POWER APPS × AI 코딩 · 실습 교재

VS Code + Claude로 Power Apps 앱 단숨에 만들기

VS Code의 Claude에게 앱을 설명하고 SharePoint 목록 컬럼만 정확히 알려 주면, 가져오기 한 번으로 데이터까지 연결된 앱이 열립니다.

윤재웅 · Power Platform & AI 업무 자동화

왜 코드로 만드나

드래그 앤 드롭의 한계를 AI가 넘어 줍니다

01

반복 작업

화면마다 컨트롤을 끌어다 놓고 위치·색·글꼴을 하나씩 맞추는 데 대부분의 시간이 듭니다.

02

앱의 실체는 텍스트

Studio에서 만든 앱은 화면별 .pa.yaml 파일과 Power Fx 수식으로 저장됩니다. 사람이 읽을 수 있는 코드입니다.

03

AI가 쓰고 검증

Claude가 요구사항을 듣고 화면·컨트롤·수식을 한꺼번에 작성하고, 오류는 스스로 고칩니다.

핵심 · 사람은 "무엇을 만들지"를 설명하고 결과를 검토합니다. 반복 배치는 AI가 맡습니다.

파일 구조

.msapp과 앱 패키지(.zip)

```
MyApp.msapp  (zip 형식)
└ Src\
   ├ App.pa.yaml            ← 앱 설정
   ├ HomeScreen.pa.yaml     ← 화면마다 1개
   ├ DetailScreen.pa.yaml
   └ Component\             ← 컴포넌트
```

내보내기 · Studio에서 저장 ▸ **복사본 다운로드**

가져오기 · make.powerapps.com ▸ 앱 ▸ **앱 가져오기** ▸ **파일에서 (.msapp)**, 또는 Studio에서 파일 ▸ 열기 ▸ 찾아보기

앱 패키지(.zip) · 앱과 연결 정보를 함께 담은 묶음입니다. 앱 ▸ 가져오기 ▸ **캔버스 앱 가져오기**로 올리면, 가져오는 단계에서 SharePoint 같은 연결을 내 계정 것으로 고를 수 있습니다.

참고 · 소스 폴더를 .msapp으로 묶는 `pac canvas pack`은 Microsoft가 사용 중단을 예고했습니다.

두 가지 경로

이 교재는 A 방식으로 실습합니다

방식	A. 앱 패키지(.zip) 가져오기 · 실무 방식	B. 공식 Canvas Apps 플러그인 + MCP
흐름	VS Code의 Claude가 앱 패키지 생성 → make.powerapps.com에서 가져오기	Studio 공동 작성 세션에 연결해 실시간 동기화
준비	VS Code + Claude, SharePoint 목록	.NET SDK 10, 플러그인 설치, 저장된 빈 앱
데이터 연결	가져오기 단계에서 SharePoint 연결 선택	Studio 데이터 패널에서 추가
장점	설치가 거의 없고 결과를 파일로 주고받기 쉬움	검증 자동화, 기존 앱을 말로 계속 수정
추천 대상	시민 개발자 입문 · 사내 교육	기존 앱 고도화 · 개발 조직

두 방식 모두 결과물은 같은 캔버스 앱 코드입니다. A로 빠르게 만들고, 운영하며 고칠 때 B를 더하면 됩니다.

핵심 · 데이터 먼저

SharePoint 컬럼만 정확히 알려 주면 거의 수정이 필요 없습니다

표시 이름	형식	예시 값	앱에서 쓰임
제목(Title)	한 줄 텍스트	A4 복사용지	목록·검색
품목코드	한 줄 텍스트	P-1001	상세 화면
수량	숫자	120	재고 경고(10 미만 빨간색)
창고	선택(서울·부산·광양)	부산	드롭다운 필터
입고일	날짜	2026-11-07	정렬
담당자	사람	홍길동	담당자 표시

예시 · 사이트 https://회사.sharepoint.com/sites/물류팀 의 목록 「재고」

알려 줄 것 · 사이트 주소, 목록 이름, 컬럼별 이름·형식·선택 값

내부 이름 주의 · 한글로 만든 컬럼은 SharePoint 안에서 다른 내부 이름(예: _x...)으로 저장될 수 있습니다. 컬럼을 영문으로 먼저 만들고 표시 이름만 한글로 바꾸거나, 목록 설정에서 확인한 내부 이름을 함께 알려 주면 수식 오류가 줄어듭니다.

형식이 중요 · 선택·사람·날짜 형식은 Power Fx 수식이 달라지므로 반드시 형식까지 말해 줍니다.

실습 · 따라 하기

설명 → 패키지 → 가져오기 → 연결, 5단계

1. **SharePoint 목록 준비** · 앱에서 쓸 컬럼을 만들고 샘플 데이터 몇 건을 넣습니다.
2. **VS Code의 Claude에게 설명** · 어떤 앱인지, 화면 구성, 목록 주소와 컬럼 명세를 알려 주고 Power Apps 캔버스 앱 패키지(.zip)로 만들어 달라고 합니다.
3. **가져오기** · make.powerapps.com ▸ 앱 ▸ 가져오기 ▸ **캔버스 앱 가져오기** ▸ 업로드에서 zip을 선택합니다.
4. **가져오기 설정** · 앱은 **새로 만들기**로, 연결 항목에서는 내 **SharePoint 연결**을 선택한 뒤 가져오기를 누릅니다.
5. **열어서 확인** · 앱을 열면 목록 데이터가 바로 보입니다. 바꿀 점은 다시 Claude에게 말로 요청해 새 패키지를 받습니다.

