6차시. 전 세계에 내 사이트 공개하기: GitHub Pages 배포와 배포 갱신

Day 1 · 15:00–15:45 (수업 45분) / 15:45–16:00 (휴식 15분)
저장소: `portfolio-web`
선수 학습: 5차시 GitHub 원격 연동 및 첫 푸시 완료

이 차시에서 하는 일

- GitHub Pages 호스팅 서비스를 활성화하여 내 포트폴리오 웹사이트를 무료 공개 주소로 배포한다.
- 전 세계 누구나 접속할 수 있는 라이브 사이트 URL(`https://계정이름.github.io/portfolio-web/`)을 확인한다.
- OpenCode로 웹페이지 하단 푸터(연락처 및 저작권)를 추가하고, 변경 사항을 push하여 배포 사이트가 자동으로 갱신되는 파이프라인을 체득한다.
- 디자인 보조 도구인 눈누(무료 폰트)와 Coolors(색상 조합)를 소개하고 폰트 라이선스 확인 원칙을 배운다.

정적 웹 배포와 실시간 공개의 목적

내가 내 컴퓨터에서 개발한 웹페이지는 주소창에 `file:///` 또는 `http://localhost:5500`으로 뜹니다. 이는 오직 내 컴퓨터 안에서만 볼 수 있는 로컬 주소이므로, 친구나 채용 담당자에게 링크를 보내주어도 상대방 컴퓨터에서는 열리지 않습니다.

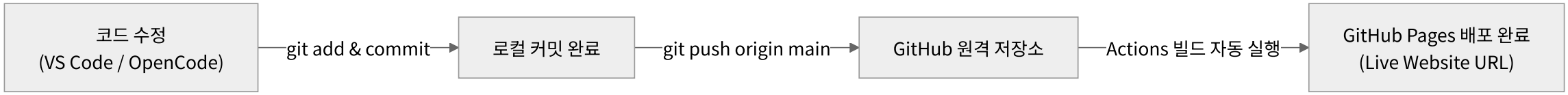
내 웹사이트를 전 세계에 공개하려면 인터넷 서버 컴퓨터를 빌리고 도메인을 연결해야 하지만, HTML·CSS·이미지로 구성된 정적 웹(Static Web)은 GitHub가 제공하는 **GitHub Pages** 서비스를 이용하면 무료로 즉시 공개할 수 있습니다.

또한 Git으로 코드를 수정하고 원격 저장소에 push하기만 하면, GitHub가 변경된 코드를 자동으로 감지하여 실제 운영 웹사이트를 최신 내용으로 알아서 갈아 끼워주는 **지속적 배포(Continuous Deployment)**의 기초를 경험할 수 있습니다.

GitHub Pages 호스팅 원리와 자동 재배포 파이프라인

GitHub Pages는 다음과 같은 자동화 흐름으로 동작합니다:

GitHub Pages 호스팅 원리와 자동 재배포 파이프라인 · 계속



1. **호스팅 진입점 (`index.html`)**
   - GitHub Pages는 지정된 브랜치(`main`)의 루트 폴더에 위치한 `index.html` 파일을 첫 대문 페이지로 인식하여 웹 브라우저에 서비스합니다. 파일명이 소문자 `index.html`이어야 올바르게 동작합니다.
2. **배포 주소 체계**
   - 기본적으로 `https://<GitHub계정명>.github.io/<저장소이름>/` 형식의 고유 주소가 생성됩니다.
   - 본 실습의 주소는 `https://<GitHub계정명>.github.io/portfolio-web/` 형태가 됩니다.
3. **자동 재배포 주기**
   - 로컬에서 코드를 고친 뒤 push하면, GitHub 내부의 배포 자동화 로봇(GitHub Actions)이 동작하여 약 1~2분 안에 실제 사이트에 반영합니다.

