

1

파트 1 - Cursor AI 설치부터 첫 "Hello World" 앱 실행까지!

▼ 📖 1. Cursor AI 설치 및 계정 설정

🖥️ 1.1 Cursor AI 설치하기

- 공식 웹사이트(<https://www.cursor.com/>)에 접속
- [Download] 버튼 클릭
- 운영체제에 맞는 버전 선택 (Windows, macOS, Linux)
- 다운로드된 설치 파일 실행
- 설치 마법사의 안내에 따라 진행

🔑 1.2 계정 생성 및 로그인

- Cursor AI 첫 실행 시 회원가입(Sign up)/로그인(Sign in) 화면 표시
- [Sign up] 클릭하여 계정 생성
- 이름과 이메일 주소 입력 or 구글 계정으로 로그인 (Continue with Google)

🇸🇰 무료 플랜 vs 유료 플랜

- 무료 플랜: 기본 기능 사용 가능 (이 강의로 충분) = 한달에 50 AI Request
- 유료 플랜(Pro): 한달에 AI 채팅 요청을 500 Request 보낼 수 있음

▼ 🖥️ 2. Cursor AI 인터페이스 및 개발 환경 최적화

📱 2.1 인터페이스 구성 요소

- 파일 탐색기 (좌측): 프로젝트 파일 및 폴더 관리

- 코드 편집 영역 (중앙): 실제 코드 작성 및 편집 공간
- AI 어시스턴트 패널 (우측): AI와 대화하여 코드 생성 요청
- 터미널 및 콘솔 (하단): 명령어 실행 및 앱 테스트

⚙️ 2.2 개발 환경 최적화 설정

📖 유용한 단축키 (윈도우 Ctrl / 맥 Cmd)

- Ctrl+I: AI 어시스턴트에 요청 → AI 채팅창이 안 보일때
- Ctrl+L: 선택한 코드에 대한 AI 설명 요청 → 채팅창 위 부분 @Add Context에 내용이 복사
- Ctrl+K: 코드 편집기에서 AI에 직접 질문

▼ 📱 3. Flutter 개발 환경 설정

🕒 3.1 모바일 앱 개발의 역사와 Flutter의 등장

📖 Flutter 이전의 모바일 앱 개발 방식

- 네이티브 개발: iOS는 Swift/Objective-C, Android는 Java/Kotlin으로 각각 따로 개발
- 두 번의 개발 비용: 같은 앱을 iOS와 Android에서 각각 따로 만들어야 함
- 서로 다른 프로그래밍 언어와 개발 환경
- 유지보수의 어려움: 버그 수정이나 기능 추가 시 두 플랫폼 모두 수정 필요

