ■#bbb;}
 slider1_side_slide_b {width: 274px; height: 130px; text-align: center; padding-block: 1.75rem;}
 slider1_side_slide_b p {font-size: larger; font-weight: 700;}
 slider1_side_slide2 {width: 274px; height: 50px; display: flex; justify-content: center; gap: 0.9rem;}
 slider1_side_slide2 button {border: none; background-color: ■white; align-items: center;}
 mySwiper_2 .swiper-controls {display: flex; justify-content: center; align-items: center; gap: 12px; margin-top: 16px;}
 .mySwiper_2 > .swiper-controls .swiper-button-next {bottom: -1; top: auto; right: 70px; color: □#333;}
 .mySwiper_2 > .swiper-controls .swiper-button-prev {bottom: -1; top: auto; left: 70px; color: □#333;}
.mySwiper_2 > .swiper-controls .swiper-button-prev::after {font-size: 20px;}
 .slide_number {position: absolute;
bottom: 10px; /* 부모 기준 아래에서 10px */
left: 50%;
transform: translateX(-50%);
font-weight: bold;
font-size: 16px;}
```

### **JavaScript**

```
var swiper = new Swiper(".mySwiper", {
   pagination: {
   el: ".swiper-pagination",
   type: "fraction",
   loop: true,
   autoplay:
       delay: 5000,
       disableOnInteraction:false
   navigation: {
   nextEl: ".swiper_controls_1 .swiper-button-next",
   prevEl: ".swiper_controls_1 .swiper-button-prev",
var swiper = new Swiper(".mySwiper_2", {
 pagination: {
   el: ".swiper-pagination",
   type: "fraction",
 loop: true,
 autoplay: {
       delay: 6000,
       disableOnInteraction:false
 navigation: {
   nextEl: ".swiper-button-next",
   prevEl: ".swiper-button-prev",
```

헤더 바로 밑 첫 번째 섹션에 책과 웹사이트 정보를 담은 슬라이드를 배치하여, 화살표 버튼과 자동 재생 기능을 통해 페이지가 전환되도록 구현함.

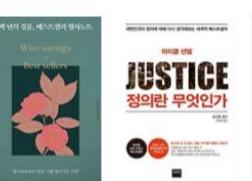
더보기+

## ○ 메인페이지코드구현(2)

leftswiper\_bookinfo {width: 500px; height: 100px; margin-top: 50px; padding: 10px; line-height: 1.6; font-size: 17px; margin-left: 10px; color: □#777;









### **CSS**

.left-swiper,

width: 1200px;

```
JavaScript
margin: 0 auto 30px;
                                                                                                 // section3 open API 책
slider-container {display: flex; width: 1200px; height: 260px; margin: 0 auto; margin-top: 10px;}.
                                                                                                 async function fetchBooks2(query) {
                                                                                                       const REST_API_KEY = '7ec0ff3e208807bf25acfcb953642046'
width: 1200px; height: 260px;
                                                                                                       const params = new URLSearchParams({
                                                                                                            target: "title",
.mySwiperLeft {
                                                                                                            query,
width: 730px;
                                                                                                            size: 50
height: 233px;
margin-right: 20px;
                                                                                                       const url = `https://dapi.kakao.com/v3/search/book?${params}`;
* .mySwiperLeft .swiper-slide {
                                                                                                       const response = await fetch(url, {
                                                                                                            method: 'GET',
.mySwiperRight {
                                                                                                            headers: {
                                                                                                                Authorization: "KakaoAK 7b2300fc6315bb65035d1a3c7b49b161"
height: 195px;
margin-top: 20px;
                                                                                                       });
background-color: #d0e0f0;
                                                                                                       if (!response.ok) {
                                                                                                            throw new Error(`HTTP 오류: ${response.status}`);
.swiper-button-prev {
                                                                                                       return response.json();
leftswiper_content {width: 730px; height: 262ppx; display: flex;}
.mySwiperLeft {overflow: visible; position: relative;}
mySwiperLeft img {width: 180px; height: 260px;}
leftswiper_text {width: 528px; height: 146px; margin-top: 30px; text-align: left; padding-left: 20px; font-size: 23px; padding-top: 20px;}
```

