



# 웹 파일 업로드와 다운로드: 기본과 보안



# 웹 파일 업로드와 다운로드: 기본과 보안

파일 업로드: 내 컴퓨터 → 서버

파일 다운로드: 서버 → 내 컴퓨터

웹 개발의 필수 기능, 보안 취약점의 주요 원인

# 파일 전송의 기본 개념

## 업로드 과정

로컬 저장소에서 파일 선택

서버로 데이터 전송

서버측 저장 및 처리

## 다운로드 과정

서버에서 파일 요청

브라우저로 데이터 전송

로컬 저장소에 저장

A photograph of a workspace featuring a laptop and a monitor. The laptop screen shows a code editor with a dark theme and a file explorer on the left. The monitor displays a web browser with a code editor interface. The background is a plain wall, and a small potted plant is visible on the left.

# 개발 환경 구축

## Flask 설치

웹 서버 구축 기반

## 모듈 설치

`pip install flask openpyxl`

## 샘플 데이터 준비

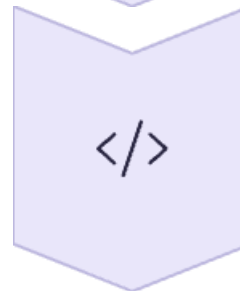
테스트용 파일 생성

# 엑셀 파일 다운로드 구현



서버에서 엑셀 생성

openpyxl 라이브러리 활용



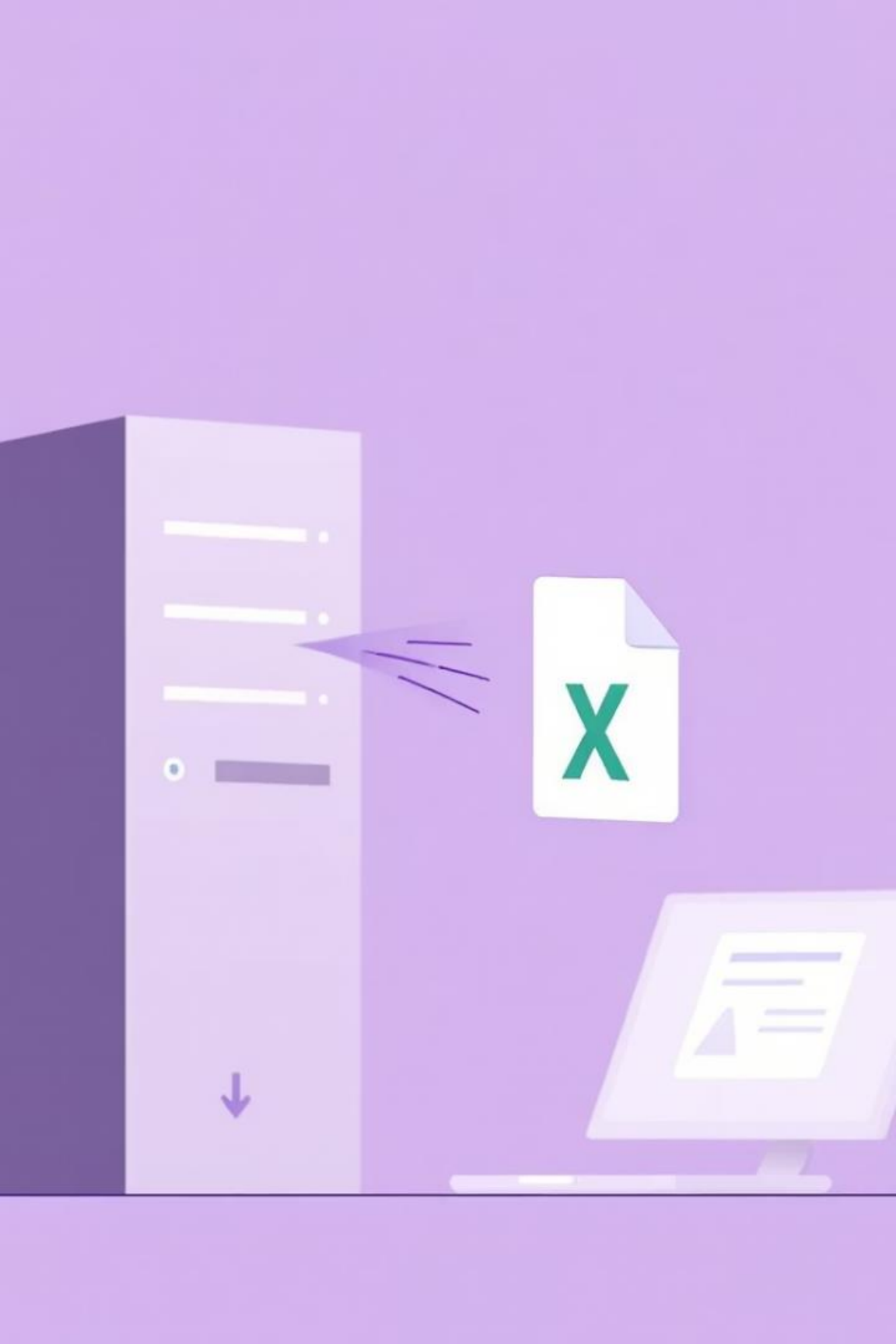
헤더 설정

Content-Disposition: attachment



클라이언트 전송

파일 스트림 응답





# 다운로드 보안 위험

## 악성코드 감염

신뢰할 수 없는 출처의 파일

## 무결성 손상

전송 중 데이터 변조 가능성

## 링크 위조

악의적 다운로드 유도



# 파일 업로드 구현

클라이언트 UI

<input type="file">

유효성 검증

MIME 타입 확인



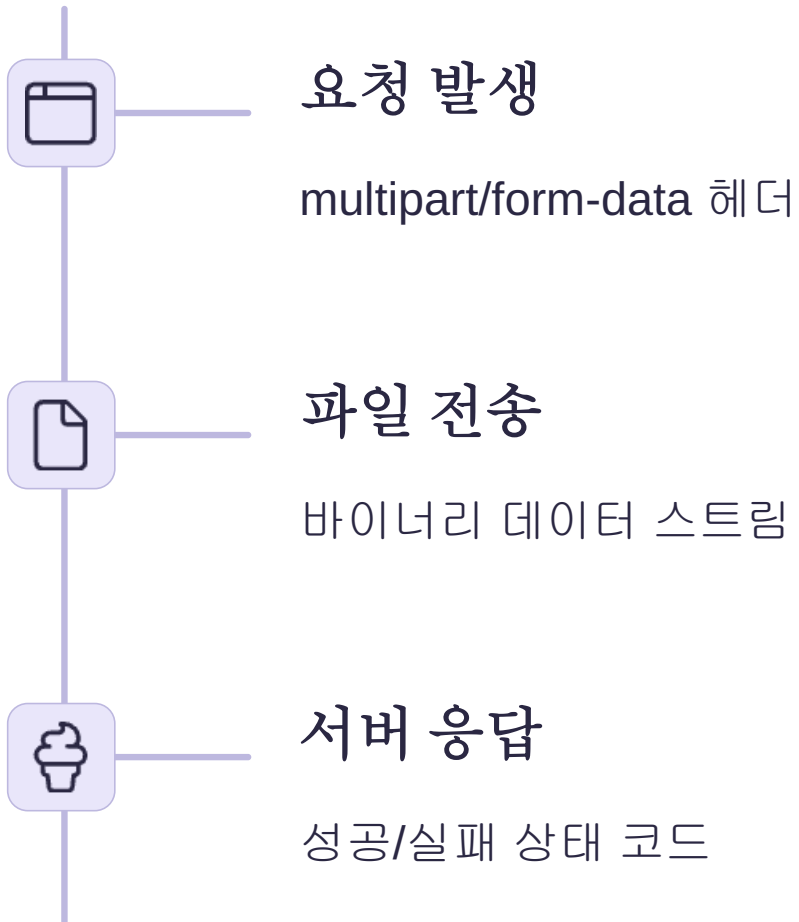
서버 수신

request.files 처리

파일 저장

안전한 경로에 저장

# 업로드 네트워크 분석





# 업로드 보안 최적화