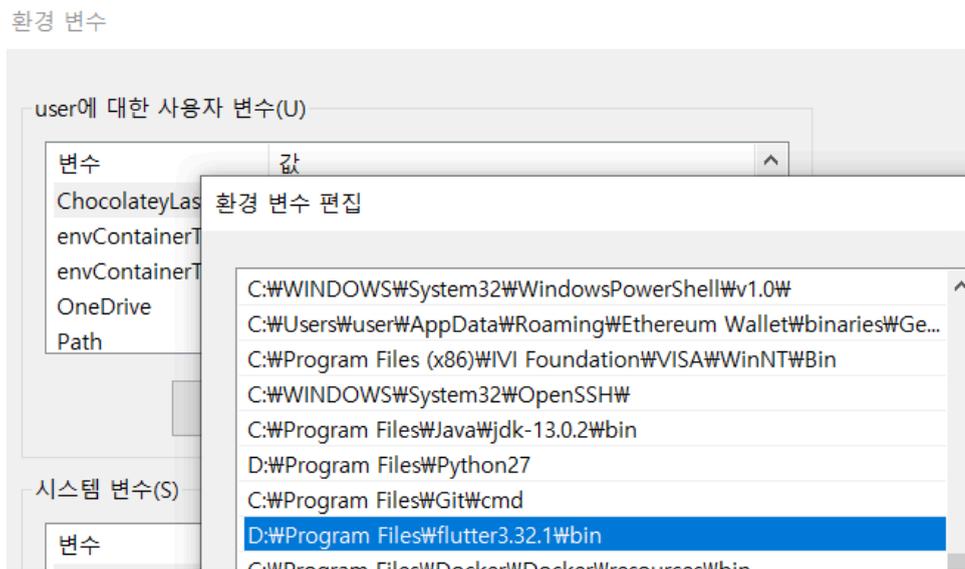
🚀 Flutter의 혁신적인 접근법

- 하나의 코드베이스로 여러 플랫폼 지원
- Dart 언어 사용: 배우기 쉬운 현대적인 프로그래밍 언어
- Hot Reload: 코드 변경 사항을 즉시 확인 가능
- 네이티브 성능: 네이티브 앱에 준하는 성능 제공

📱 3.2 Flutter SDK 설치

- flutter.dev 웹사이트 접속

- [Get started] 클릭
- 운영체제에 맞는 설치 가이드 선택
- ZIP 파일 다운로드 및 원하는 위치에 압축 해제
 - C:\flutter
- 환경 변수 설정 (PATH에 Flutter 경로 추가)
 - Windows에서는 시스템 속성 > 고급 시스템 설정 > 환경 변수로 이동해서 PATH에 C:\flutter\bin을 추가해주시면 됩니다.



- 설치 확인: 명령 프롬프트(터미널)에서 'flutter doctor' 명령어 실행 -> 아래와 같이 나오지 않으면, path문제나 체크 안된 부분 확인

```
PS D:\> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.32.1, on Microsoft Windows [Version 10.0.19045.5854], locale ko-KR)
[✓] Windows Version (10 Home 64비트, 22H2, 2009)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[✓] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.45)
    X The current Visual Studio installation is incomplete.
      Please use Visual Studio Installer to complete the installation or reinstall Visual Studio.
[✓] Android Studio (version 2024.3)
[✓] VS Code (version 1.57.1)
[✓] Connected device (3 available)
[✓] Network resources
```

- Cursor AI에서 플러그인 설치하기 flutter 설치

▼ 맥에서 설치와 확인

1단계: Flutter SDK 다운로드

프로세서 타입에 따라 선택:

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

Intel Processor

Apple Silicon

`flutter_macos_3.32.4-stable.zip`

`flutter_macos_arm64_3.32.4-stable.zip`

For other release channels, and older builds, check out the [SDK archive](#).

The Flutter SDK should download to the macOS default download directory: `~/Downloads/`.

2. Create a folder where you can install Flutter.

Consider creating a directory at `~/development/`.

3. Extract the file into the directory you want to store the Flutter SDK.

```
$ unzip ~/Downloads/flutter_macos_arm64_3.32.4-stable.zip \  
-d ~/development/
```

When finished, the Flutter SDK should be in the `~/development/flutter` directory.

- **Intel Mac:** `flutter_macos_3.32.4-stable.zip`
- **Apple Silicon (M1/M2/M3):** `flutter_macos_arm64_3.32.4-stable.zip`

프로세서 확인 방법:

```
# 터미널에서 확인  
uname -m  
# arm64 → Apple Silicon  
# x86_64 → Intel
```

2단계: 설치 폴더 생성

공식 문서에서 권장하는 위치: `~/development/`

```
# development 폴더 생성  
mkdir ~/development
```

3단계: Flutter SDK 압축 해제

방법 1: 터미널 사용 (공식 방법)

```
# Downloads 폴더에서 development 폴더로 직접 압축 해제
unzip ~/Downloads/flutter_macos_arm64_3.32.4-stable.zip -d ~/development/
```