```
async function bookData2() {
       const querys = ["베스트셀러"];
       for (const q of querys) {
           const data = await fetchBooks2(q);
           console.log(data);
            const origin = data.documents;
            let book = origin.filter((val)=>{
               return val.thumbnail != '' && val.contents !='';});
           // .box 요소 전체 선택
       const swiperContainer = document.querySelector(".mySwiperRight");
       const boxElements = swiperContainer.querySelectorAll(".swiper-slide");
       // documents 데이터를 각 box에 대응하여 렌더링
       boxElements.forEach((box, i) => {
           const doc = data.documents[i];
           if (!doc) return; // 데이터가 부족할 경우 생략
```

두 개의 슬라이드를 하나의 섹션으로 묶어, 슬라이드 전환 시 왼쪽과 오른쪽 슬라이드가 동시에 넘어가도록 구현함.

03 구현

#### $\circ$ 0

# 메인 페이지 코드 구현 (3)



#### 61인 작가가 권하는 한 권의 책



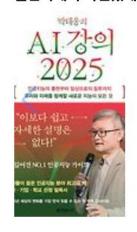








#### 출판사에서 자신있게 추천해요

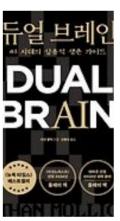














### CSS

```
.section_5 {width: 1328px; height: 378px; margin: 0 auto; margin-top: 5rem;}
.section_5_box1 {width: 1200px; height: 34px; margin: 0 auto; display: flex;}
.section_5_box1_h2 {width: 1145px; height: 34px; margin: 0 auto;}
section_5_box1_a {margin-left: auto; padding-top: 10px; font-weight: 700;}
.section_5_box2 {width: 1328px; height: 324px; margin: 0 auto; align-items: center;}
section_5_box2 >div {width: 1200px; height: 325px; margin: 5px auto; display: flex;}.
```

### **JavaScript**

```
/ section 5 open API
async function fetchBooks(query) {
        const REST_API_KEY = '7ec0ff3e208807bf25acfcb953642046'
        const params = new URLSearchParams({
            target: "title",
            query,
            size: 50
        const url = `https://dapi.kakao.com/v3/search/book?${params}`;
        const response = await fetch(url, {
            method: 'GET',
               Authorization: "KakaoAK 7b2300fc6315bb65035d1a3c7b49b161"
       });
       if (!response.ok) {
            throw new Error(`HTTP 오류: ${response.status}`);
        return response.json();
    async function bookData() {
           const querys = ["\E"];
            for (const q of querys) {
               const data = await fetchBooks(q);
                console.log(data);
                 const origin = data.documents;
                let book = origin.filter((val)=>{
                    return val.thumbnail != '' && val.contents !='';});
```

```
// .box 요소 전체 선택
       const swiperContainer = document.querySelector(".mySwiper_3");
       const boxElements = swiperContainer.querySelectorAll(".swiper-slide");
       // documents 데이터를 각 box에 대응하여 렌더링
       boxElements.forEach((box, i) => {
          const doc = data.documents[i];
           if (!doc) return; // 데이터가 부족할 경우 생략
          const img = document.createElement("img");
           img src = doc thumbnail;
           box.appendChild(img);
          const h3 = document.createElement("h3");
          h3.textContent = doc.title;
           box.appendChild(h3);
  } catch (error) {
       console.log('에러발생', error);
bookData();
```

카카오 open API를 활용해 관련 주제의 책 정보를 query로 요청하고, 이를 슬라이드 형식으로 보여주도록 구현함.

03 구현

### 0 0 0

# Q 메인 페이지 코드 구현 (4)





#### CSS

```
.section_7 {width: 1328px; height: 378px; margin: 0 auto; margin-top: 5rem;}
.section_7_1 {width: 1200px; height: 34px; margin: 0 auto; display: flex; justify-content: space-between; align-items: center; margin-bottom: 15px;}
.section_7_1_h2 {width: 320px; height: 34px;}
.section_7_1_tab {display: flex; gap: 15px;}
.section_7_1_a {margin-left: auto; padding-top: 10px; font-weight: 700;}
.section_7_box2 {width: 1328px; height: 324px; margin: 0 auto; align-items: center;}
.section_7_box2 >div {width: 1200px; height: 325px; margin: 5px auto; display: flex;}
.section_7_1_tab li.active {color: darkcyan; font-weight: bold;}
.section_8_a1 {width: 1200px; height: 180px; margin: 0 auto; margin-top: 50px; display: flex; gap: 20px;}
.section_8_a1 {border-radius: 0.75rem; overflow: hidden;}
.section_8_a2 {border-radius: 0.75rem; overflow: hidden;}
```

### **JavaScript**