선택 보조 도구: 눈누와 Coolors

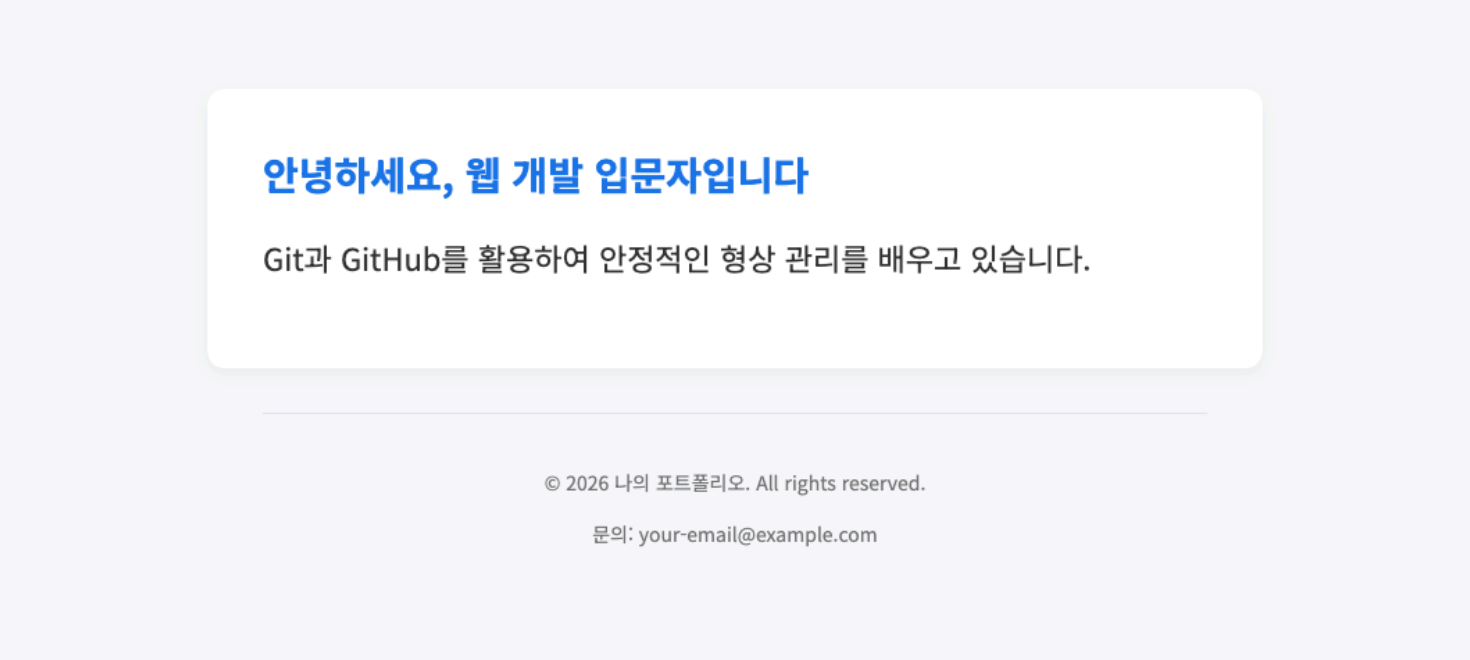
포트폴리오 웹페이지를 가꿀 때 유용한 참고 도구입니다:

- **눈누 (noonnu.com)**: 다양한 한글 폰트를 미리 보고 웹폰트 코드를 복사할 수 있는 사이트입니다.
  - **주의**: 눈누의 폰트들이 일괄 상업적 무료인 것은 아닙니다. 폰트마다 인쇄용, 웹 임베드, BI/CI 등 허용 범위가 다르므로 프로젝트 적용 전 개별 라이선스를 반드시 확인해야 합니다.
- **Coolors (coolors.co)**: 어울리는 색상 팔레트를 탐색하고 HEX 색상 코드(예: #1a73e8)를 추천받는 도구입니다.
- 자세한 사용법과 적용 안내는 부록 3. 웹 보조 도구: 눈누 · Bulma · Coolors를 참조합니다.

푸터 추가 코드

실시간 배포 갱신 실습을 위해 `index.html` 하단에 추가할 푸터 마크업과 스타일입니다.

HTML (index.html 본문 하단)



HTML/CSS 실제 브라우저 렌더 결과

전체 예제 코드 · portfolio-1.html

CSS (style.css 하단)

전체 예제 코드 · portfolio-1.css

따라하기

1. 브라우저에서 GitHub의 `portfolio-web` 저장소 페이지를 엽니다.
2. 상단 탭 메뉴에서 `Settings`를 클릭합니다.
3. 좌측 사이드바 메뉴에서 `Pages`를 클릭합니다.
4. `Build and deployment` 영역의 설정을 진행합니다:
   - **Source**: `Deploy from a branch` 선택 상태 유지
   - **Branch**: 드롭다운에서 `None`을 클릭하여 `main`으로 변경
   - **Folder**: `/(root)` 선택 상태 유지
   - 우측의 `Save` 버튼을 클릭합니다.
5. 페이지 상단에 파란색/초록색 알림 상자가 뜨며 배포가 시작됩니다. 약 1~2분 후 새로고침하면 `Your site is live at https://...` 링크가 표시됩니다.