방법 2: Finder 사용

1. Downloads 폴더에서 flutter zip 파일 더블클릭
2. 압축 해제된 flutter 폴더를 ~/development/ 로 이동

4단계: 결과 확인

압축 해제 완료 후 폴더 구조:

```
~/development/
├── flutter/
│   ├── bin/
│   ├── packages/
│   ├── examples/
│   └── ... (기타 파일들)
```

5단계: PATH 환경변수 설정

```
# .zshrc 파일 편집
nano ~/.zshrc
vi ~/.zshrc # nono 설치가 안되어 있는 경우

vi .zprofile

# 다음 줄 추가
export PATH="$PATH:$HOME/development/flutter/bin"

export PATH="$PATH:/Volumes/MacPro16/workspace/lib"

# 변경사항 적용
source ~/.zshrc
```

6단계: 설치 확인

```
# Flutter 버전 확인
flutter --version

# 시스템 환경 체크
flutter doctor
```

중요한 점:

- 공식 문서에서는 `~/development/` 경로를 권장합니다
- 이 위치는 시스템 권한 문제 없이 안전하게 사용할 수 있습니다
- 압축 해제 후 최종 경로는 `~/development/flutter/` 가 됩니다

이렇게 하면 Flutter 공식 문서의 권장 방식대로 설치가 완료됩니다!

▼ 📱 4. 첫 "Hello World" Flutter 앱 만들기

🌟 4.1 새 Flutter 프로젝트 생성

- Cursor AI에서 터미널 열기 (메뉴에서 Terminal → New Terminal)
- 명령어 입력: `flutter create hello_world_app``
- 생성된 프로젝트 폴더로 이동: `cd hello_world_app``
- Cursor AI에서 프로젝트 열기: File > Open Folder

📝 4.2 기본 코드 구조 이해

- `lib/main.dart`: 앱의 메인 코드 파일
- `pubspec.yaml`: 의존성 및 프로젝트 설정 파일
- `android/ios/web/windows` 폴더: 각 플랫폼별 네이티브 코드

🌟 4.3 Cursor AI로 코드 수정하기

🤖 AI 어시스턴트에 요청할 내용:

- "lib/main.dart 파일을 수정해서 간단한 Hello World 앱을 만들어주세요"

- "화면 중앙에 'Hello World!' 텍스트를 표시하고 싶어요"
- "배경색을 파란색으로 변경해주세요"
- **처음부터 커서 AI 채팅창에 "Flutter로 Hello World 새로운 앱을 만들어주세요" 이
렇게 입력**

4.4 앱 실행 및 테스트

- **실행: 터미널에서 `flutter run`**
 - 위 명령어 에러가 나오면 아래 명령어를 복사해서 실행하기

```
flutter run
```

```
-- flutter run 에서 에러가 발생하면 아래 명령어로
```

```
flutter run -d chrome --web-hostname=127.0.0.1
```

- **Hot Reload 테스트: 코드 수정 후 'r' 키 눌러 즉시 반영 확인**

▼ 5. 주요 학습 내용 요약

- Cursor AI 설치 및 계정 설정 완료
- Flutter SDK 설치 및 개발 환경 구축
- 첫 번째 Flutter 앱 생성 및 실행
- AI 어시스턴트를 활용한 코드 작성과 수정 경험
- Hot Reload 기능을 통한 빠른 개발 사이클 체험

다음 강의 예고

다음 강의에서는 더 복잡한 TO-DO 앱을 만들면서 Cursor AI의 고급 기능들을 활용해 보겠습니다. 사용자 인터페이스 디자인부터 데이터 저장까지, 실제 사용 가능한 앱을 만들어보세요!

▼ [팁] Mac에서 터미널 명령어

Mac에서 터미널을 처음 사용할 때 자주 쓰는 **기본 명령어**들을 정리해드릴게요. 이 리스트는 **초보자 기준**으로, 실제 터미널 작업 시 자주 사용되는 명령어 위주입니다.

