








PDF 플립북 뷰어 (PDF Flipbook Viewer)

프로젝트 개요

이것은 현대적이고 반응형인 PDF 플립북 뷰어입니다. 데스크톱과 모바일 환경에서 최적화된 사용 경험을 제공하며, 아름다운 페이지 넘김 효과와 직관적인 네비게이션을 특징으로 합니다.

주요 특징:

-  완전한 모바일/데스크톱 반응형 디자인
-  부드러운 플립북 페이지 넘김 효과
-  키보드 단축키 지원
-  터치 제스처 및 스와이프 지원
-  직관적인 UI/UX
-  빠른 PDF 렌더링
-  다크 테마 기반 인터페이스

주요 기능

1. PDF 로딩 및 렌더링

- LocalStorage에 저장된 PDF URL에서 자동으로 PDF 파일 로드
- PDF.js를 사용한 고품질 렌더링
- 페이지별로 Canvas에 렌더링 후 이미지로 변환

2. 지능형 방향 감지

- 세로(Portrait) 모드: 데스크톱에서 2페이지 뷰 지원
- 가로(Landscape) 모드: 데스크톱에서 1페이지 뷰 제공
- 모바일: 항상 1페이지 뷰 (전체 화면)

3. 네비게이션 옵션

네비게이션 방법	설명

마우스 클릭	왼쪽 영역 클릭 (이전), 오른쪽 영역 클릭 (다음)
터치 제스처	모바일에서 스와이프 (좌/우)
키보드	← → (페이지 이동), Home (첫 페이지), End (마지막 페이지)
버튼	⏮ (첫 페이지), ⏭ (마지막 페이지)
설정 버튼	⚙ (설정 페이지로 이동)

4. 반응형 레이아웃

- 데스크톱: 고정 크기 플립북 + 부동 네비게이션 버튼
- 모바일: 전체 화면 뷰 + 터치 최적화
- 자동 리사이징: 화면 크기 변경 시 자동 조정

5. 사용자 인터페이스

- 로딩 화면 (애니메이션 포함)
- 현재 페이지 정보 표시
- 시각적 피드백 (버튼 호버 효과)
- 부드러운 애니메이션 전환

기술 스택

Frontend Framework: Vanilla JavaScript (ES6+)
PDF Processing: PDF.js (v3.11.174)
Flipbook Effect: Turn.js (v3)
DOM Manipulation: jQuery (v3.6.0)
UI/UX: CSS3 (Grid, Flexbox, Animations)
Font: Sora (Google Fonts)

파일 구조

```
project/  
├─ flipPDF_v2.html           # 메인 뷰어 페이지 (이 파일)
```

└─ setting.html	# 설정 페이지 (PDF URL 입력)
└─ readme.md	# 설명 문서 (이 파일)

설치 및 사용 방법

1. 기본 설정

```
# 로컬 서버에서 실행 (필수)
python -m http.server 8000
# 또는
python3 -m http.server 8000
```

2. PDF 설정

1. setting.html에 접속
2. PDF 파일의 공개 공유 URL 입력
 - Google Drive 공유 링크:
`https://drive.google.com/uc?export=download&id=FILE_ID`
 - 웹 서버 URL 등
3. URL은 자동으로 localStorage에 저장됨

3. 뷰어 사용

1. flipPDF_v2.html로 이동
2. PDF가 자동으로 로드됨
3. 각 네비게이션 방법으로 페이지 이동

주요 함수 및 기능 설명

초기화 함수

detectDevice()

모바일/데스크톱 기기 자동 감지

```
// 반환값: isMobileDevice (boolean)
```

loadPDFFromStorage()

LocalStorage에서 PDF URL 로드 및 렌더링

- URL이 없으면 `setting.html`로 자동 리다이렉트
 - 에러 발생 시 설정 페이지로 이동
-

PDF 처리 함수

`analyzePDF()`

PDF의 페이지 방향 자동 분석

- 가로 페이지 비율 계산
- `pdfOrientation` 설정 (`portrait/landscape`)

`renderFlipbook()`

모든 PDF 페이지를 Canvas에 렌더링

- 각 페이지를 고해상도 이미지로 변환
 - DOM에 `<div class="page">` 요소 생성
-

플립북 초기화

`initFlipbook()`

Turn.js 플립북 효과 초기화

```
// 설정 옵션:  
- width, height: 최적 크기 자동 계산  
- display: single (모바일) / double (데스크톱)  
- duration: 애니메이션 속도 (모바일 500ms, 데스크톱 1000ms)  
- elevation: 3D 그림자 효과
```

`getOptimalSize()`

화면 크기와 PDF 방향에 따른 최적 크기 계산

```
// 반환값:  
{  
  width: number,      // 플립북 너비  
  height: number,     // 플립북 높이  
  display: string     // "single" 또는 "double"  
}
```