```
var swiper = new Swiper(".mySwiper_5", {
  slidesPerView: 6,
  spaceBetween: 30,
  navigation: {
      nextEl: ".swiper-button-next",
      prevEl: ".swiper-button-prev",
ar swiper = new Swiper(".mySwiper_6", {
  slidesPerView: 6,
  spaceBetween: 30,
  navigation: {
      nextEl: ".swiper-button-next",
      prevEl: ".swiper-button-prev",
ar swiper = new Swiper(".mySwiper_7", {
  slidesPerView: 6,
  spaceBetween: 30,
  navigation: {
      nextEl: ".swiper-button-next",
      prevEl: ".swiper-button-prev",
```

```
tabContent_1.forEach((tc,j) => {
          tc.style.display = (i == j) ? 'block' : 'none';
tabContent_1.forEach((tc, i) => {
   tc.style.display = (i === 0) ? 'block' : 'none';
// sectiopn7 open API 책 정보
async function fetchBooks4(query) {
   const REST_API_KEY = '7ec0ff3e208807bf25acfcb953642046'
    const params = new URLSearchParams({
      target: "title",
      query,
                                                              async function bookData4() {
       size: 50
                                                                  try {
                                                                      // 탭별 검색어 (국내도서, 외국도서, eBook)
   const url = `https://dapi.kakao.com/v3/search/book?${params}`;
                                                                      const queries = ["한국", "Harry", "망나니"];
   const response = await fetch(url, {
                                                                      const swipers = [".mySwiper_5", ".mySwiper_6", ".mySwiper_7"];
          Authorization: "KakaoAK 7b2300fc6315bb65035d1a3c7b49b161"
                                                                      for (let i = 0; i < queries.length; i++) {
                                                                      const data = await fetchBooks(queries[i]);
                                                                      const books = data.documents.filter(val => val.thumbnail && val.contents);
   if (!response.ok) {
      throw new Error(`HTTP 오류: ${response.status}`);
                                                                      const swiperContainer = document.querySelector(swipers[i]);
                                                                      const boxElements = swiperContainer.querySelectorAll(".swiper-slide");
   return response.json();
                                                                      boxElements.forEach((box, j) => {
                                                                          const doc = books[j];
                                                                          if (!doc) return;
                                                                          box innerHTML =
                                                                          <img src="${doc.thumbnail}" alt="${doc.title}" />
                                                                          <h3>${doc.title}</h3>
                                                                          ${doc.authors.join(", ")}
                                                                  } catch (error) {
                                                                      console.log('에러발생', error);
                                                                  bookData4();
```

const tabMenu\_1 = document.querySelectorAll('.section\_7\_1\_tab li');
const tabContent\_1 = document.querySelectorAll('.section\_7\_box2 .swiper')

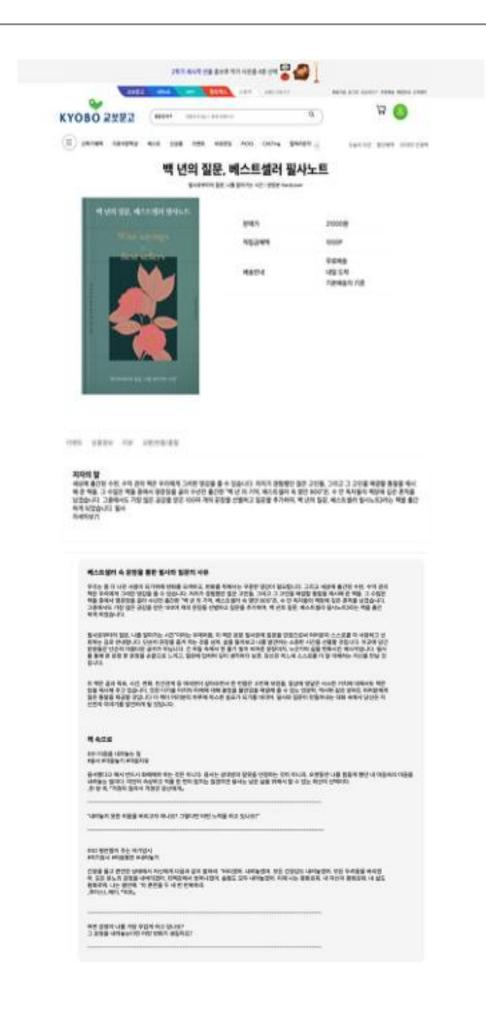
tabMenu\_1.forEach((tm, i) => {

tm.addEventListener('click', () => {
 tabMenu\_1.forEach(item => {

tm.classList.add('active');

item.classList.remove('active');

국내도서, 외국 도서, 이북 탭 중 하나를 선택하면, 해당 탭과 연동된 책 정보를 open API로 불러와 슬라이드 형식으로 페이지가 전환되도록 구현함.



# 식 서브 페이지 코드 구현



**JavaScript** 