따라하기 · 계속

6. 제공된 배포 링크(`https://계정이름.github.io/portfolio-web/`)를 새 탭에서 열어 내 사이트가 정상적으로 열리는지 확인합니다.
7. VS Code로 돌아와 OpenCode 또는 직접 편집으로 위의 푸터 HTML과 CSS 코드를 추가하고 저장합니다.
8. 터미널에서 변경 사항을 커밋하고 푸시합니다: `git add index.html style.css` `git commit -m "feat: 푸터 연락처 및 저작권 영역 추가"` `git push origin main`
9. GitHub 저장소 상단의 `Actions` 탭을 클릭하여 `pages build and deployment` 워크플로가 돌아가는 것을 확인합니다.
10. 초록색 체크 표시가 뜨면(약 1분 소요), 브라우저의 배포 URL 탭을 강력 새로고침(`Ctrl + Shift + R` 또는 `Cmd + Shift + R`)하여 푸터가 반영되었는지 확인합니다.

Git 명령

```
git status
git add index.html style.css
git commit -m "feat: 푸터 연락처 및 저작권 영역 추가"
```

Git 실습 실행 결과

```
$ git log --oneline -2
d4395cb feat: document deployed portfolio
46c457e feat: add portfolio page
```

격리된 실습 저장소의 실제 실행 출력 (원격 배포 화면 아님)

Git 명령 · 계속

전체 예제 코드 · git-commands-1.sh

에이전트 요청 예시

> index.html 하단 연락처 이메일과 카피라이트 푸터 HTML/CSS 추가
> Coolors 색상 코드 추천 및 배경 포인트 색상 적용

확인법

- ☐ GitHub Pages 설정에서 Branch가 `main` / `/(root)`로 저장되어 있다.
- ☐ 브라우저에서 `https://<계정이름>.github.io/portfolio-web/` 접속 시 첫 웹페이지가 정상 출력된다.
- ☐ 코드를 고치고 `git push`한 후 약 1~2분 뒤 새로고침했을 때 수정된 내용이 라이브 사이트에 반영된다.
- ☐ GitHub `Actions` 탭에서 배포 빌드 작업의 성공(초록색 체크)을 확인할 수 있다.

오류 복구

증상	원인	대응
배포 주소에서 404 Not Found 발생	배포 빌드 진행 중이거나 브랜치 설정 미저장	Settings → Pages에서 Branch가 `main`으로 저장되었는지 확인하고 2~3분 기다린 뒤 다시 접속한다.
첫 화면 파일 인식 불가	파일명이 대문자이거나 이름이 다름	파일명이 반드시 소문자 `index.html`인지 확인한다.
push 후에도 사이트 내용이 안 바뀜	브라우저 캐시로 인해 이전 버전 표시	브라우저 강력 새로고침(`Ctrl + F5` 또는 `Ctrl + Shift + R`)을 누르거나 시크릿 창에서 접속한다.
CSS 스타일이 적용되지 않고 깨짐	`link` 태그의 CSS 파일 경로 오류	`<link rel="stylesheet" href="style.css">`처럼 같은 폴더 상대 경로로 지정되었는지 확인한다.

안전 주의

- GitHub Pages로 배포된 사이트는 인터넷에 완전히 공개되는 웹페이지입니다. 개인 휴대폰 번호, 주민등록번호, 실제 집 주소 등 민감한 개인정보를 올리지 않습니다.
- 눈누 사이트 등에서 웹폰트를 가져올 때는 라이선스 표기 및 상업적/비상업적 허용 범위를 반드시 확인합니다. 세부 안내는 부록 3. 웹 보조 도구: 눈누 · Bulma · Coolors를 참조합니다.
- Day 1의 6개 차시 공식 일정이 완료되었습니다. 남는 버퍼 시간의 추가 심화 실습은 보충안 심화, 복습 퀴즈는 보충안 퀴즈, 에이전트 주도 Git은 보충안 에이전트에서 다룹니다.
- Day 2에는 Markdown 문서화, Issue 관리, 브랜치와 Pull Request 협업을 이어갑니다.