팁 · 처음에는 화면 2~3개, 컬럼 5~6개로 작게 시작합니다. 앱이 열리면 기능을 하나씩 더해 갑니다.

프롬프트 예시

앱 설명 + 컬럼 명세를 한 번에 전달합니다

Power Apps 캔버스 앱을 만들어 줘. 결과는 make.powerapps.com에서 가져올 수 있는 **앱 패키지(.zip)**로.

[앱] 물류팀 재고 관리 · 태블릿 가로 화면 · 남색·주황색 기업형 테마
[화면] ① 목록: 제목 검색, 창고 드롭다운 필터, 수량 10 미만은 빨간색 ② 상세: 선택 품목 전체 정보 ③ 입력 폼: 새 품목 등록·수정 후 목록으로 이동
[데이터] SharePoint 사이트 https://회사.sharepoint.com/sites/물류팀 , 목록 「재고」
[컬럼] Title(한 줄 텍스트, 표시명 제목) · ItemCode(한 줄 텍스트, 품목코드) · Qty(숫자, 수량) · Warehouse(선택: 서울/부산/광양, 창고) · InDate(날짜, 입고일) · Owner(사람, 담당자)

괄호 안처럼 **내부 이름 · 형식 · 표시 이름**을 함께 적으면 수식이 처음부터 맞게 만들어집니다.

앱 속의 코드

.pa.yaml과 Power Fx 읽기

```
Screens:
  HomeScreen:
    Properties:
      Fill: =RGBA(247, 246, 242, 1)
    Children:
      - TitleLabel:
          Control: Label
          Properties:
            Text: ="재고 현황"
            Size: =28
      - CountLabel:
          Control: Label
          Properties:
            Text: ="품목 " & CountRows(재고) & "개"
```

Claude가 만든 패키지 안의 앱은 이런 코드로 되어 있습니다. **Screens** 아래에 화면, **Children** 아래에 컨트롤이 들어갑니다.

Control은 컨트롤 종류, **Properties**는 속성입니다.

=로 시작하는 값은 Power Fx 수식입니다. Excel 수식처럼 데이터 개수를 세고, 화면을 이동(Navigate)하고, 데이터를 저장(Patch)합니다.

Studio의 컨트롤 ▸ **코드 보기**로 지금 만든 앱의 YAML을 확인하고 복사·붙여넣기 할 수도 있습니다.

심화 · B 방식

공식 Canvas Apps 플러그인으로 실시간 동기화

준비: .NET SDK 10 이상, 저장된 빈 캔버스 앱, Studio 설정 ▸ 업데이트 ▸ 공동 작성 켜기

Claude Code

```
/plugin marketplace add microsoft/power-platform-skills
/plugin install canvas-apps@power-platform-skills
/configure-canvas-mcp
```

VS Code

확장(Extensions)에서 `@agentPlugins canvas apps` 검색 ▸ Microsoft의 **Canvas Apps** 설치 ▸ Chat을 Agent 모드로 열기

1. Studio에서 앱을 열어 둔 채 브라우저 주소(URL)를 복사합니다. 탭은 끝까지 닫지 않습니다.
2. **/configure-canvas-mcp**를 실행하고 URL을 붙여 넣은 뒤 로그인합니다. 환경·앱 ID는 자동으로 읽습니다.
3. 연결 확인: **“List the available Canvas App controls”**라고 물어 컨트롤 목록이 나오면 성공입니다.

잘 쓰는 법과 보안

작게 시작하고, 사람이 검토합니다

잘 쓰기

작게 → 크게

기본 화면부터 만들고 기능을 하나씩 더합니다. 큰 변경 후에는 바로 미리 보기로 테스트합니다. 레이아웃·색상 방향을 구체적으로 말할수록 결과가 좋습니다.

검증

생성 코드는 내 책임

AI는 최선을 다해 작성하지만, 회사 기준·접근성·보안 요건 충족 여부는 사람이 확인합니다. 개발 환경에서 먼저 검증한 뒤 운영 앱에 적용합니다.

보안

권한과 정책

신뢰할 수 있는 마켓플레이스에서만 설치하고, 필요한 권한만 가진 계정으로 연결합니다. 프롬프트에 비밀번호·토큰을 넣지 않고, 회사의 DLP(데이터 손실 방지) 정책을 따릅니다.

정리

설명은 사람이, 배치는 AI가,
검토는 다시 사람이.

데이터(SharePoint 컬럼)를 정확히 정의하는 것이 앱 품질을 결정합니다. 캔버스 앱이 코드로 저장되기 때문에 가능한 방식입니다. 시민 개발자는 업무 설명에 집중하고, 반복적인 화면 작업은 AI에게 맡깁니다.

출처 · Microsoft Learn: Create and edit canvas apps with AI code generation tools / View source code files for canvas apps / pac canvas (Power Platform CLI) / Export and import canvas apps as a package / Export and import a canvas app as a single app file · 실습 방식은 필자의 사내 교육 경험 기준