파일 및 디렉토리 관련

명령어	설명
<code>ls</code>	현재 디렉토리의 파일 목록 보기
<code>ls -al</code>	숨김 파일 포함 상세 목록 보기
<code>cd [폴더명]</code>	해당 디렉토리로 이동
<code>cd ..</code>	상위 디렉토리로 이동
<code>cd</code>	홈 디렉토리로 이동
<code>pwd</code>	현재 디렉토리 경로 출력
<code>mkdir [폴더명]</code>	새 폴더 생성
<code>touch [파일명]</code>	새 파일 생성
<code>rm [파일명]</code>	파일 삭제
<code>rm -r [폴더명]</code>	폴더 및 내부 파일 삭제
<code>cp [원본] [대상]</code>	파일 또는 폴더 복사
<code>mv [원본] [대상]</code>	파일 또는 폴더 이동/이름 변경

시스템 및 환경 확인

명령어	설명
<code>echo \$PATH</code>	환경변수 PATH 확인
<code>which [명령어]</code>	명령어 위치 확인 (<code>which python</code> 등)
<code>top</code>	실시간 시스템 사용량 확인
<code>whoami</code>	현재 사용자 이름 확인
<code>uname -a</code>	시스템 정보 확인
<code>df -h</code>	디스크 사용량 확인
<code>du -sh [폴더명]</code>	해당 폴더의 용량 확인

실행 및 설정

명령어	설명
<code>open .</code>	현재 폴더를 Finder에서 열기

<code>open [파일명]</code>	파일을 기본 앱으로 열기
<code>code .</code>	VSCode로 현재 폴더 열기 (VSCode 설치 시)
<code>source ~/.zshrc</code>	Zsh 설정 다시 불러오기
<code>export PATH=...</code>	환경 변수 설정 (일시적)

💡 참고

- macOS 기본 셸은 **zsh** (Catalina 이후 기준)입니다.
- 설정 파일은 보통 `~/.zshrc` 또는 `~/.bash_profile` 에 작성합니다.
- `man [명령어]` 를 입력하면 해당 명령어의 설명서(manual)를 볼 수 있습니다.

🎯 강의 미션

- Cursor AI 설치부터 계정 설정까지 완벽히 마스터하기
- Flutter 개발 환경을 첫 단계부터 직접 설정해보기
- 첫 번째 Flutter 앱 프로젝트를 생성하고 실행하기
- AI 도구와 전통적인 개발 환경의 차이점 체험하기

📝 수업노트

🔧 설치 단계별 가이드

- Cursor AI 다운로드 및 설치 (cursor.com 사이트에서 운영체제별 설치 프로그램 다운로드)
- OpenAI API 키 연동 및 계정 설정
- Flutter SDK 설치 (flutter.dev에서 공식 설치 가이드 확인)
- 개발 환경 확인: flutter doctor 명령어로 설치 상태 체크

🌐 주요 기능 마스터하기

- Cursor AI의 메인 기능: Ctrl+K (코드 생성), Ctrl+L (채팅), Ctrl+I (코드 편집)
- 파일 명, 폴더 구조 설정 및 프로젝트 시작 방법
- Hello World 앱 생성 명령어: flutter create hello_world_app
- 앱 실행 명령어: flutter run (또는 F5 키 사용)

🔍 트러블슈팅 가이드

- 사용자 종속성 문제: flutter doctor --android-licenses 명령으로 해결
- 에뮬레이터/시뮬레이터 연결 문제: Android Studio/Xcode 설치 및 디바이스 연결
- Hot Reload 기능 사용법: 코드 수정 후 r 키 또는 저장으로 실시간 반영

🌟 첫 성공 단계

- 앱이 에뮬레이터나 실제 디바이스에서 정상 실행되는 것을 확인
- Cursor AI의 자동 완성 및 코드 생성 기능 체험
- 기본 Flutter 앱의 구조와 주요 파일들 이해 (main.dart, pubspec.yaml 등)