```
border-bottom: 1px solid ■#eaeaea;
 span_list {width: 1200px; height: 40px; display: flex; gap: 30px; margin: 0 auto;
font—size: 20px; color: □#777; text—align: center; padding—left: 20px;}
   width: 1200px;
   margin: 0 auto;
 contextbox {
   margin: 0 auto;
   margin-top: 10px;
   border: 1px solid ■#eaeaea;
 * .hr_1 {width: 1000px; margin: 0 auto;} */
#tmpBox {
   width: 1200px;
   height: 1390px;
   margin: 0 auto;
   /* border-radius: 7px; */
   padding: 30px;
   margin-top: 30px;
   font-size: 20px;
   background-color: ■whitesmoke;
   border: 1px solid ■#eaeaea;
   border-radius: 20px;
   padding-bottom: 30px;
```

```
async function bookData() {
   const REST_API_KEY = '7b2300fc6315bb65035d1a3c7b49b161';
    const params = new URLSearchParams({
       target: "title",
    const url = `https://dapi.kakao.com/v3/search/book?${params}`;
      const response = await fetch(url, {
           headers: {
               Authorization: `KakaoAK ${REST_API_KEY}`
        if (!response.ok) {
           throw new Error(`HTTP 오류! 상태 코드: ${response.status}`)
       const data = await response.json();
       const subBox = document.querySelector(".sub_box");
       const contextBox = document.querySelector(".contextbox");
       const priceNum = document.querySelector(".pricenum");
       const book = data.documents[0];
       const { title, thumbnail, authors, price, contents } = book;
       priceNum.textContent += price + "원";
       contextBox.innerHTML = '
       <h3>저자의 말</h3>
       <h6>${authors && authors.length > 0 ? authors[0] : ""}</h6>
        <span>자세히보기</span>
    } catch (error) {
       console.log('에러발생', error);
```

```
//메모장으로 sub 텍스트 가져오기, 서버에 올려야 보임

document.addEventListener("DOMContentLoaded", async function () {

    try {

        const response = await fetch("./subpagetxt.txt");
        if (!response.ok) {

            throw new Error("Network response was not ok");
        }
        const data = await response.text();
        document.getElementById("tmpBox").innerHTML = data;

} catch (error) {

        console.error("There was a problem with the fetch operation:",
        error);
    }
});
```

메인 페이지와 연결된 서브 페이지에서는 책의 상세 정보 화면을 구현하였으며, 저자의 말은 open API로 불러오고, contextbox는 JavaScript를 통해 txt 파일과 연동하여 구현함.

# Q 소김

0 0 0

첫 번째 프로젝트를 통해 수업에서 배운 HTML, CSS, JavaScript를 실제로 활용하는 방법을 익힐 수 있었습니다. 교보문고 사이트를 구현하면서 웹 구조와 동작 원리를 보다 깊이 이해할 수 있어 매우 의미 있는 경험이었습니다. 다만, 배운 내용을 활용해 구현할 수 있는 범위가 제한적이었고, 코드와 기능을 이해하고 사용하는 데 시간이 오래 걸려 아쉬움이 남았습니다. 다음 프로젝트에서는 실력을 더욱 쌓아 보다 완성도 높은 결과물을 만들고 싶습니다.