네비게이션 함수

setupTouchZones ()

터치 이벤트 및 스와이프 지스처 설정

- 좌측 30% 영역: 이전 페이지
- 우측 30% 영역: 다음 페이지
- 스와이프 임계값: 50px

updatePage ()

현재 페이지 정보 업데이트

- 페이지 표시 (예: "3 / 50")
- 네비게이션 버튼 활성화 상태 업데이트

updateNavButtons ()

네비게이션 버튼 비활성화 상태 관리

- 첫 페이지에서 ◀◀ 버튼 비활성화
- 마지막 페이지에서 ▶▶ 버튼 비활성화

반응형 처리

handleResize ()

화면 크기 변경 시 플립북 재조정

- 디스플레이 모드 변경 감지
- 필요시 플립북 재초기화

유틸리티 함수

debounce (func, delay)

resize 이벤트 성능 최적화

- 연속 이벤트를 단일 호출로 통합

- 기본 지연: 300ms

goToSettings ()

설정 페이지로 이동

```
window.location.href = 'setting.html';
```

키보드 단축키

키	기능
←	이전 페이지
→	다음 페이지
Home	첫 페이지로 이동
End	마지막 페이지로 이동

반응형 디자인 상세

모바일 (≤768px)

- 디스플레이: 전체 화면 1페이지 뷰
- 네비게이션: 터치 제스처 (스와이프)
- 버튼 크기: 44×44px
- 페이지 정보: 하단 중앙
- 로딩 아이콘: 4em

데스크톱 (>768px)

- 세로 PDF: 2페이지 뷰 (대면도 레이아웃)
- 가로 PDF: 1페이지 뷰
- 네비게이션: 부동 버튼 (좌측/우측)
- 버튼 크기: 48×48px

- 플립북 중앙: 화면 중앙에 정렬



색상 및 스타일

요소	색상	설명
배경	#000000	순수 검정
네비게이션 버튼	#7c3aed → #6d28d9	자주색 그래디언트
텍스트	#FFFFFF 또는 #1a1a1a	흰색/검정 대비
로딩 화면 배경	#1a1a1a → #2a2a2a	다크 그래디언트



커스터마이제이션 옵션

플립북 속도 조정

```
// initFlipbook() 내 duration 값 변경
duration: isMobileDevice ? 500 : 1000 // ms 단위
```

터치 스와이프 감도 조정

```
// setupTouchZones() 내 임계값 변경
if (diff > 50) { // 50을 다른 값으로 변경
  $('#flipbook').turn('next');
}
```

3D 그림자 효과 강도

```
// initFlipbook() 내 elevation 값 변경
elevation: isMobileDevice ? 20 : 50 // 숫자가 클수록 강함
```

색상 테마 변경

CSS의 `--color` 또는 직접 색상 코드 수정:

```
.nav-btn {  
  background: linear-gradient(135deg, #7c3aed 0%, #6d28d9 100%);  
  /* 위 색상을 원하는 색상으로 변경 */  
}
```



주의사항




CORS 정책

- PDF URL은 CORS 정책을 허용하는 서버에서 제공되어야 함
- Google Drive 공유 링크 사용 권장

로컬 파일 보안

- `file://` 프로토콜로 실행 불가
- 반드시 로컬 웹 서버 필요

브라우저 호환성

- Chrome/Edge:  완벽 지원
- Firefox:  완벽 지원
- Safari:  일부 제한 (모바일 방향 감지)



일반적인 문제 해결

"PDF 로드 실패" 오류

- 원인: CORS 오류 또는 잘못된 URL
- 해결:
 - Google Drive: `https://drive.google.com/uc?export=download&id=FILE_ID` 형식 사용
 - 다른 서버: CORS 헤더 확인

페이지가 회전하지 않음

- 원인: 모바일에서 화면 회전 비활성화

- 해결: HTML meta 태그 확인
- `<meta name="viewport" content="...viewport-fit=cover, maximum-scale=1.0, user-scalable=no">`

버튼이 반응하지 않음

- 원인: jQuery 또는 Turn.js 라이브러리 로드 실패
- 해결: 개발자 도구 콘솔에서 오류 확인, CDN 연결 상태 확인

지원 정보

제작: Perplexity AI

웹사이트: <https://welfareact.net>

라이선스: MIT (자유로운 사용 가능)

버전 히스토리

버전	날짜	변경 사항
v1.0	2025-11-02	초기 릴리스

감사의 말

이 프로젝트는 다음의 오픈소스 라이브러리를 사용합니다:

- [PDF.js](#) - Mozilla의 PDF 렌더링 라이브러리
- [Turn.js](#) - 플립북 효과 라이브러리
- [jQuery](#) - DOM 조작 라이브러리
- [Sora Font](#) - Google Fonts

행복한 PDF 읽기되세요! 